

Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Département de Génie Electrique
Laboratoire de Commande des Processus
Spécialité : **Automatique**
Option : **Systemes Intelligents de Commande et Robotique**

Mémoire de Magister

Présenté par : **BOUAKRIF Farah**
Ingénieur d'Etat en automatique

Thème

COMMANDE PAR APPRENTISSAGE ITERATIF APPLICATION EN ROBOTIQUE

Soutenu le 28 Octobre 2003 devant le jury composé de :

Dr. M.O.MAHMOUDI	Maître de conférences à l'ENP	Président
Pr. F.BOUDJEMA	Professeur à l'ENP	Rapporteur
Dr. D.BOUKHETALA	Maître de conférences à l'ENP	Rapporteur
Dr. M.S.BOUCHERIT	Maître de conférences à l'ENP	Examineur
Dr. M.TADJINE	Maître de conférences à l'ENP	Examineur
Dr. H.CHEKIREB	C.C (Docteur d'Etat)	Examineur
Dr. H.SARI	Maître de conférences	Invité

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique



Remerciements



Les travaux présentés dans ce mémoire ont été effectués au Laboratoire de Commande des Processus (LCP) du Département de Génie Electrique de l'Ecole Nationale Polytechnique (ENP), dirigé par Monsieur le professeur F.BOUDJEMA que je remercie de m'avoir accueilli au sein de cette famille « LCP ».

Je voudrais témoigner ma reconnaissance à Monsieur F.BOUDJEMA, professeur à l'Ecole Nationale Polytechnique, directeur de thèse, pour avoir assuré la responsabilité de mes travaux. Sa disponibilité, et ses discussions trouve ici l'expression de mes vifs remerciements.

J'exprime mes profondes gratitudee et sincères remerciements à Monsieur D.BOUKHETALA, maître de conférences à l'Ecole Nationale Polytechnique, directeur de thèse, pour avoir assuré la responsabilité de mes travaux. Son esprit critique et la confiance qu'il m'a témoigné ont apporté beaucoup à l'aboutissement de ce travail.

Je remercie Monsieur M.O.MAHMOUDI, maître de conférences à l'Ecole Nationale Polytechnique, pour sa participation au jury en tant que président et pour avoir accepté d'être un des examinateurs.

Toute ma reconnaissance va à Monsieur M.S.BOUCHERIT, maître de conférences à l'Ecole Nationale Polytechnique, je lui exprime mes sincères remerciements pour sa participation au jury et pour avoir accepté de juger ce travail en tant que examinateur.

J'exprime mes vifs remerciements à Monsieur M.TADJINE, maître de conférences à l'Ecole Nationale Polytechnique, pour l'honneur qu'il m'a fait en acceptant d'y participer au jury.

Je remercie très profondément Monsieur H.CHEKIREB, Docteur d'Etat à l'Ecole Nationale Polytechnique, pour l'honneur qu'il m'a fait en acceptant d'être membre de jury et pour le grand intérêt porté aux résultats de ce travail.

Que Monsieur M.SARI, maître de conférences à l'Ecole de la Marine, soit vivement remercié pour avoir accepté d'être membre de jury en tant que invité.

Mes très sincères remerciements vont également à Monsieur A.TAYEBI, Docteur à l'Université de Lakehead au Canada, qui a présenté une conférence sur le thème à l'ENP

A toute les personnes qui ont été contribuées de prés ou de loin à l'aboutissement de ce travail trouvent ici le témoignage de ma profonde reconnaissance et je les remercie vivement pour leurs aides efficaces.

Sommaire

Introduction générale 1

Chapitre 1 Notions préliminaires sur la commande par apprentissage itératif

1.1 Introduction	4
1.2 Principe de base	5
1.3 Hypothèses	5
1.4 La commande par apprentissage itératif avec les autres techniques	6
1.5 Les avantages de la commande par apprentissage itératif	7
1.6 Les inconvénients de la commande par apprentissage itératif	7
1.7 Théorie de la commande par apprentissage itératif	8
1.7.1 Objectif	8
1.7.2 Conditions de convergence	9
1.7.2.1 Vitesse de convergence	9
1.7.2.2 Erreur asymptotique	11
1.7.3 Influence des perturbations	12
1.8 Conclusion	15

Chapitre 2 Commande itérative

2.1 Introduction	16
2.2 Description du système	17
2.3 Normes	17
2.4 Hypothèses	19
2.5 Lemmes	20
2.6 Commande par apprentissage itératif autonome <i>off-line</i>	21
2.6.1 Théorème	122
2.6.2 Preuve	22
2.6.3 Exemple de simulation 1	24

3.7.3 Exemple d'application	62
3.8 Commande par apprentissage itératif des systèmes non-linéaires avec degré relatif arbitraire	63
3.8.1 Théorème	64
3.8.2 Preuve	66
3.8.3 Exemples d'application	71
3.9 Conclusion	74
Chapitre 4 Commande par apprentissage itératif appliquée au robot manipulateur	
4.1 Introduction	75
4.2 Formulation du problème	76
4.2.1 Normes utilisées	76
4.2.2 Objectif	78
4.2.3 Lemmes	78
4.3 Elaboration de la loi de commande	79
4.3.1 Hypothèse	81
4.3.2 Proposition	81
4.3.3 Preuve	81
4.4 Résultats de simulation	88
4.5 Conclusion	98
Conclusion générale	99
Annexe	

Introduction générale

L'idée d'utiliser une méthode itérative pour compenser une erreur répétitive n'est pas nouvelle. Lorsqu'une machine exécute la même tâche à plusieurs reprises (cycles), il devient intéressant d'utiliser les informations des cycles précédents en vue de réduire l'erreur à la prochaine exécution de cette tâche. Ceci caractérise l'approche de commande par apprentissage itératif CAI.

En général le problème de commande par apprentissage itératif se pose comme suit : étant donné un système dynamique et une trajectoire de référence, on cherche une commande par apprentissage itératif permettant d'assurer la convergence de l'erreur de poursuite de trajectoire.

En théorie de la commande, si le modèle dynamique du système est disponible, la loi de commande peut être calculée en utilisant la dynamique inverse. Cette approche a été appliquée dans divers domaines, en particulier, la robotique. Lorsque le modèle du système est mal connu ou ses paramètres varient dans le temps, cette technique de commande devient inefficace et la commande adaptative pourra être utilisée. La commande par apprentissage itératif CAI est une alternative intéressante des deux méthodes sus citées, car elle permet, en général, d'éliminer leurs défauts. En effet, elle ne nécessite pas la connaissance exacte du modèle et ne s'intéresse qu'à l'erreur de poursuite pour la correction.

L'idée de base de la commande par apprentissage itératif "CAI" a été publiée en 1968 au Japon par Uchiyama [1], Par ailleurs en 1971 Garden a breveté ses résultats déjà réalisés en 1967 aux Etats-Unis. L'approche consiste à stocker un signal de commande dans la mémoire de l'ordinateur et mettre à jour itérativement le signal de commande en utilisant l'erreur entre la réponse réelle et la réponse désirée. Ceci est fait sans que l'équation de la CAI soit donnée d'une manière explicite. Ce n'est qu'à partir de 1984 et après les travaux de Arimoto [2],

Dans le deuxième chapitre, nous avons présenté les lois de commande itérative connectée *on-line* et autonome *off-line*, et nous avons proposé une commande par apprentissage itératif connectée *on-line* en tenant compte de l'erreur précédente. Des simulations numériques seront, également, donnés.

Dans le troisième chapitre, l'observateur d'état proposé par Tayebi et al. [24] est présenté, sur la base du quel nous avons proposés une commande par apprentissage itératif avec observateur d'état des systèmes non-linéaires à degré relatif arbitraire. Des résultats de simulations seront, également, présentés.

Le dernier chapitre est consacré à l'application de la commande par apprentissage itératif type P au robot manipulateur PUMA 560 et des conditions de convergences sont obtenues.

Chapitre 1

Notions préliminaires sur la commande par apprentissage itératif

1.1 Introduction

La commande par apprentissage itératif est bien connue par ses performances dans la commande des systèmes répétitifs partiellement inconnus et/ou à paramètres variants dans le temps. Initialement cette approche a été proposée pour la commande des robots manipulateurs comme un mécanisme intelligent pour améliorer les performances de poursuite [2].

Cette commande consiste à trouver un mécanisme itératif adéquat permettant d'apprendre les erreurs des cycles antérieurs et exécuter progressivement mieux le nouveau cycle, et par la suite améliorer la poursuite à travers les itérations.

Dans ce chapitre, nous donnons des notions préliminaires de la commande par apprentissage itératif, en mettant en évidence le principe de base de cette dernière, ainsi que l'influence des perturbations sur le système.

- La trajectoire est bornée dans le temps, cette condition provient du fait que la CAI est une méthode autonome. Tous les calculs sont exécutés après que la trajectoire soit terminée.
- Les erreurs sont répétées.
- Les conditions initiales sont les mêmes pour chaque cycle.
- La même trajectoire est parcourue à chaque cycle.

1.4 La commande par apprentissage itératif avec les autres techniques

Afin de situer la commande par apprentissage itératif par rapport aux autres approches, nous donnons quelques comparaisons :

- Commande robuste (CR) :

Dans la commande robuste, un régulateur robuste est synthétisé pour la commande d'un processus. La robustesse d'un système peut être définie comme la capacité de garder les performances vis à vis des variations paramétriques ou incertitudes de modélisation.

- Commande optimale (CO) :

CO détermine une entrée optimale u en fonction du coût prédéfini basée sur le modèle de processus P . Pour la commande optimale, non seulement le modèle exacte de processus est exigé, mais aussi le modèle exacte des perturbations. Ainsi l'inconvénient de cette technique est l'absence de perturbations dans le modèle, et par conséquent, elle ne seront pas compensés. C.à.d que l'optimum calculé n'est pas vraiment optimal.

- Commande adaptative (CA) :

En se basant sur le modèle du processus P , une structure de commande convenable est choisie. La commande adaptative est recherchée en ligne pour que la sortie du système soit optimale, en adaptant les paramètres de ce régulateur.

Bien que maintenant les paramètres soient réglés en ligne basé sur les vraies dimensions

1.7 Théorie de la commande par apprentissage itératif

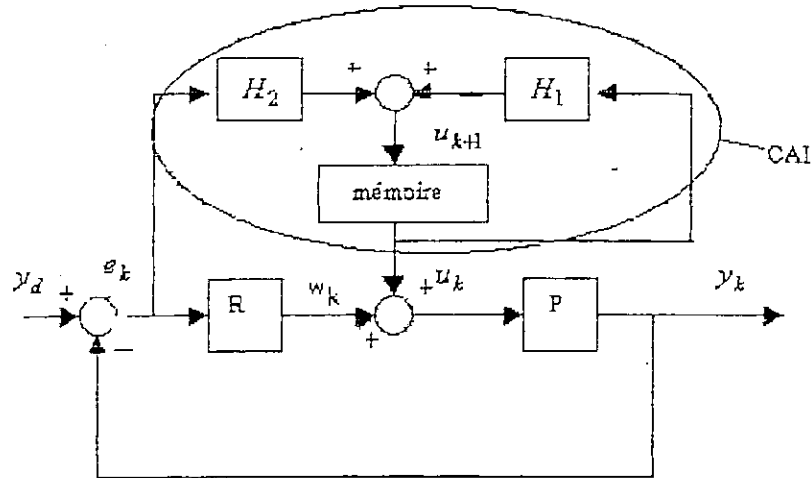


Figure 1.2

On considère le système linéaire suivant (figure 1.2), dont les tâches sont répétées dans un intervalle de temps:

$$Y_k(s) = V_k(s)P(s)$$

avec $V_k(s) = U_k(s) + W_k(s)$

avec k est le nombre des itérations, s est la variable de Laplace.

On suppose que y_d est la trajectoire désirée.

1.7.1 Objectif

Etant donné une trajectoire désirée y_d , en commençant par une commande arbitraire $u_0(t)$, on cherche une commande $u_k(t)$ pour que $y_k(t) \rightarrow y_d(t)$, lorsque $k \rightarrow \infty$.

Plusieurs algorithmes de la commande par apprentissage itératif existent, et la forme la plus générale de ces algorithmes est la suivante:

$$U_{k+1}(s) = H_1(s)U_k(s) + H_2(s)E_k(s) \tag{1.1}$$

H_1, H_2 seront définis par la suite.

$$E_{k+1}(s) = \left(H_1(s) - \frac{P(s)}{1+R(s)P(s)} H_2(s) \right) E_k(s) \quad (1.4)$$

En multipliant les deux membres par e^{-2t} , et en appliquant les normes, on obtient :

$$\|E_{k+1}(s)\|_2 \leq \left\| H_1(s) - \frac{P(s)}{1+R(s)P(s)} H_2(s) \right\|_{\infty} \|E_k(s)\|_2 \quad (1.5)$$

d'où

$$\|E_k(s)\|_2 \leq \left\| H_1(s) - \frac{P(s)}{1+R(s)P(s)} H_2(s) \right\|_{\infty}^k \|E_0(s)\|_2 \quad (1.6)$$

En supposant que $\|E_0(s)\|_2$ est borné sur une certaine gamme de fréquence, alors si

$$\left\| H_1(s) - \frac{P(s)}{1+R(s)P(s)} H_2(s) \right\|_{\infty} < 1 \quad (1.7)$$

on peut assurer que $\|E_k(s)\|_2$ est borné et converge vers zéro lorsque $k \rightarrow \infty$.

En utilisant le théorème de Parseval [19] ($E_k(s)$ est stable), on obtient :

$$\begin{aligned} \|E_k(s)\|_2 &= \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |E_k(j\omega)|^2 d\omega \right)^{1/2} \\ &= \left(\int_{-\infty}^{\infty} |e_k(t)|^2 dt \right)^{1/2} \\ &= \|e_k(t)\|_2 \end{aligned} \quad (1.8)$$

Si (1.7) est réalisé, alors $e_k(t) \rightarrow 0$ dans le sens de \mathcal{L}_2 norme, lorsque $k \rightarrow \infty$.

pour $k \rightarrow \infty$, substituant $E_{k+1} = E_k = E_\infty$:

$$E_\infty = (1 - H_1)E_0 + (H_1 - \frac{P}{1 + RP}H_2)E_\infty \quad (1.16)$$

$$= \frac{(1 - H_1)}{(1 - H_1 + \frac{P}{1 + RP}H_2)} E_0 \quad (1.17)$$

afin d'obtenir une erreur asymptotique nulle, on doit choisir $H_1 = 1$. Et d'après la condition (1.7), H_2 est choisie comme suit:

$$H_2 = \left(\frac{P}{1 + RP}\right)^{-1} \quad (1.18)$$

finalement la loi de commande itérative est donnée par:

$$U_{k+1} = U_k + \left(\frac{P}{1 + RP}\right)^{-1} E_k \quad (1.19)$$

1.7.3 Influence des perturbations

Jusqu'à maintenant le système a été considéré sans perturbations. Ainsi, on va voir maintenant l'influence des perturbations agissantes sur le système.

en ajoutant et en retranchant $H_1 e_0$, $(P/(1+RP))H_1 d_k$ et $(1/(1+RP))H_1 n_k$, on obtient:

$$\begin{aligned}
 e_{k+1} = & e_0 - \frac{P}{1+RP} H_1 u_k - \frac{P}{1+RP} H_2 e_k - \frac{P}{1+RP} d_{k+1} \\
 & - \frac{1}{1+RP} n_{k+1} + H_1 e_0 - H_1 e_0 + \frac{P}{1+RP} H_1 d_k \\
 & - \frac{P}{1+RP} H_1 d_k + \frac{1}{1+RP} H_1 n_k - \frac{1}{1+RP} H_1 n_k
 \end{aligned} \tag{1.24}$$

et par conséquent:

$$\begin{aligned}
 e_{k+1} = & (1-H_1)e_0 + H_1 \left(e_0 - \frac{P}{1+RP} u_k - \frac{P}{1+RP} d_k - \frac{1}{1+RP} n_k \right) \\
 & + \frac{P}{1+RP} H_1 d_k + \frac{1}{1+RP} H_1 n_k \\
 & - \frac{P}{1+RP} H_2 e_k - \frac{P}{1+RP} d_{k+1} - \frac{1}{1+RP} n_{k+1}
 \end{aligned} \tag{1.25}$$

finalment, on trouve:

$$\begin{aligned}
 e_{k+1} = & (1-H_1)e_0 + (H_1 - \frac{P}{1+RP} H_2) e_k + \frac{P}{1+RP} (H_1 d_k - d_{k+1}) \\
 & - \frac{1}{1+RP} (H_1 n_k - n_{k+1})
 \end{aligned} \tag{1.26}$$

pour l'erreur asymptotique e_∞ , ($e_{k+1} = e_k = e_\infty$) on trouve:

Chapitre 2

Commande itérative

2.1 Introduction

La commande par apprentissage itératif a été largement étudiée pour des systèmes linéaires à temps continu, et elle a été développée pour certaines classes de systèmes non linéaires. Dans [10], la convergence et la robustesse ont été étudiées sur certaines lois de commande itératives. Dans [20], des commandes itératives à temps discret sont proposées, elles sont basées sur l'approximation de celles à temps continu équivalentes. La convergence de l'entrée et de l'état vers la trajectoire désirée avec une erreur bornée, est montrée. Cependant, le développement de la commande par apprentissage itératif pour des systèmes à temps discret a été restreint à des systèmes linéaires [21].

Dans ce chapitre, nous analysons une classe des systèmes non linéaires, dont la sortie peut être complètement non linéaire. Les lois de la CAI autonome "off-line", et, connectée "on-line" sont présentées pour des systèmes non linéaires avec une loi d'apprentissage itératif pour état initial. La convergence des algorithmes sera établie ou l'information sur l'état désiré et l'entrée désirée n'est pas utilisée dans l'analyse de la convergence de notre algorithme.

En second lieu, nous examinerons, des systèmes non linéaires perturbés à temps variant. Les limites pour les erreurs de poursuite sont obtenues, dépendant des perturbations du système, et des erreurs initiales. De plus, les états initiaux désirés peuvent être identifiés à travers l'apprentissage itératif.

Le problème de la commande par apprentissage itératif pour des systèmes dynamiques non linéaires à temps discret sera ensuite étudié [22]. En premier temps, α -norme sera définie pour des fonctions à temps discret, suite de la quelle une commande par apprentissage itératif robuste sera présentée.

Considérant le système (1), et avec une trajectoire désirée $y_d(t)$, $t \in [0, T]$. Une loi de commande par apprentissage itératif autonome "off-line", est donnée par:

$$u_{k+1}(t) = \Phi(u_k(t), e_k(t), t) \quad , \quad k = 1, 2, 3, \dots$$

et la loi de commande par apprentissage itératif connectée "on-line", est donnée par:

$$u_{k+1}(t) = \Psi(u_k(t), e_{k+1}(t), t) \quad , \quad k = 1, 2, 3, \dots$$

$u_k(t)$, $x_k(t)$, $e_k(t) = y_d(t) - y_k(t)$ designent respectivement l'entrée, l'état et l'erreur à l'itération k . Les schémas de la CAI autonome *off-line*, et connectée *on-line*, sont présentés par les figures 2.1 et 2.2.

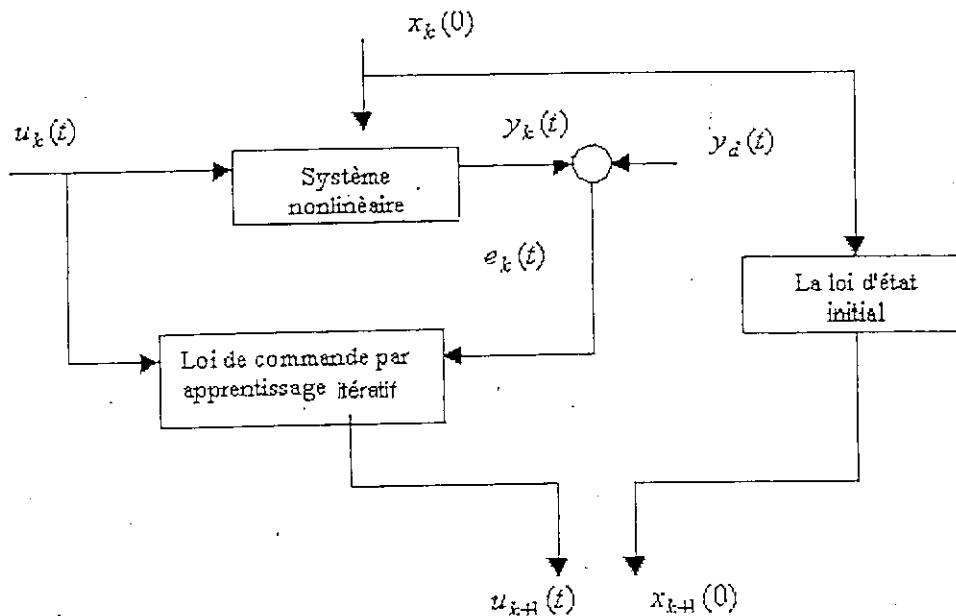


figure 2.1 Schéma de principe de la CAI autonome *off-line*

2.5 Lemmes [14]

Lemme 1

soit $z(t) = [z_1(t), z_2(t), \dots, z_n(t)]^T \in \mathbb{R}^n$ définie pour $t \in [0, T]$, alors pour $t \in [0, T]$:

$$\left(\int_0^t \|z(s)\| ds \right) e^{-\lambda t} \leq \frac{1}{\lambda} \|z(t)\|_\lambda$$

Preuve

$$\begin{aligned} \left(\int_0^t \|z(s)\| ds \right) e^{-\lambda t} &= \int_0^t \|z(s)\| e^{-\lambda s} e^{\lambda(t-s)} ds \\ &\leq \left[\sup_{0 \leq s \leq t} \{ \|z(s)\| e^{-\lambda s} \} \right] \int_0^t e^{-\lambda(t-s)} ds \\ &\leq \left[\sup \{ \|z(t)\| e^{-\lambda t} \} \right] \frac{(1 - e^{-\lambda t})}{\lambda} \\ &\leq \frac{1}{\lambda} \|z(t)\|_\lambda \end{aligned}$$

lemme 2 (Gronwall-Bellman).

soit $f(t)$ et $g(t) \geq 0$ sont des fonctions scalaires réelles localement intégrables dans $[a, b]$, et L est une constante. Si la fonction $f(t)$ satisfait l'inégalité suivante

$$f(t) \leq L + \int_0^t g(\tau) f(\tau) d\tau \quad , \quad t \in [a, b]$$

alors

$$f(t) \leq L \exp\left(\int_0^t g(\tau) d\tau \right)$$

lemme 3

soit $\xi(t), \eta_1(t), \eta_2(t)$ des fonctions continues dans $[0, T]$, et

$$\|\xi(t)\| \leq \Gamma \int_0^t \|\xi(s)\| ds + \Gamma_1 \int_0^t \|\eta_1(s)\| ds + \Gamma_2 \|\eta_2(t)\| \quad \text{pour } t \in [0, T], \text{ alors:}$$

$$\|\xi(t)\|_\lambda \leq \left(\frac{1}{\lambda} \Gamma_1 \|\eta_1(t)\|_\lambda + \Gamma_2 \|\eta_2(t)\|_\lambda \right) \exp\left(\frac{\Gamma}{\lambda} \right)$$

2.6.1 Théorème 1 [14]

On suppose que le système (2.1) satisfait l'hypothèse H1, et la sortie désirée $y_d(t)$ satisfait l'hypothèse H2. Pour n'importe quel état initial $x_1(0)$ et commande admissible $u_1(t) \in U$, en appliquant la loi de commande itérative (2.3) et la loi itérative de l'état initial (2.4) au système (2.1), alors l'erreur converge vers zéro $\forall t \in [0, T]$ lorsque $k \rightarrow \infty$, c.a.d

$$\lim_{k \rightarrow \infty} y_k(t) = y_d(t), \quad \forall t \in [0, T]$$

Si

-1- $h_x(x, t)$ est bornée dans $R^n \times [0, T]$;

-2- $\sup_{(x,t) \in R^n \times [0,T]} \|I_m - h_x(x, t)B(t)K(t)\| < 1$.

2.6.2 Preuve

Considérons le système (2.1), avec les loi (2.3) et (2.4), on aura:

$$\begin{aligned} x_{k+1}(t) &= x_{k+1}(0) + \int_0^t (f(x_{k+1}(s), s) + B(s)u_{k+1}(s)) ds \\ &= x_k(0) + B(0)K(0)e_k(0) + \int_0^t f(x_{k+1}(s), s) ds + \int_0^t B(s)(u_k(s) + K(s)\dot{e}_k(s)) ds \\ &= x_k(0) + B(0)K(0)e_k(0) + \int_0^t f(x_{k+1}(s), s) ds + \int_0^t B(s)u_k(s) ds + B(t)K(t)e_k(t) \\ &\quad - B(0)K(0)e_k(0) - \int_0^t \frac{d(B(s)K(s))}{ds} e_k(s) ds \\ &= x_k(0) + \int_0^t [f(x_{k+1}(s), s) + B(s)u_k(s)] ds + \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds \\ &\quad + B(t)K(t)e_k(t) - \int_0^t \frac{d(B(s)K(s))}{ds} e_k(s) ds \\ &= x_k(t) + \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds + B(t)K(t)e_k(t) - \int_0^t \frac{d(B(s)K(s))}{ds} e_k(s) ds \end{aligned}$$

par conséquent

$$x_{k+1}(t) - x_k(t) = \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds + B(t)K(t)e_k(t) - \int_0^t \frac{d(B(s)K(s))}{ds} e_k(s) ds \quad (2.5)$$

d'après l'hypothèse H1

$$\sup_{(x,t) \in R \times [0,T]} \|I_m - h_x(x(t),t)B(t)K(t)\| + \frac{\delta}{\lambda} [\alpha \gamma \exp(\frac{\alpha}{\lambda}) + b] = \rho < 1$$

donc $\|e_{k+1}(t)\|_{\lambda} \leq \rho \|e_k(t)\|_{\lambda}$, $k = 1, 2, 3, \dots$

finalemt $\lim_{k \rightarrow \infty} \|e_k(t)\|_{\lambda} = 0$, $t \in [0, T]$

remarque: il est claire que le résultat est indépendant de l'état désiré et de l'entrée désirée, l'état initial et l'entrée initiale de la première itération sont choisis arbitrairement.

2.6.3 Exemple de simulation 1

On considère le modèle dynamique d'un bras manipulateur donné par Bien et al.(1991) [9] comme suit:

$$J_m \ddot{q}(t) + Sg \sin(q(t)) = u(t)$$

avec g est l'accélération gravitationnelle, $u(t)$ est le couple appliqué, et $q(t)$ est l'angle de rotation du bras en radians.

On pose $x_1 = q$ et $x_2 = \dot{q}$, donc notre système est décrit par

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -J_m^{-1} Sg \sin x_1 \end{bmatrix} + \begin{bmatrix} 0 \\ J_m^{-1} \end{bmatrix} u$$

La sortie choisie est $y = \frac{2}{5} x_2$

avec $J_m = 14 \text{kgm}^2$, $S = 6 \text{kg.m}$, $g = 9.8 \text{m/s}^2$, $T = 2 \text{s}$. alors on trouve: $h_x = \begin{bmatrix} 0, \frac{2}{5} \end{bmatrix}$,

$$\alpha = \frac{6}{7} \times 9.8 = 8.4.$$

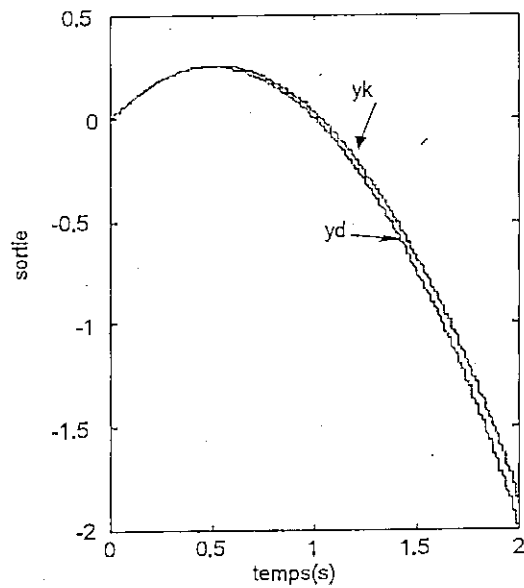


Fig.2.5 Réponse pour k=10

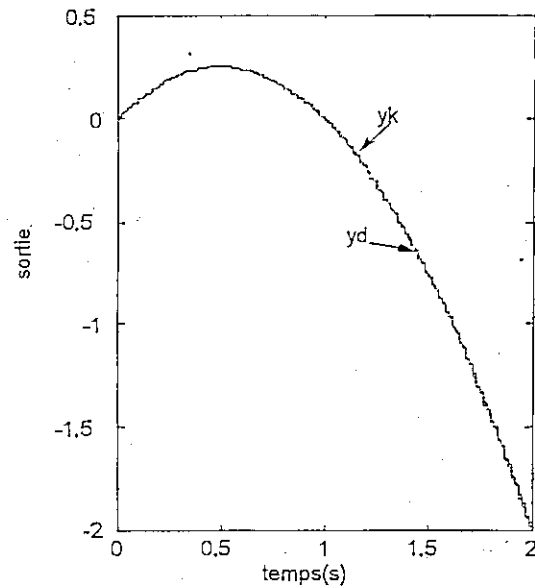


Fig.2.6 Réponse pour k=20

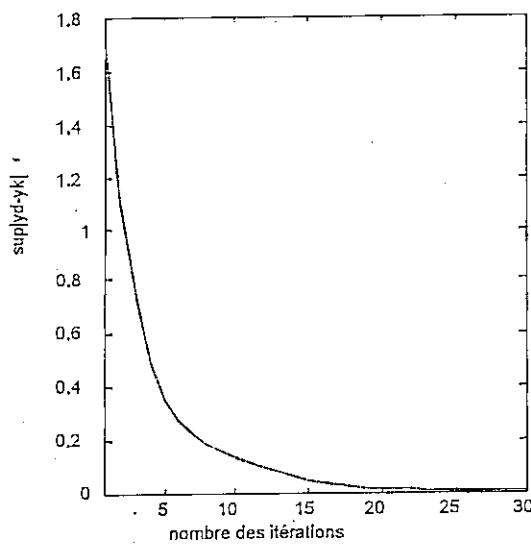


Fig.2.7 Erreur de poursuite

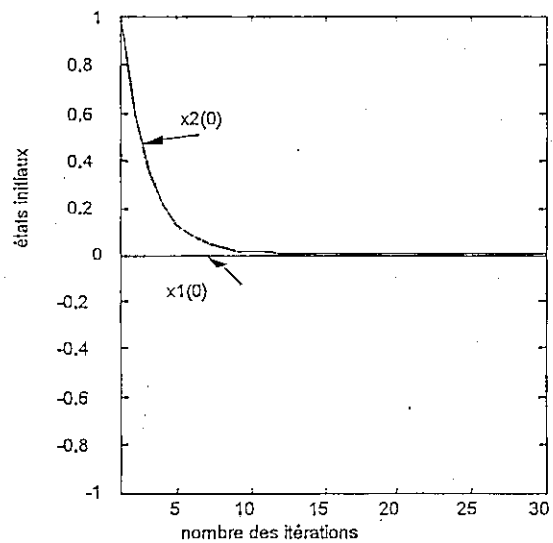


Fig.2.8 Etats initiaux

On remarque qu'après 20 itérations, la sortie réelle tend vers la sortie désirée. Ce résultat nous a prouvé l'efficacité de la loi de commande par apprentissage itératif autonome "off-line".

2.7 Commande par apprentissage itératif connecté on-line

On considère le système (2.1) :

$$\begin{cases} \dot{x}(t) = f(x(t), t) + B(t)u(t) \\ y(t) = h(x(t), t) \end{cases}$$

d'après la loi (2.9), on trouve:

$$(I_n + B(0)K(0)h_x(x(0)))x_{k+1}(0) = x_k(0) + B(0)K(0)y_d(0)$$

par conséquent:

$$x_{k+1}(0) = x_k(0) + B(0)K(0)(y_d(0) - y_{k+1}(0)) = x_k(0) + B(0)K(0)e_{k+1}(0) \quad (2.11)$$

de (2.10) et (2.11) on obtient:

$$x_{k+1}(t) - x_k(t) = \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds + B(t)K(t)e_{k+1}(t) - \int_0^t \frac{d(B(s)K(s))}{ds} e_{k+1}(s) ds \quad (2.12)$$

de la supposition 1, on trouve:

$$\|x_{k+1}(t) - x_k(t)\| \leq \alpha \int_0^t \|x_{k+1}(s) - x_k(s)\| ds + a \|e_{k+1}(t)\| + b \int_0^t \|e_{k+1}(s)\| ds$$

En appliquant le lemme 3, on trouve :

$$\|x_{k+1}(t) - x_k(t)\|_{\lambda} \leq \gamma \exp\left(\frac{\alpha}{\lambda}\right) \|e_{k+1}(t)\|_{\lambda} \quad (2.13)$$

avec a , b et γ sont définis au paravent.

De (2.7) et (2.12), on obtient :

$$e_{k+1}(t) = e_k(t) - h_x(x_k(t), t)B(t)K(t)e_{k+1}(t) - h_x(x_k(t), t) \left[\int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds - \int_0^t \frac{d(B(s)K(s))}{ds} e_{k+1}(s) ds \right]$$

Pour pouvoir générer la sortie désirée $y_d(t)$ continue dans $[0, T]$, nous choisissons

$$u_d(t) = 35 - 70t - 58.8 \sin\left(\frac{5t^2(2t-3)}{12}\right)$$

avec $y_d(t) = h(x_d(t), t)$, dont $x_d(t)$ est générée par $u_d(t)$.

avec $x_1(0) = [0, 1]$, et $u_1(t) = 1$.

Les résultats de simulation sont présentés par les figures suivantes :

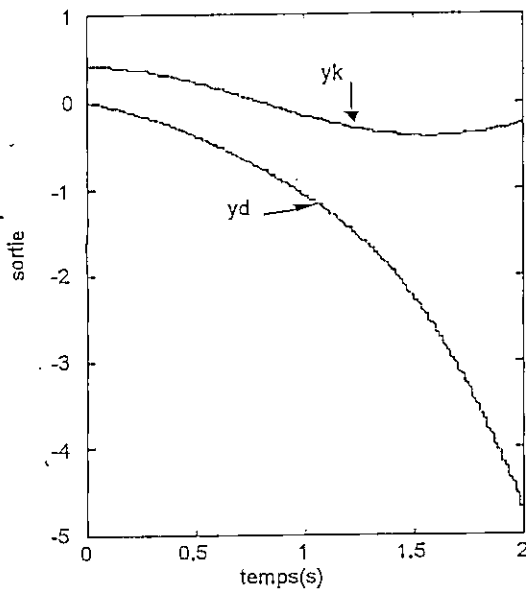


Fig.2.9 Réponse pour $k=1$

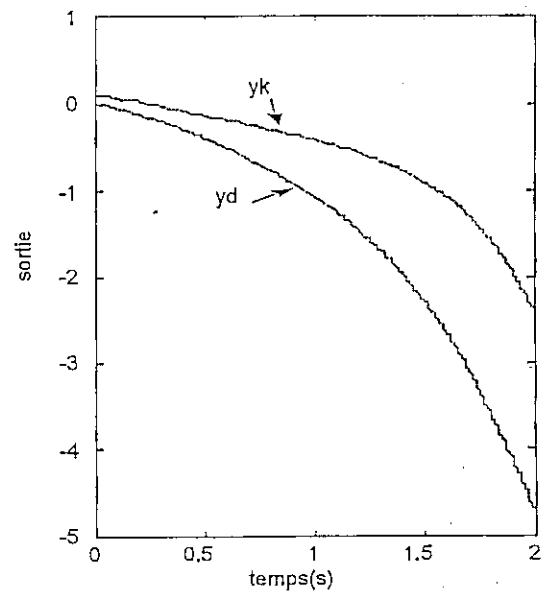


Fig.2.10 Réponse pour $k=5$

2.8 Commande par apprentissage itératif connectée *on-line* en tenant compte l'erreur précédente

On considère le système non linéaire variant dans le temps:

$$\begin{cases} \dot{x}(t) = f(x(t), t) + B(t)u(t) \\ y(t) = h(x(t), t) \end{cases}$$

La loi de commande par apprentissage itératif 2D-type, et la loi itérative de l'état initial utilisées sont les suivantes :

$$\begin{cases} u_{k+1}(t) = u_k(t) + L(t)\dot{e}_k(t) + K(t)\dot{e}_{k+1}(t) & k = 1, 2, \dots \\ x_{k+1}(0) = (I_n + B(0)K(0)h_x(x(0)))^{-1}(x_k(0) + B(0)K(0)y_d(0)) + B(0)L(0)e_k(0) & k = 1, 2, \dots \end{cases} \quad (2.15)$$

$$(2.16)$$

avec $e_k(t) = y_d(t) - y_k(t)$ et $e_{k+1}(t) = y_d(t) - y_{k+1}(t)$ sont l'erreur de sortie pour $k^{ième}$ itération et $(k+1)^{ième}$ itération respectivement, $y_k(t)$ et $x_k(t)$, $y_{k+1}(t)$ et $x_{k+1}(t)$, sont les sorties et les états stimulés par $u_k(t)$ et $u_{k+1}(t)$ dans la $k^{ième}$ et la $(k+1)^{ième}$ itération respectivement, $K(t) \in R^{r \times m}$ et $L(t) \in R^{r \times m}$ sont deux matrices de gain.

$I_n + B(0)K(0)h_x(x(0))$ doit être non singulière.

2.8.1 Proposition

On suppose que le système (2.1) satisfait l'hypothèse H1, et la sortie désirée $y_d(t)$ satisfait l'hypothèse H2. Pour n'importe quel état initial $x_1(0)$ et commande admissible $u_1(t) \in U$, en appliquant la loi de commande itérative (2.15) et la loi itérative de l'état initial (2.16) au système (2.1), alors l'erreur converge uniformément vers zéro $\forall t \in [0, T]$ lorsque $k \rightarrow \infty$, c.a.d

$$\lim_{k \rightarrow \infty} y_k(t) = y_d(t), \quad \forall t \in [0, T]$$

Si

- 1- $h_x(x, t)$ est bornée dans $R^n \times [0, T]$;
- 2- $I_m + h_x(x(t), t)B(t)K(t)$ est non singulière ;
- 3- $\sup_{(x,t) \in R^n \times [0, T]} \left\| (I_m - h_x(x, t)B(t)L(t)) \cdot (I_m + h_x(x, t)B(t)K(t))^{-1} \right\| < 1$.

$$\begin{aligned}
 [I_m + h_x(x_k(t), t)B(t)K(t)]e_{k+1}(t) = & [I_m - h_x(x_k(t), t)B(t)L(t)]e_k(t) - h_x(x_k(t), t) \\
 & \int_0^t [f(x(s), s) - f(x_k(s), s)]ds - h_x(x_k(t), t) \quad (2.19) \\
 & \left[\int_0^t \frac{d(B(s)L(s))}{ds} e_k(s)ds + \int_0^t \frac{d(B(s)K(s))}{ds} e_{k+1}(s)ds \right]
 \end{aligned}$$

de (2.18), (2.19) et en tenant compte la supposition 1, on trouve:

$$\|e_{k+1}(t)\|_\lambda \leq \left[\frac{\max_{t \in [0, T]} \|(I_m - h_x(x(t), t)B(t)L(t))\| + \frac{\delta}{\lambda} \left[\alpha \gamma_1 \exp\left(\frac{\alpha}{\lambda}\right) + b \right]}{\max_{t \in [0, T]} \|(I_m + h_x(x(t), t)B(t)K(t))\| - \frac{\delta}{\lambda} \left[\alpha \gamma_2 \exp\left(\frac{\alpha}{\lambda}\right) + d \right]} \right] \|e_k(t)\|_\lambda$$

en choisissant λ ($\lambda > 0$) largement grand, on obtient:

$$\|e_{k+1}(t)\|_\lambda \leq \max_{t \in [0, T]} \left\{ \|(I_m - h_x(x(t), t)B(t)L(t))(I_m + h_x(x(t), t)B(t)K(t))^{-1})\| \right\} \|e_k(t)\|_\lambda$$

Si $\max_{t \in [0, T]} \left\{ \|(I_m - h_x(x(t), t)B(t)L(t))(I_m + h_x(x(t), t)B(t)K(t))^{-1})\| \right\} < 1$

Alors $y_k(t)$ converge uniformément vers $y_d(t)$, pour $t \in [0, T]$, lorsque $k \rightarrow \infty$.

2.8.3 Exemple de simulation 3

On prend le même exemple précédent, la loi de commande itérative proposée est donnée par:

$$\begin{aligned}
 u_{k+1}(t) &= u_k(t) + 50\dot{e}_k(t) + 14\dot{e}_{k+1}(t) \\
 x_{k+1}(0) &= \begin{bmatrix} 1 & 0 \\ 0 & 5/7 \end{bmatrix} x_k(0) + \begin{bmatrix} 0 \\ 5/7 \end{bmatrix} y_d(0) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e_k(0)
 \end{aligned}$$

avec $L(t) = 10$ et $K(t) = 14$ pour que les conditions dans le théorème 3 soient satisfaites.

Pour une fonction continue $y_d(t)$ dans $[0, T]$, il est existe $u(t) \in U$, tel que $y_d(t) = h(x_d(t), t)$, avec $x_d(t)$ est générée par $u_d(t)$, qui est donnée par:

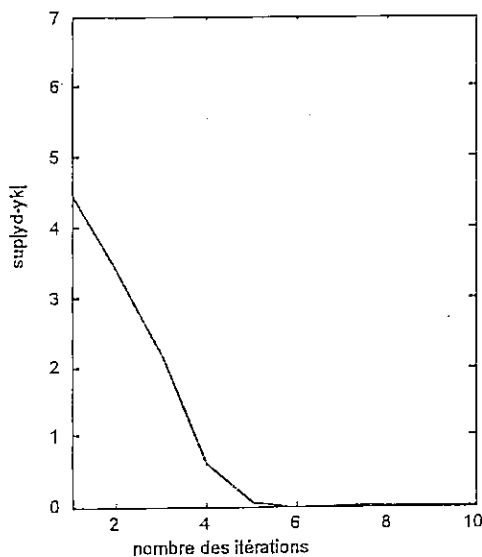


Fig.2.19 Sup de l'erreur de poursuite

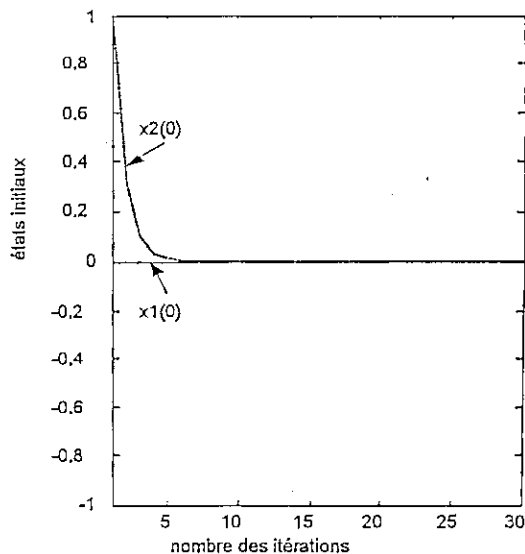


Fig.2.20 Etats initiaux

On remarque qu'après 10 itérations le système se corrige, et la sortie réelle tend vers la sortie désirée. Donc on peut conclure que la CAI "on-line" en tenant compte l'erreur précédente est plus efficace que la CAI "on-line" et la CAI "off-line".

2.9 Commande par apprentissage itératif pour un système non linéaire perturbé

On considère le système non linéaire variant dans le temps en présence de perturbation

$$\begin{cases} \dot{x}_k(t) = f(x_k(t), t) + B(t)u_k(t) + w_k(t) \\ y_k(t) = h(x_k(t), t) + v_k(t) \end{cases} \quad (2.20)$$

avec $x_k \in R^n$, $u_k \in R^r$ et $y_k \in R^m$ représentent respectivement, le vecteur d'état, la commande d'entrée et la sortie de système, $w_k(t)$ et $v_k(t)$ sont des perturbations. La fonction f est continue et peut être inconnue, $h(x(t), t)$ et $B(t)$ sont des fonctions continues. L'intervalle du temps associé est $[0, T]$.

La commande par apprentissage itératif proposée de type PD, et la loi d'apprentissage de l'état initial sont données par :

$$u_{k+1}(t) = u_k(t) + K(t)\dot{e}_k(t) + L(t)e_k(t) \quad k = 1, 2, \dots \quad (2.21)$$

$$x_{k+1}(0) = x_k(0) + B(0)K(0)e_k(0) \quad k = 1, 2, \dots \quad (2.22)$$

$$\lim_{k \rightarrow \infty} e_k(t) \rightarrow 0, \quad t \in [0, T].$$

$$\text{avec : } \mu_1 = \sup_{t \in [0, T]} \|B(t)K(t)\|, \quad \mu_2 = \sup_{t \in [0, T]} \|B(t)L(t)\|, \quad \mu_3 = \sup_{t \in [0, T]} \left\| \frac{d[B(t)K(t)]}{dt} \right\|$$

$$\delta = \sup_{(x, t) \in R \times [0, T]} \{ \|h_x(x, t)\| \}.$$

2.9.3 Preuve

Considérons le système (2.20), avec les loi (2.21) et (2.22), on aura:

$$\begin{aligned} x_{k+1}(t) &= x_{k+1}(0) + \int_0^t (f(x_{k+1}(s), s) + B(s)u_{k+1}(s) + w_{k+1}(s)) ds \\ &= x_k(0) + B(0)K(0)e_k(0) + \int_0^t f(x_{k+1}(s), s) ds + \int_0^t B(s)(u_k(s) + K(s)\dot{e}_k(s) \\ &\quad + L(s)e_k(s)) ds + \int_0^t w_{k+1}(s) ds \\ &= x_k(0) + B(0)K(0)e_k(0) + \int_0^t f(x_{k+1}(s), s) ds + \int_0^t B(s)u_k(s) ds + B(t)K(t)e_k(t) \\ &\quad - B(0)K(0)e_k(0) - \int_0^t \frac{d(B(s)K(s))}{ds} e_k(s) ds + \int_0^t B(s)L(s)e_k(s) ds + \int_0^t w_{k+1}(s) ds \\ &= x_k(0) + \int_0^t [f(x_{k+1}(s), s) + B(s)u_k(s) + w_k(s)] ds + \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds \\ &\quad + B(t)K(t)e_k(t) - \int_0^t \frac{d(B(s)K(s))}{ds} e_k(s) ds + \int_0^t B(s)L(s)e_k(s) ds + \int_0^t (w_{k+1}(s) - w_k(s)) ds \\ &= x_k(t) + \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds + B(t)K(t)e_k(t) - \int_0^t \frac{d(B(s)K(s))}{ds} e_k(s) ds \\ &\quad + \int_0^t B(s)L(s)e_k(s) ds + \int_0^t (w_{k+1}(s) - w_k(s)) ds \end{aligned}$$

par conséquent

$$\begin{aligned} x_{k+1}(t) - x_k(t) &= \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds + B(t)K(t)e_k(t) - \int_0^t \frac{d(B(s)K(s))}{ds} e_k(s) ds \\ &\quad + \int_0^t B(s)L(s)e_k(s) ds + \int_0^t (w_{k+1}(s) - w_k(s)) ds \end{aligned} \quad (2.23)$$

$$\|e_{k+1}(t)\|_\lambda \leq \left\{ \|I_m - h_x(x_k(t), t)B(t)K(t)\| + \frac{\delta}{\lambda} \left[\mu_2 + \mu_3 + \alpha \left(\mu_1 + \frac{\mu_2 + \mu_3}{\lambda} \right) \exp\left(\frac{\alpha}{\lambda}\right) \right] \right\} \|e_k(t)\|_\lambda + \delta \cdot b_w T \left(1 + \frac{\alpha}{\lambda} \exp\left(\frac{\alpha}{\lambda}\right) \right) + b_v$$

avec $\delta = \sup_{(x,t) \in R \times [0,T]} \{ \|h_x(x,t)\| \}$.

avec la condition (2), il existe λ ($\lambda > 0$) assez grand, pour tout $t \in [0, T]$, on aura:

$$\sup_{(x,t) \in R \times [0,T]} \left\{ \|I_m - h_x(x_k(t), t)B(t)K(t)\| + \frac{\delta}{\lambda} \left[\mu_2 + \mu_3 + \alpha \left(\mu_1 + \frac{\mu_2 + \mu_3}{\lambda} \right) \exp\left(\frac{\alpha}{\lambda}\right) \right] \right\} = \rho < 1$$

et donc

$$\|e_k(t)\|_\lambda \leq \frac{\delta \cdot b_w T \left(1 + \frac{\alpha}{\lambda} \exp\left(\frac{\alpha}{\lambda}\right) \right) + b_v}{1 - \left(\|I_m - h_x(x_k(t), t)B(t)K(t)\| + \frac{\delta}{\lambda} \left[\mu_2 + \mu_3 + \alpha \left(\mu_1 + \frac{\mu_2 + \mu_3}{\lambda} \right) \exp\left(\frac{\alpha}{\lambda}\right) \right] \right)}$$

finalement , si $b_w = 0$ et $b_v = 0$, alors $\lim_{k \rightarrow \infty} \|e_k(t)\|_\lambda = 0$, $t \in [0, T]$.

2.9.4 Exemple de simulation 4

Etant donné le système non linéaire variant dans le temps perturbé est donné par [23]

$$\begin{bmatrix} \dot{x}_{1k} \\ \dot{x}_{2k} \end{bmatrix} = \begin{bmatrix} \alpha_1 \sin(x_{2k}) & 1 + \alpha_1 \sin(x_{1k}) \\ -2 - 5t & -3 - 2t \end{bmatrix} \begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix} + \begin{bmatrix} u_{1k} \\ 2u_{2k} \end{bmatrix} + \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix}$$

$$\begin{bmatrix} y_{1k}(t) \\ y_{2k}(t) \end{bmatrix} = \begin{bmatrix} 2x_{1k}(t) \\ x_{2k}(t) \end{bmatrix} + \begin{bmatrix} v_{1k}(t) \\ v_{2k}(t) \end{bmatrix}$$

Le temps $t \in [0, T]$. Les perturbations sont données par :

2.10 Commande par apprentissage itératif des systèmes non linéaire discret

2.10.1 Préliminaires

Les notations suivantes sont utilisées. R^n est l'espace Euclidien de dimension n , avec la norme $\|x\| = (x^T x)^{1/2}$ pour $x \in R^n$. $A \in R^{n \times m}$ est une matrice de dimension $(n \times m)$ avec des éléments réels, et $\|A\| = \sqrt{\lambda_{\max}(A^T A)}$ est la norme de la matrice, avec $\lambda_{\max}(\cdot)$ est la "eigenvalue" maximale pour la matrice symétrique. Soit $\Omega = [0, 1, 2, \dots, N]$, avec N assez grand. De plus, on définit les normes suivantes [22]:

Définition 1: α -norme d'une fonction réelle positive g est définie par:

$$g: k \in \Omega \rightarrow R$$

$$\|g(\cdot)\|_{\alpha} = \sup_{k \in \Omega} g(k) (1/\alpha)^k \quad \text{avec } \alpha \geq 1$$

Définition 2: ∞ -norme d'une fonction réelle positive g est définie par:

$$g: k \in \Omega \rightarrow R$$

$$\|g(\cdot)\|_{\infty} = \sup_{k \in \Omega} g(k)$$

Lemme 1: Les normes des deux définitions 1 et 2 sont équivalentes.

Preuve: le lemme est justifié en notant:

$$\|g(k)\|_{\alpha} \leq \|g(k)\|_{\infty} \leq \alpha^N \|g(k)\|_{\alpha}$$

Lemme 2: En donnant l'inégalité suivante:

$$z(k+1) \leq \beta(k) + hz(k) \tag{2.27}$$

$$\begin{aligned}x_d(k+1) &= f(x_d(k), k) + B(x_d(k), k)u_d(k) \\ y_d(k) &= A(k)x_d(k)\end{aligned}$$

h2 Les fonctions $f(x(k), k)$ et $B(x(k), k)$ sont satisfaites à la condition de continuité de lipschitz dans x sur Ω , c.a.d:

$$\begin{aligned}\|f(x_1(k), k) - f(x_2(k), k)\| &\leq c_f \|x_1(k) - x_2(k)\|, \quad \text{et} \\ \|B(x_1(k), k) - B(x_2(k), k)\| &\leq c_B \|x_1(k) - x_2(k)\|,\end{aligned}$$

pour $k \in \Omega$ et c_f, c_B sont des constants positifs.

h3 Les matrices $B(x(k), k)$ et $A(x(k), k)$ sont bornées, dans le sens de :

$$\|B(x(k), k)\| \leq b_B \quad \text{et} \quad \|A(k)\| \leq b_A \quad \text{pour } (x, k) \in R^n \times \Omega, \text{ avec } b_b \text{ et } b_A \text{ sont des constants positifs. en outre, le rang de la matrice } AB \text{ est plein.}$$

h4 Les perturbations sont bornées dans le sens de : $\|v_i(k)\| \leq b_v$ et $\|w_i(k)\| \leq b_w$ pour tout $i > 0$, $k \in \Omega$, avec b_v et b_w sont des constants positifs.

h5 Une séquence de sortie $y_d(k)$, $k \in \Omega$ est donnée comme la sortie désirée. Elle est réalisable avec une séquence d'entrée $u_d(k)$, $k \in \Omega$ lorsque $w(k)$ et $v(k)$ sont égaux à zéro, pour $k \in \Omega$. De plus, $\|u_d(k)\| \leq b_{ud}$, avec b_{ud} est un constant positif. Pour la même séquence d'entrée, correspond une séquence des états désirés $x_d(k)$, $k \in \Omega$, qui est bornée aussi.

h6 Tout les opérations commencent avec une valeur au voisinage de $x_d(0)$ dans le sens de :

$$\|x_d(0) - x_i(0)\| \leq b_{x0} \quad \text{pour } i \geq 0.$$

2.10.3 Loi de commande par apprentissage itératif

La loi de commande proposée, après la $i^{\text{ème}}$ course d'opération est la suivante :

$$u_{i+1}(k) = u_i(k) + L(k)[y_d(k+1) - y_i(k+1)] \quad (2.31)$$

$$\|\delta u_{i+1}(k)\| \leq \rho \|\delta u_i(k)\| + b_1 + h_2 \|\delta x_i(k)\| \quad (2.34)$$

avec $\|I - L(k)A(k+1)B(x(k), k)\| \leq \rho < 1$,

$$b_1 = b_L (b_A b_w + b_v),$$

$$h_1 = c_f + b_{ud} c_B,$$

et $h_2 = b_L b_A h_1$.

En prenant la différence entre l'équation

$$x_d(k+1) = f(x_d(k), k) + B(x_d(k), k)u_d(k)$$

et l'équation (2.29), on trouve :

$$\begin{aligned} \delta x_i(k+1) &= f(x_d(k), k) - f(x_i(k), k) + [B(x_d(k), k) - B(x_i(k), k)]u_d(k) \\ &\quad + B(x_i(k), k)\delta u_i(k) - w_i(k) \end{aligned} \quad (2.35)$$

en prenant les normes des deux membres de l'équation (2.35), on trouve :

$$\|\delta x_{i+1}(k)\| \leq h_1 \|\delta x_i(k)\| + b_B \|\delta u_i(k)\| + b_w \quad (2.36)$$

en utilisant le lemme 2 dans l'équation (2.36), on obtient :

$$\|\delta x_i(k)\| \leq \sum_{j=0}^{k-1} h_1^{k-1-j} [b_B \|\delta u_i(j)\| + b_w] + h_1^k b_{x0} \quad (2.37)$$

Substituant (2.37) dans (2.34), on trouve :

$$\|\delta u_{i+1}(k)\| \leq \rho \|\delta u_i(k)\| + b_1 + h_2 h_1^k b_{x0} + h_2 \sum_{j=0}^{k-1} h_1^{k-1-j} [b_B \|\delta u_i(j)\| + b_w] \quad (2.38)$$

En multipliant les deux membres de l'équation (2.38) par $(1/\alpha)^k$, avec $\alpha > \max[1, h_1]$, on trouve:

$$\begin{aligned} \|\delta u_{i+1}(k)\| (1/\alpha)^k &\leq \rho \|\delta u_i(k)\| (1/\alpha)^k + b_1 (1/\alpha)^k + h_2 b_{x0} (h_1/\alpha)^k \\ &\quad + \frac{h_2}{\alpha} \sum_{j=0}^{k-1} (h_1/\alpha)^{k-1-j} [b_B \|\delta u_i(j)\| (1/\alpha)^j + b_w (1/\alpha)^j] \end{aligned} \quad (2.39)$$

$$\limsup_{i \rightarrow \infty} \|\delta.x_i\|_{\alpha} \leq \frac{b_B [1 - (h_1/\alpha)^N] \varepsilon}{(\alpha - h_1)(1 - \hat{\rho})} + \frac{b_w [1 - (h_1/\alpha)^N]}{\alpha - h_1} + b_{x0} \quad (2.46)$$

finalement, de l'équation (2.30), on trouve

$$\|\delta.y_i\|_{\alpha} \leq b_A \|\delta.x_i\|_{\alpha} + b_v \quad (2.47)$$

ce qui nous donne

$$\limsup_{i \rightarrow \infty} \|\delta.y_i\|_{\alpha} \leq \frac{b_A b_B [1 - (h_1/\alpha)^N] \varepsilon}{(\alpha - h_1)(1 - \hat{\rho})} + \frac{b_A b_w [1 - (h_1/\alpha)^N]}{\alpha - h_1} + b_{x0} b_A + b_v \quad (2.48)$$

Dans l'absence de w_i et v_i , et lorsque $b_{x0} = 0$, alors on obtient $\varepsilon = 0$, et par la suite $\|\delta.u_i\|_{\alpha} \rightarrow 0$, $\|\delta.x_i\|_{\alpha} \rightarrow 0$ et $\|\delta.y_i\|_{\alpha} \rightarrow 0$ comme résultat.

Remarque : Dans la loi de commande (2.31), on a utilisé l'erreur de poursuite $y_d(k+1) - y_i(k+1)$, parce que dans la course précédente, l'erreur de poursuite à $k+1$ est du de l'action de la commande $u_i(k)$.

2.11 Conclusion

Dans ce chapitre, nous avons résolu le problème de la commande par apprentissage itératif des systèmes non linéaires variants dans le temps discret, en proposant une loi de commande itérative type P, et en assurant la convergence de l'entrée, de l'état et de la sortie vers leurs trajectoires désirées, dans la présence de certaines perturbations bornées. Dans le cas où ces perturbations tendent vers zéro, alors on obtient une convergence idéale.

Chapitre 3

Commande par apprentissage itératif avec observateur d'état

3.1 Introduction

L'idée de la commande par apprentissage itératif CAI est d'utiliser la connaissance des itérations antérieures de la même tâche pour réduire l'erreur de poursuite de la prochaine exécution de la tâche.

Dans [24] Tayebi a proposé un observateur d'état pour un système non-linéaire avec degré relatif égal à un. Dans [25], Sun et Gang ont proposés une commande par apprentissage itératif pour des systèmes non-linéaires avec degré relatif arbitraire.

Dans ce chapitre, nous présentons d'abord l'observateur d'état proposé par Tayebi [24]. Par la suite, nous proposons une commande par apprentissage itératif avec observateur d'état pour un système non-linéaire avec degré relatif arbitraire. Dans un premier temps, l'observateur d'état est dérivé du système considéré, et nous donnons les conditions de la convergence de l'erreur estimée. Puis nous présentons le degré relatif et le degré relatif étendu des systèmes non-linéaires, et enfin, nous évaluons l'erreur de poursuite, en proposant une loi de commande itérative basée sur l'observateur d'état proposé.

3.2 Formulation de problème

Le système considéré est:

$$\begin{cases} \dot{x}(t) = f(x(t), t) + B(x(t), t)u(t) + \phi(x(t), u(t), t) \\ y(t) = g(x(t), t) \end{cases} \quad (3.1)$$

$x \in R^n$, $u \in R^m$ et $y \in R^p$ représentent respectivement, le vecteur d'état, la commande d'entrée et la sortie du système. Les fonctions $f(*) \in R^n$, $B(*) = [b_1(*), \dots, b_r(*)] \in R^{n \times r}$ et $g(*) = [g_1(*) \dots \dots g_m(*)]^T \in R^m$ sont unis dans leurs domaine de définition, et elles sont connues uniquement avec certaines propriétés. A chaque $x(0)$ fixe, S dénote le plan de $(x(0), u(t), t \in [0, T])$ vers $(x(t), t \in [0, T])$, et O dénote le plan de $(x(0), u(t), t \in [0, T])$ vers $(y(t), t \in [0, T])$. Avec ces notations, $x(*) = S(x(0), u(*))$ et $y(*) = O(x(0), u(*))$.

Si le système est répétitif alors, on obtient le système suivant :

$$\begin{cases} \dot{x}_k(t) = f(x_k(t), t) + B(x_k(t), t)u_k(t) + \phi(x_k(t), u_k(t), t) \\ y_k(t) = g(x_k(t), t) \end{cases} \quad (3.2)$$

ϕ est la fonction du vecteur estimé.

3.2.1 Objectif

Etant donné une trajectoire désirée différentiable et un système non-linéaire avec degré relatif arbitraire μ . En commençant par une commande arbitraire $u_0(t)$, on cherche à trouver une loi de commande itérative pour que:

$$\lim_{k \rightarrow \infty} \|y_d(t) - y_k(t)\| \leq \varepsilon \quad \text{pour } t \in [0, T]$$

3.2.2 hypothèses [24, 25]

h1 La gamme admissible de la commande. $u_k(t)$ est donnée par $\|u_k(t)\| \leq u_{\max}$ ou u_{\max} est une valeur connue obtenue par les limitations physiques du système.

3.2.3 Normes utilisées

$$\|M\| = \max_{1 \leq i \leq n} \left(\sum_{j=1}^m |m_{ij}| \right) \quad \text{avec } M = (m_{ij}), \quad 1 \leq i \leq n, \quad 1 \leq j \leq m$$

$$\|V\| = \max_{1 \leq i \leq n} |V_i| \quad \text{avec } V = (V_i), \quad 1 \leq i \leq n$$

$$\|* (t)\|_{\infty} = \sup_{t \in [0, T]} \|* (t)\|$$

$$\|* (t)\|_{\lambda} = \sup_{t \in [0, T]} (e^{-\lambda t} \|* (t)\|)$$

3.3 Degré relatif d'un système non-linéaire

Le degré relatif d'un système non-linéaire est le nombre de fois qu'on dérive la sortie pour que la commande apparaît dans la sortie.

Le degré relatif égal à μ s.s.i:

$$\begin{cases} L_B L_f^k g(x) = 0 & 0 \leq k \leq \mu - 2 \\ L_B L_f^{\mu-1} g(x) \neq 0 \end{cases}$$

3.4 Degré relatif étendu [25, 26]

Le degré relatif étendu d'un système non-linéaire est le nombre de fois qu'on intègre certain terme pour que la sortie $y(t+\sigma)$ soit dynamiquement retardée par rapport à $u(t)$.

Le degré relatif étendu égal à μ s.s.i:

$$\begin{cases} L_B L_f^i g(x) = 0 & 0 \leq i \leq \mu_q - 2 \\ \int_t^{t+\sigma} \int_t^{t_1} \dots \int_t^{t_{\mu-1}} L_B L_f^{\mu-1} g(x(t_{\mu})) dt_{\mu} \dots dt_1 \neq 0 \end{cases}$$

Donc la formule de $y_q(t+\sigma)$ est la suivante:

$$y_q(t+\sigma) = g(x(t)) + \sigma L_f g(x(t)) + \dots + \frac{\sigma^{\mu_q-1}}{(\mu_q-1)!} L_f^{\mu_q-1} g(x(t)) + \int_t^{t+\sigma} \int_t^{t_1} \dots \int_t^{t_{\mu_q-1}} [L_f^{\mu_q} g(x(t_{\mu_q})) + [L_{B_1} L_f^{\mu_q-1} g(x(t_{\mu_q})), \dots, L_{B_r} L_f^{\mu_q-1} g(x(t_{\mu_q}))] u(t_{\mu_q})] dt_{\mu_q} \dots dt_1$$

3.5 Observateur d'état [24]

On considère l'observateur d'état suivant :

$$\begin{cases} \dot{\hat{x}}_k(t) = f(\hat{x}_k(t), t) + B(\hat{x}_k(t), t)u_k(t) + \phi(\hat{x}_k(t), u_k(t), t) + L(y_k(t) - \hat{y}_k(t)) \\ \hat{y}_k(t) = g(\hat{x}_k(t), t) \end{cases} \quad (3.3)$$

L est le gain de l'observateur.

En premier temps, il faut vérifier que $\hat{x}_k \rightarrow x_k$

soit :

$$e(t) = x_k(t) - \hat{x}_k(t)$$

En dérivant, on obtient :

$$\dot{e}(t) = \dot{x}_k(t) - \dot{\hat{x}}_k(t)$$

ou encore

$$\dot{e}_k = [f_x(x_k) - Lg_x(x_k)]e_k + (B_k - \hat{B}_k)u_k + \phi_k - \hat{\phi}_k$$

avec

$$B_k = B(x_k(t), t), \quad \phi_k = \phi(x_k(t), u_k(t), t),$$

$$\begin{aligned} &\leq 2\|e_k\|^2 \|P\| (l_B u_{\max} + l_{\phi_x}) \\ &\leq e_k^T \lambda^2 P P e_k + e_k^T e_k \end{aligned}$$

avec $\lambda = l_B u_{\max} + l_{\phi_x}$

finalement on trouve:

$$\dot{V} \leq e_k^T Q e_k$$

avec

$$\begin{cases} Q = A^T P + P A + \lambda^2 P P + I \\ A = f_x(x_k) - L g_x(x_k) \end{cases}$$

d'après [24, 27], on a le lemme suivant :

3.6 Lemme [24, 27]

Si $(f_x(x_k), g_x(x_k))$ observable, et si $(f_x(x_k) - L g_x(x_k))$ stable, et

$$\min_{w \in \mathbb{R}^+} \sigma_{\min}(f_x(x) - L g_x(x) - jwI) > \lambda,$$

avec $\sigma_{\min}(\ast)$ est la plus petite valeur singulière de (\ast) , alors l'erreur d'observation e_k est bornée, et converge asymptotiquement vers zéro.

3.7 Commande par apprentissage itératif

La commande par apprentissage itératif proposée dans [24] est la suivante:

$$\begin{aligned} v_{k+1}(t) &= u_k(t) + \Gamma_k(t) [\dot{y}_d(t) - g_x(x_k) f(\hat{x}_k(t), t) - g_x(x_k) B(\hat{x}_k(t), t) u_k(t) \\ &\quad - g_x(x_k) \phi(\hat{x}_k(t), u_k(t), t)] \\ u_{k+1}(t) &= \text{sat}(v_{k+1}(t)) \end{aligned} \tag{3.4}$$

$$\gamma_1 = l_{\phi,u} + c_B$$

$$\gamma_2 = \|Lg_x(x_k)\|$$

$$\zeta_1 = \|I - \Gamma_k(t)g_x(x_k)B(\hat{x}_k(t),t)\|_\infty + l_{\phi,u}\|\Gamma_k(t)g_x(x_k)\|_\infty$$

$$\zeta_2 = \frac{\alpha\gamma_1}{\lambda - \alpha} (1 - e^{(\alpha-\lambda)T}) \|\Gamma_k(t)g_x(x_k)\|_\infty$$

$$\zeta_3 = \frac{\gamma_2\zeta_2}{\gamma_1} \|e_k\|_\lambda + \alpha \|\Gamma_k(t)g_x(x_k)\|_\infty \|\tilde{x}_k(0)\|$$

$$\alpha = \|f_x(x_k)\| + l_B \|u_d\|_\infty + l_{\phi,x}$$

$$\varepsilon = \min \left\{ 2u_{\max} \frac{\zeta_3}{1 - \zeta_1 - \zeta_2} \right\}$$

3.7.2 Preuve

Soit l'erreur de commande $\tilde{u}_k = u_d - u_k$ et $\tilde{v}_k = u_d - v_k$, ainsi

$$\begin{aligned} \tilde{v}_{k+1} &= u_d - v_{k+1} \\ &= \tilde{u}_k - (v_{k+1} - u_k) \\ &= \tilde{u}_k - \Gamma_k(\dot{y}_d - g_x(x_k)f(\hat{x}_k) - g_x(x_k)B(\hat{x}_k)u_k - g_x(x_k)\phi(\hat{x}_k, u_k)) \\ &= \tilde{u}_k - \Gamma_k(g_x(x_k)f(x_d) + g_x(x_k)B(x_d)u_d + g_x(x_k)\phi(x_d, u_d) \\ &\quad - g_x(x_k)f(\hat{x}_k) - g_x(x_k)B(\hat{x}_k)u_k - g_x(x_k)\phi(\hat{x}_k, u_k)) \\ &= (I - \Gamma_k g_x(x_k)B(\hat{x}_k))\tilde{u}_k - \Gamma_k g_x(x_k)f(\tilde{x}_k) - \Gamma_k g_x(x_k)(B(x_d) - B(\hat{x}_k))u_d \\ &\quad - \Gamma_k g_x(x_k)(\phi(x_d, u_d) - \phi(\hat{x}_k, u_k)) \end{aligned} \quad (3.5)$$

avec $\tilde{x}_k = x_d - \hat{x}_k$, et x_d est le vecteur d'état généré par la commande u_d .

D'après la supposition h2 et la définition de u_{k+1} dans (3.4), on peut conclure que $\|\tilde{u}_{k+1}\| \leq \|\tilde{v}_{k+1}\|$.

De h2 et h5, et en prenant la norme de (3.5), on trouve :

En multipliant les deux membres de l'inégalité (3.9) par $e^{-\lambda t}$, avec $\lambda > 0$, et en appliquant λ -norme, on obtient :

$$\begin{aligned}
\|\tilde{u}_{k+1}\|_\lambda &\leq \left(\|I - \Gamma_k g_x(x_k) B(\hat{x}_k)\|_\infty + l_{\phi,u} \|\Gamma_k g_x(x_k)\|_\infty \right) \|\tilde{u}_k\|_\lambda + \sup_{t \in [0, T]} \left\{ \alpha \|\Gamma_k g_x(x_k)\| \|\tilde{x}_k(0)\| e^{(\alpha-\lambda)t} \right\} \\
&\quad + \sup_{t \in [0, T]} \left\{ \alpha \|\Gamma_k g_x(x_k)\| \int_0^t e^{-\lambda \tau} (\gamma_1 \|\tilde{u}_k\| + \gamma_2 \|e_k\|) e^{(\alpha-\lambda)(t-\tau)} d\tau \right\} \\
&\leq \left(\|I - \Gamma_k g_x(x_k) B(\hat{x}_k)\|_\infty + l_{\phi,u} \|\Gamma_k g_x(x_k)\|_\infty \right) \|\tilde{u}_k\|_\lambda \\
&\quad + \sup_{t \in [0, T]} \left\{ \alpha \|\Gamma_k g_x(x_k)\| \int_0^t \gamma_1 e^{(\alpha-\lambda)(t-\tau)} d\tau \right\} \|\tilde{u}_k\|_\lambda \\
&\quad + \sup_{t \in [0, T]} \left\{ \alpha \|\Gamma_k g_x(x_k)\| \int_0^t \gamma_2 e^{(\alpha-\lambda)(t-\tau)} d\tau \right\} \|e_k\|_\lambda + \sup_{t \in [0, T]} \left\{ \alpha \|\Gamma_k g_x(x_k)\| \|\tilde{x}_k(0)\| e^{(\alpha-\lambda)t} \right\}
\end{aligned} \tag{3.10}$$

d'où

$$\|\tilde{u}_{k+1}\|_\lambda \leq (\zeta_1 + \zeta_2) \|\tilde{u}_k\|_\lambda + \zeta_3 \tag{3.11}$$

Avec

$$\zeta_1 = \|I - \Gamma_k(t) g_x(\hat{x}_k) B(\hat{x}_k(t), t)\|_\infty + l_{\phi,u} \|\Gamma_k(t) g_x(x_k)\|_\infty \tag{3.12}$$

$$\zeta_2 = \frac{\alpha \gamma_1}{\lambda - \alpha} (1 - e^{-(\lambda - \alpha)T}) \|\Gamma_k(t) g_x(x_k)\|_\infty \tag{3.13}$$

$$\zeta_3 = \frac{\gamma_2 \zeta_2}{\gamma_1} \|e_k\|_\lambda + \alpha \|\Gamma_k(t) g_x(x_k)\|_\infty \|\tilde{x}_k(0)\| \tag{3.14}$$

En choisissant le gain tel que $\|\Gamma_k g_x(x_k)\|_\infty$ soit borné, il est clair qu'il existe λ largement grand pour que ζ_2 soit arbitrairement petit. Donc, si $\zeta_1 < 1$, il existe λ tel que $\zeta_1 + \zeta_2 < 1$.

d'où

$$\lim_{k \rightarrow \infty} \|\tilde{u}_k\|_\lambda \leq \varepsilon \tag{3.15}$$

$$\lim_{k \rightarrow \infty} \|y_d(t) - y_k(t)\|_\lambda \leq \|g_x(x_k)\| \frac{\gamma_1 (1 - e^{-(\alpha - \lambda)T})}{(\lambda - \alpha)} \varepsilon \quad (3.21)$$

Si la condition dans le lemme 1 est vérifiée, alors $\|e_k\|_\lambda$ est bornée. d'où, $\zeta_3 \rightarrow \alpha \| \Gamma_k(t) g_x(x_k) \|_\infty \| \tilde{x}_k(0) \|$ lorsque $k \rightarrow \infty$. En plus, si $\tilde{x}_k(0) = 0$ alors $\zeta_3 \rightarrow 0$ et donc $\varepsilon \rightarrow 0$, ce qui implique, l'erreur de poursuite tend vers zéro lorsque $k \rightarrow \infty$.

3.7.3 Exemple d'application

On considère le système suivant :

$$\begin{cases} \dot{x}_1 = x_2 + 0.1e^{-t} \sin(x_1) \\ \dot{x}_2 = x_1 + 0.1 \sin(x_2) + (0.05 + 0.05e^{-t})u \\ y = x_2 \end{cases}$$

$$f(x) = \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}, \quad B(x(t), t) = \begin{pmatrix} 0 \\ 0.05 + 0.05e^{-t} \end{pmatrix},$$

$$\phi(x) = \begin{pmatrix} 0.1e^{-t} \sin(x_1) \\ 0.1 \sin(x_2) \end{pmatrix}, \quad g(x) = \begin{pmatrix} 0 \\ x_2 \end{pmatrix}^T$$

avec la commande admissible $|u| \leq 5$. $l_{\phi,u} = 0$, $l_{\phi,x} = 0.1$, $c_B = 0.1$, $l_B = 0.1$ et $\lambda = 0.2$, $\sigma = 0.1$.

On prend $L = [L_1 \ L_2]^T = [5 \ 4]^T$, $f(x) - L_x g_x(x)$ est stable, et $\min_{\omega \in \mathbb{R}^+} \sigma_{\min} (f_x(x) - L g_x(x) - j\omega I) = 0.509 > 0.2$, $y_d(t) = 0.05t^2(1-t)$, $t \in [0,1]$, en utilisant la loi de commande (3.4), avec $G_k(t) = \frac{10}{0.05 + 0.05e^{-t}}$, et l'observateur d'état (3.3), avec $x_1(0) = \hat{x}_1(0) = 0$, $x_2(0) = \hat{x}_2(0) = 0$, $u_1(t) = 0$, on obtient le résultat présenté dans la figure 1.

$$sat(v) = [sat(v_1), \dots, sat(v_r)]^T$$

Avec

$$sat(v_p) = \begin{cases} v_p & \dots \text{if } |v_p| \leq \delta_p \\ \text{sgn}(v_p)\delta_p & \dots \text{if } |v_p| > \delta_p \end{cases}$$

δ_p est la borne de saturation d'entrée $p=0 \dots r$, on défini $\delta = \max_{1 \leq p \leq r} \{\delta_p\}$.

3.8.1 Proposition

Etant donné une trajectoire $y_d(t)$, $t \in [0, T]$ et le système (3.2) avec degré relatif étendu $\{\mu_1, \dots, \mu_m\}$, en appliquant la loi de commande par apprentissage itérative (3.25), avec \hat{y} en fonction de \hat{x} qui est donnée par l'observateur d'état (3.3). En supposant que les conditions dans le lemme 1 sont satisfaites, et que les suppositions (h1-h7) sont réalisées.

1- $t \in [0, T - \sigma]$

i) Si $\|I - \Gamma_k(t)D_k(t)\| < 1$, et, il existe $\lambda > \alpha$,

alors la sortie converge vers la sortie désirée dans le sens de :

$$\limsup_{k \rightarrow \infty} \sup_{t \in [0, T - \sigma]} \left\{ e^{-\lambda t} \|y_d(t) - y_k(t)\| \right\} \leq b_g \rho_1 \frac{\rho_s}{1 - \rho}$$

ii) Si $\|I - \Gamma_k(t)D_k(t)\| < 1$, et $\{\hat{x}_k(0) = x_d(0), y_k \rightarrow \hat{y}_k\}$,

alors la sortie converge vers la sortie désirée dans le sens de:

$$\limsup_{k \rightarrow \infty} \sup_{t \in [0, T - \sigma]} \left\{ e^{-\lambda t} \|y_d(t) - y_k(t)\| \right\} \leq b_g \rho_1 \frac{\rho_s}{1 - \rho} \max_{1 \leq q \leq m} \left(\frac{\sigma^{\mu_q}}{\mu_q!} \right)$$

2- $t \in [T - \sigma, T]$

i) Si $\|I - \Gamma_k(t)D_k(t)\| < 1$, et, il existe $\lambda > \alpha$,

alors la sortie converge vers la sortie désirée dans le sens

3.8.2 Preuve

1/ $t \in [0, T - \sigma]$

$$\begin{aligned}\Delta v_{k+1}(t) &= u_d(t) - v_{k+1}(t) \\ &= \Delta v_k(t) - (v_{k+1}(t) - v_k(t)) \\ &= \Delta v_k(t) - \Gamma_k(t)[y_d(t+\sigma) - \hat{y}_k(t+\sigma)]\end{aligned}$$

$$\text{avec } \hat{y}_k(t+\sigma) = [\hat{y}_{1,k}(t+\sigma), \dots, \hat{y}_{m,k}(t+\sigma)]^T$$

alors on obtient:

$$\Delta v_{k+1}(t) = \Delta v_k(t) - \Gamma_k(t) \begin{bmatrix} y_d(t+\sigma) - \hat{y}_{1,k}(t+\sigma) \\ \vdots \\ y_d(t+\sigma) - \hat{y}_{2,k}(t+\sigma) \end{bmatrix}$$

avec $y_{q,k}(t+\sigma)$ est donnée par:

$$\begin{aligned}y_q(t+\sigma) &= g_q(x(t)) + \sigma L_f g_q(x(t)) + \dots + \frac{\sigma^{\mu_q-1}}{(\mu_q-1)!} L_f^{\mu_q-1} g_q(x(t)) + \int_t^{t+\sigma} \int_t^{\dots} \int_t^{\mu_q-1} [L_f^{\mu_q} g_q(x(t_{\mu_q})) \\ &+ [L_{B_1} L_f^{\mu_q-1} g_q(x(t_{\mu_q})), \dots, L_{B_r} L_f^{\mu_q-1} g_q(x(t_{\mu_q}))] u(t_{\mu_q})] dt_{\mu_q} \dots dt_1\end{aligned}$$

donc on obtient:

$$\Delta v_{k+1}(t) = (I - \Gamma_k(t) D_k(t)) \Delta u_k(t) - \Gamma_k(t) [\xi_k(t) + \zeta_k(t) + \varpi_k(t)]$$

$$\xi_k(t) = [\xi_{1,k}(t), \dots, \xi_{m,k}(t)]^T$$

$$\text{avec } \zeta_k(t) = [\zeta_{1,k}(t), \dots, \zeta_{m,k}(t)]^T$$

$$\varpi_k(t) = [\varpi_{1,k}(t), \dots, \varpi_{m,k}(t)]^T$$

$$\begin{aligned} \Delta x_k(t) &= x_d(t) - \hat{x}_k(t) \\ &= \Delta x_k(0) + \int_0^t \left\{ f(x_d) - f(\hat{x}_k) + (B(x_d) - B(\hat{x}_k))u_d + B(\hat{x}_k)[u_d - u_k] \right. \\ &\quad \left. + \phi(x_d, u_d) - \phi(\hat{x}_k, u_k) + L(y_d - \hat{y}_k) \right\} ds \end{aligned}$$

donc on trouve:

$$\|\Delta x_k(t)\| \leq \|\Delta x_k(0)\| + \int_0^t [l_f + l_B c_u + l_{\phi, x}] \|\Delta x_k(s)\| ds + \int_0^t \left\{ (c_B + l_{\phi, u}) \|\Delta u_k(s)\| + c_L \|e_k(s)\| \right\} ds$$

En appliquant le lemme de Bellman-Gronwall, on obtient:

$$\|\Delta x_k(t)\| \leq \|\Delta x_k(0)\| e^{\alpha t} + \int_0^t [B_1 \|\Delta u_k(s)\| + B_2 \|e_k(s)\|] e^{\alpha(t-s)} ds$$

On trouve:

$$\begin{aligned} \|\Delta v_{k+1}(t)\| &\leq \rho \|\Delta u_k(t)\| + c_1 e^{\alpha t} \|\Delta x_k(0)\| + c_1 \int_0^t [B_1 \|\Delta u_k(s)\| + B_2 \|e_k(s)\|] e^{\alpha(t-s)} ds \\ &\quad + c_2 \|\Delta x_k(0)\| \left[\int_t^{t+\sigma} \int_t^{t_1} \dots \int_t^{t_{\mu_1-1}} e^{\alpha t_{\mu_1}} dt_{\mu_1} \dots dt_1 \right] + c_2 \left[\int_t^{t+\sigma} \int_t^{t_1} \dots \int_t^{t_{\mu_1-1}} B_1 \|u_k(s)\| e^{\alpha(t_{\mu_1}-s)} ds dt_{\mu_1} \dots dt_1 \right] \\ &\quad \vdots \\ &\quad \left[\int_t^{t+\sigma} \int_t^{t_1} \dots \int_t^{t_{\mu_m-1}} e^{\alpha t_{\mu_m}} dt_{\mu_m} \dots dt_1 \right] + c_2 \left[\int_t^{t+\sigma} \int_t^{t_1} \dots \int_t^{t_{\mu_m-1}} B_1 \|u_k(s)\| e^{\alpha(t_{\mu_m}-s)} ds dt_{\mu_m} \dots dt_1 \right] \end{aligned}$$

$$\|\delta x_k(t)\| \leq \|\delta x_k(0)\| e^{\alpha t} + \int_0^t B_1 \|u_k(s)\| e^{\alpha(t-s)} ds$$

en multipliant par $e^{-\lambda t}$ on trouve:

$$\sup_{t \in [0, T-\sigma]} \{ e^{-\lambda t} \|\delta x_k(t)\| \} \leq \sup_{t \in [0, T-\sigma]} \{ e^{(\alpha-\lambda)t} \|\delta x_k(0)\| \} + B_1 \frac{1 - e^{(\alpha-\lambda)T}}{\lambda - \alpha} \sup_{t \in [0, T-\sigma]} \{ e^{-\lambda t} \|\Delta u_k(t)\| \}$$

mais $x_d(0) - x_k(0)$ alors $\delta x_k(0) = 0$

finalemt on trouve:

$$\limsup_{k \rightarrow \infty} \sup_{t \in [0, T-\sigma]} \{ e^{-\lambda t} \|y_d(t) - y_k(t)\| \} \leq b_g \rho_1 \frac{\rho_3}{1 - \rho}$$

avec $\rho, \bar{\rho}, \rho_1, \rho_2, \rho_3, \rho_4$ sont définis dans le théorème.

- Si $\hat{x}_k(0) = x_d(0)$ et $\hat{y}_k(t) \rightarrow y_k(t)$, alors :

$$\limsup_{k \rightarrow \infty} \sup_{t \in [0, T-\sigma]} \{ e^{-\lambda t} \|y_d(t) - y_k(t)\| \} \leq b_g \rho_1 \frac{\rho_4}{1 - \bar{\rho}} \max_{1 \leq q \leq m} \left(\frac{\sigma^{\mu_q}}{\mu_q!} \right)$$

2/ Le cas ou $t \in [T - \sigma, T]$:

$$\|\delta x_k(t)\| \leq \|\delta x_k(0)\| e^{\alpha t} + \int_0^{T-\sigma} B_1 \|u_k(s)\| e^{\alpha(t-s)} ds + \int_{T-\sigma}^T B_1 \|u_k(s)\| e^{\alpha(t-s)} ds$$

pour $\lambda > \alpha$ alors:

$$\sup_{t \in [T-\sigma, T]} \{ e^{-\lambda t} \|\delta x_k(t)\| \} \leq \sup_{t \in [T-\sigma, T]} \{ e^{(\alpha-\lambda)t} \|\delta x_k(0)\| \} + B_1 \frac{1 - e^{(\alpha-\lambda)T}}{\lambda - \alpha} \sup_{t \in [T-\sigma, T]} \{ e^{-\lambda t} \|\Delta u_k(t)\| \} + 2\delta B_1 \sigma$$

d'après la supposition h6/, $\|\delta x_k(0) = 0\|$, alors:

$$\phi(x) = \begin{pmatrix} 0.1e^{-t} \sin(x_1) \\ 0.1 \sin(x_2) \end{pmatrix}, \quad g(x) = \begin{pmatrix} x_1 \\ 0 \end{pmatrix}^T$$

on a aussi la commande admissible est $|u| \leq 1$. $l_{\phi,u} = 0$, $l_{\phi,x} = 0.1$, $c_B = 0.1$, $l_B = 0.1$ et $\lambda = 0.2$, $\sigma = 0.1$, $\mu = 2$.

On prend $L = [L_1 \ L_2]^T = [5 \ 4]^T$, $f(x) - L_x g_x(x)$ est stable, et $\min_{w \in \mathbb{R}^+} \sigma_{\min}(f_x(x) - L g_x(x) - j\omega I) = 0.509 > 0.2$, $y_d(t) = 0.05t^2(1-t)$, $t \in [0,1]$, en utilisant la loi de commande (3.4), avec $G_k(t) = \frac{10}{0.05 + 0.05e^{-t}}$, et l'observateur d'état (3.3), avec $x_1(0) = \hat{x}_1(0) = 0$, $x_2(0) = \hat{x}_2(0) = 0$, $u_1(t) = 0$, on obtient le résultat présenté dans la figure 2.

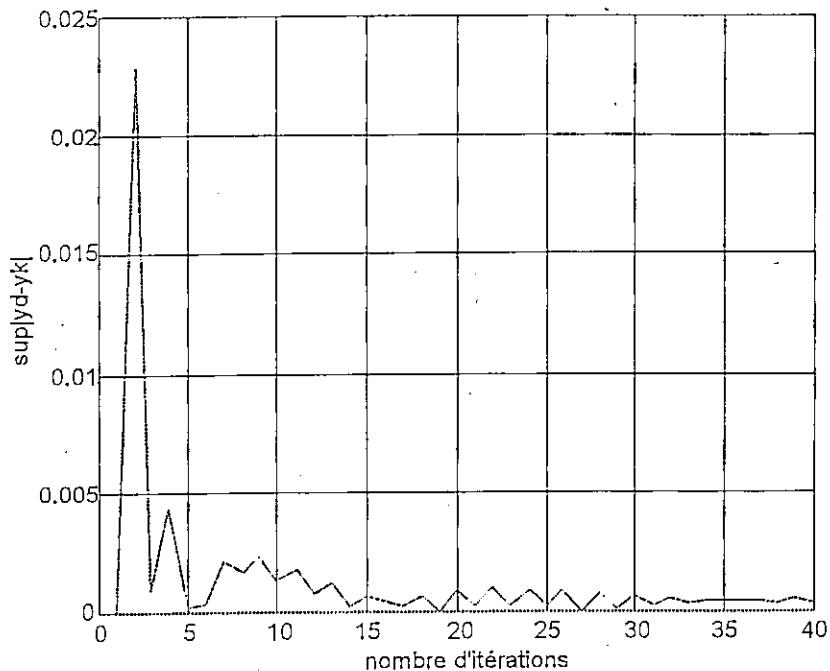


Figure 3.2 Norme sup. de l'erreur de poursuite

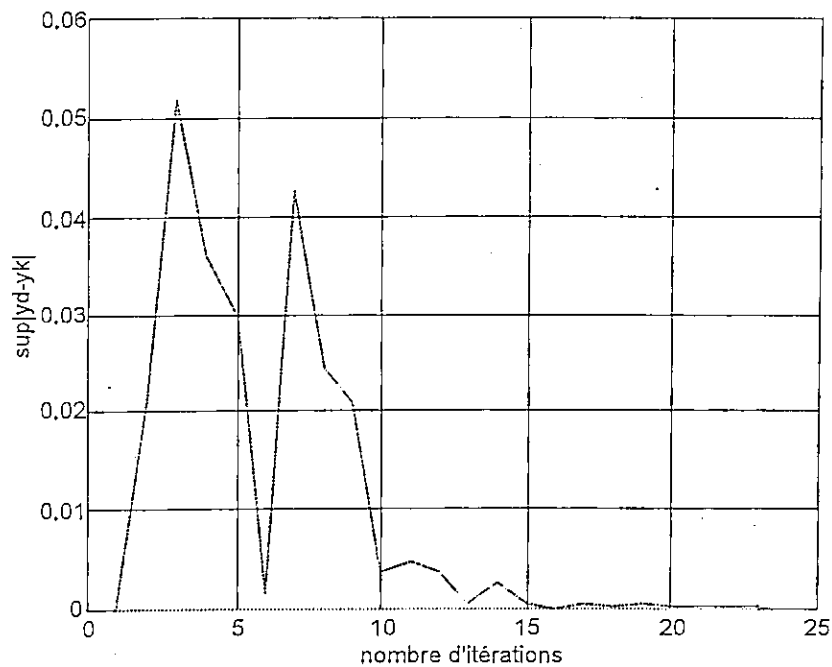


Figure 3.3 Norme sup. de l'erreur de poursuite

3.9 Conclusion

Un observateur d'état avec commande par apprentissage itératif [24] est présenté, les conditions suffisantes de la convergence de l'erreur de l'observation et de l'erreur de poursuite sont données. Une commande par apprentissage itératif avec observateur d'état pour des systèmes non-linéaires avec degré relatif arbitraire est proposée. Les résultats de simulation montrent que les performances du système sont améliorées en appliquant cette loi de commande proposée, par rapport à l'application de loi de commande présentée dans [24].

Chapitre 4

Commande par apprentissage itératif appliquée au robot manipulateur

4.1 Introduction

Beaucoup de machines et robots fonctionnant actuellement sont programmées pour exécuter la même tâche à plusieurs reprises. En observant l'erreur de poursuite à chaque itération de la même tâche, il devient clair que c'est réellement répétitif.

L'idée de base de la commande par apprentissage itératif "CAI", est d'utiliser l'information obtenue de la tâche précédente pour améliorer la commande de la prochaine tâche. Après un certain nombre d'itérations (répétition de tâche), le système de commande se corrige et les performances en poursuite s'améliorent.

La commande par apprentissage itératif a été appliquée avec succès sur les robots manipulateurs, notamment, le travail de Sang et al. [15], le travail de Joon-Young Choi et al. [16] qui ont proposés une "CAI" et ils l'ont appliqués au robot manipulateur de 2ddl, l'application de "CAI" au robot industriel ABB Irb-2000 [17], et au manipulateur de type pont roulant [18].

Dans ce chapitre, une loi de commande par apprentissage itératif P-type est présentée, puis appliquée au robot manipulateur PUMA 560.

4.2 Formulation du problème

On considère l'équation dynamique des robots manipulateurs suivante [16, 30] :

$$M(q(t))\ddot{q}(t) + N(q(t), \dot{q}(t))\dot{q}(t) + G(q(t)) = T(t) \quad (4.1)$$

avec

- $q(t)$: vecteur des positions des articulations;
 $\dot{q}(t)$: vecteur des vitesses des articulations;
 $\ddot{q}(t)$: vecteur des accélérations des articulations;
 $M(q(t)) \in R^{n \times n}$: matrice d'inertie;
 $N(q(t), \dot{q}(t))\dot{q}(t) \in R^n$: vecteur de forces ou couples de coriolis et centrifuge;
 $G(q(t)) \in R^n$: vecteur de forces ou couples dus aux forces de gravitation;
 $T(t) \in R^n$: vecteur de forces ou couples moteurs.

4.2.1 Normes utilisées

$$\|D\| = \max_{1 \leq i \leq n} \left(\sum_{j=1}^m |d_{ij}| \right) \quad \text{avec } D = (d_{ij}), \quad 1 \leq i \leq n, \quad 1 \leq j \leq m$$

$$\|V\| = \max_{1 \leq i \leq n} |V_i| \quad \text{avec } V = (V_i), \quad 1 \leq i \leq n$$

$$\|* (t)\|_{\infty} = \sup_{t \in [0, T]} \|* (t)\|$$

$$\|* (t)\|_{\lambda} = \sup_{t \in [0, T]} (e^{-\lambda t} \|* (t)\|)$$

L'équation dynamique (4.1) possède les propriétés suivantes [16, 28, 29, 30]:

Propriété 1:

La matrice d'inertie $M(q(t))$ est symétrique, définie positive, et bornée :

$$0_n < \beta_1 I_n \leq M(q(t)) \leq \beta_2 I_n \quad (4.2)$$

pour tout $q \in R^n$, et $\beta_2 > \beta_1 > 0$.

4.2.2 Objectif

Le problème de la commande itérative se pose comme suit :

étant donné le modèle dynamique (4.1) du robot manipulateur, la trajectoire désirée $q_d(t)$ est prise comme étant l'entrée de référence du système (4.1), on cherche une séquence de commande $T_k(t) \in R^n$ ($t \in [0, t_f]$) telle que la trajectoire du système $q_k(t)$ converge vers $q_d(t)$, avec une certaine erreur ε , en d'autres termes, $\forall \varepsilon > 0, \exists \eta$ tq :

$$\|q_d(t) - q_k(t)\| \leq \varepsilon \quad \forall t \in [0, t_f], \text{ et } k \geq \eta ;$$

avec k désigne la $k^{\text{ième}}$ itération.

Dans ce qui suit, le système (4.1) est supposé répétitif pour tout $t \in [0, t_f]$, dans le cas où les paramètres du robot restent inchangés quand le système répète ses tâches.

$q_d(t), \dot{q}_d(t)$ et $\ddot{q}_d(t)$ sont respectivement les vecteurs de position, de vitesse et d'accélération désirées.

$q_k(t), \dot{q}_k(t), \ddot{q}_k(t)$ désignent les vecteurs précédents à la $k^{\text{ième}}$ itération, et $T_k(t)$ la commande correspondante.

On suppose que $q_k(0) = q_d(0)$. [16]

4.2.3 Lemmes

Les lemmes suivants sont utilisés dans le développement qui suivent :

Lemme 1 [14] : Soit $z(t) = [z_1(t), z_2(t), \dots, z_n(t)]^T \in R^n$ défini pour $t \in [0, T]$. alors pour $t \in [0, T]$:

$$\left(\int_0^t \|z(s)\| ds \right) e^{-\lambda t} \leq \frac{1}{\lambda} \|z(t)\|_{\lambda} \quad (4.7)$$

Preuve : voir chapitre 2.

lemme 2 [14] : (Gronwall-Bellman).

Soit $f(t)$ et $g(t) \geq 0$ deux fonctions scalaires réelles localement intégrables dans $[a, b]$, et L est une constante. Si la fonction $f(t)$ satisfait l'inégalité intégrante

$$\tau_k(t) = M(q_k(t))[\ddot{q}_d(t) + k_v \dot{e}_k(t) + k_p e_k(t)] + N(q_k(t), \dot{q}_k(t))\dot{q}_k(t) + G_k(t) \quad (4.11)$$

avec

$e_k(t) = q_d(t) - q_k(t)$: vecteur de l'erreur de poursuite en position.

$\dot{e}_k(t) = \dot{q}_d(t) - \dot{q}_k(t)$: vecteur de l'erreur en vitesse.

k_p : matrice diagonale représentant les constantes de l'action proportionnelle du régulateur.

k_v : matrice diagonale représentant les constantes de l'action dérivée du régulateur.

La commande par apprentissage itératif type P $u_k(t)$ a été proposée comme suit :

$$u_{k+1}(t) = u_k(t) + \alpha e_k(t) \quad (4.12)$$

α : matrice diagonale représentant les gains d'apprentissage.

Le schéma de commande est représenté par la figure suivante :

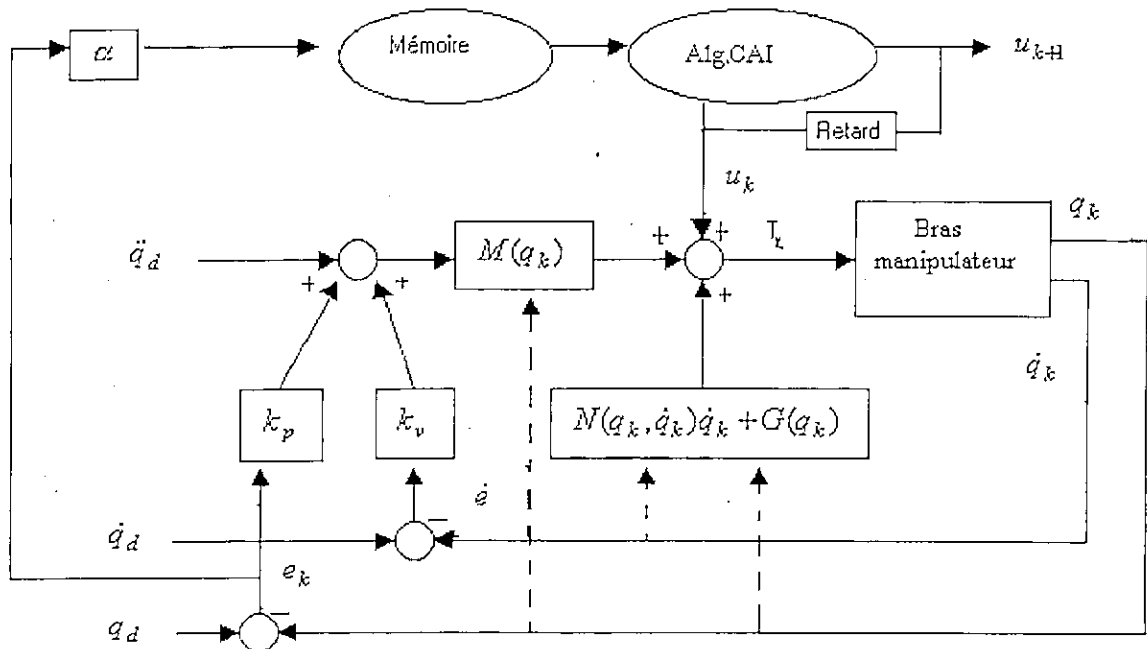


Fig.4.1 schéma de commande itérative combinée avec la commande à couple calculé

$$z_k = \dot{e}_k + \alpha.e_k \quad (4.15)$$

à la $k^{\text{ième}}$ itération, le système (4.1) s'écrit :

$$M_k \ddot{q}_k + N_k \dot{q}_k + G_k = T_k \quad (4.16)$$

En ajoutant et en retranchant $M_k \ddot{q}_d$, $\alpha M_k \dot{e}_k$, $N_k \dot{q}_d$ et $\alpha N_k e_k$, on trouve:

$$\begin{aligned} T_k = & M_k \ddot{q}_k - M_k \ddot{q}_d + M_k \ddot{q}_d - \alpha M_k \dot{e}_k + \alpha M_k \dot{e}_k \\ & + N_k \dot{q}_k - N_k \dot{q}_d + N_k \dot{q}_d - \alpha N_k e_k + \alpha N_k e_k + G_k \end{aligned} \quad (4.17)$$

ce qui nous donne :

$$M_k (\ddot{e}_k + \alpha \dot{e}_k) + N_k (\dot{e}_k + \alpha e_k) = M_k (\ddot{q}_d + \alpha \dot{e}_k) + N_k (\dot{q}_d + \alpha e_k) + G_k - T_k \quad (4.18)$$

soit

$$y_k = M_k (\ddot{q}_d + \alpha \dot{e}_k) + N_k (\dot{q}_d + \alpha e_k) + G_k \quad (4.19)$$

de (4.15), (4.18) et (4.19), on obtient :

$$M_k \dot{z}_k + N_k z_k = y_k - T_k \quad (4.20)$$

sachant que $T_k = \tau_k + u_k$, on trouve :

$$M_k \dot{z}_k + N_k z_k = y_k - \tau_k - u_k \quad (4.21)$$

avec

$$\tau_k = M_k [\ddot{q}_d + k_v \dot{e}_k + k_p e_k] + N_k \dot{q}_k + G_k \quad (4.22)$$

si on fait (4.29)-(4.28), on trouve :

$$\dot{z}_{k+1} - \dot{z}_k = -\delta.(z_{k+1} - z_k) - \gamma.(e_{k+1} - e_k) - (M_{k+1}^{-1} - M_k^{-1})u_k - M_{k+1}^{-1}e_k \quad (4.30)$$

En intégrant les deux membres :

$$z_{k+1} - z_k = -\int_0^t \delta.(z_{k+1} - z_k) ds - \int_0^t \gamma.e_{k+1} ds - \int_0^t (\gamma + \alpha M_{k+1}^{-1})e_k ds - \int_0^t (M_{k+1}^{-1} - M_k^{-1})u_k ds \quad (4.31)$$

En appliquant les normes, et en tenant compte la propriété 1 et le lemme 3, on trouve :

$$\|z_{k+1} - z_k\| \leq a_1 \int_0^t \|z_{k+1} - z_k\| ds + a_2 \int_0^t \|e_{k+1}\| ds + a_3 \int_0^t \|e_k\| ds \quad (4.32)$$

avec $a_1 = \sup|\delta|$

$$a_2 = \sup|\gamma| + l_m \beta_1^{-2} l_u \quad ; \quad l_u = \sup|u_k|$$

$$a_3 = \sup|\gamma| + l_m \beta_1^{-2} l_u + \sup|\alpha|$$

En multipliant les deux membres de (4.32) par $e^{-\lambda t}$, on obtient :

$$\|z_{k+1} - z_k\|_\lambda \leq a_1 \int_0^t \|z_{k+1} - z_k\|_\lambda e^{-\lambda(t-s)} ds + a_2 \left(\int_0^t \|e_{k+1}\| ds \right) e^{-\lambda t} + a_3 \left(\int_0^t \|e_k\| ds \right) e^{-\lambda t} \quad (4.33)$$

d'après le lemme 1, on trouve :

$$\|z_{k+1} - z_k\|_\lambda \leq a_1 \int_0^t \|z_{k+1} - z_k\|_\lambda e^{-\lambda(t-s)} ds + \left(\frac{a_2}{\lambda} \|e_{k+1}\|_\lambda + \frac{a_3}{\lambda} \|e_k\|_\lambda \right) \quad (4.34)$$

En appliquant le lemme 2 de Beiman-Gronwall :

En choisissant

$$\gamma M_k e_k + u_k = l_n |\dot{q}_k| z_k, \quad (4.42)$$

on trouve :

$$\dot{w}_k \leq -\delta m_2 \|z_k\|^2 - z_k^T (l_n |\dot{q}_k| - N_k) z_k \quad (4.43)$$

pour que $\dot{w}_k < 0$ il faut vérifier

$$l_n |\dot{q}_k| - N_k > 0$$

avec $\delta > 0$.

et d'après la propriété 4, cette condition est toujours vérifiée.

De (4.12) et (4.42), pour la $k^{\text{ième}}$ et la $(k+1)^{\text{ième}}$ itération, on obtient :

$$u_k = l_n |\dot{q}_k| z_k - \gamma M_k e_k \quad (4.44)$$

$$u_{k+1} = l_n |\dot{q}_{k+1}| z_{k+1} - \gamma M_{k+1} e_{k+1} = u_k + \alpha e_k \quad (4.45)$$

en faisant (4.45)-(4.44), avec $l_q = \sup |\dot{q}_k|$, on trouve :

$$l_n l_q (z_{k+1} - z_k) - \gamma M_{k+1} e_{k+1} = (\alpha - \gamma M_k) e_k \quad (4.46)$$

ce qui nous donne :

$$e_{k+1} = \frac{l_n l_q}{\gamma M_{k+1}} (z_{k+1} - z_k) + \frac{\gamma M_k - \alpha}{\gamma M_{k+1}} e_k \quad (4.47)$$

4.4 Résultats de simulation

Dans cette partie, nous allons présenter les résultats de simulations de la commande par apprentissage itératif appliquée au robot manipulateur PUMA 560. La méthode numérique de Runge-Kutta d'ordre 0.01 a été utilisée.

Le modèle dynamique du robot PUMA 560 est le suivant :

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) = T - mJ^T(q)[J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} + g]$$

à la $k^{\text{ième}}$ itération, on applique la commande :

$$T_k(t) = \tau_k(t) + u_k(t).$$

Avec

$$\tau_k(t) = M(q_k(t))[\ddot{q}_d(t) + k_v \dot{e}_k(t) + k_p e_k(t)] + N(q_k(t), \dot{q}_k(t))\dot{q}_k(t) + G_k(t)$$

$$u_{k+1}(t) = u_k(t) + \alpha e_k(t)$$

Les valeurs des éléments de M, N et G sont données dans l'annexe .

Les valeurs des éléments de la matrice de gain, et les vecteurs k_v et k_p sont comme suit :

$$\alpha = \text{diag}\{300, 310, 5500\} ; k_v = \text{diag}\{310, 311, 5508\} \text{ et } k_p = \text{diag}\{10000, 2000, 50000\}.$$

$$t \in [0, 4].$$

$$q_{1d}(t) = -\frac{\pi}{2} + \frac{1}{4} \left(2\pi \frac{t}{4} - \sin \left(2\pi \frac{t}{4} \right) \right)$$

$$q_{2d}(t) = \frac{1}{4} \left(2\pi \frac{t}{4} - \sin \left(2\pi \frac{t}{4} \right) \right)$$

$$q_{3d}(t) = -\frac{\pi}{2} + \frac{1}{4} \left(2\pi \frac{t}{4} - \sin \left(2\pi \frac{t}{4} \right) \right)$$

$$q_k(0) = [-\pi/2 \quad 0 \quad -\pi/2]^T, \quad u_1(t) = [0 \quad 0 \quad 0]^T$$

Les figures suivantes montrent les positions des trois premières articulations du robot manipulateur PUMA 560, les erreurs en poursuite, et la commande appliquée après 1 itération, 5 itérations et 20 itérations. Les performances obtenues après 20 itérations sont nettement supérieure à celles obtenues après 1 itération.

Nous remarquons qu'à travers les itérations le comportement du robot en poursuite s'améliore, et les erreurs de poursuite sont bien moins importantes, ceci justifie l'efficacité de la commande par apprentissage itératif.

Nous constatons qu'après 20 itérations la poursuite est satisfaisante et les erreurs de poursuite sont très faibles.

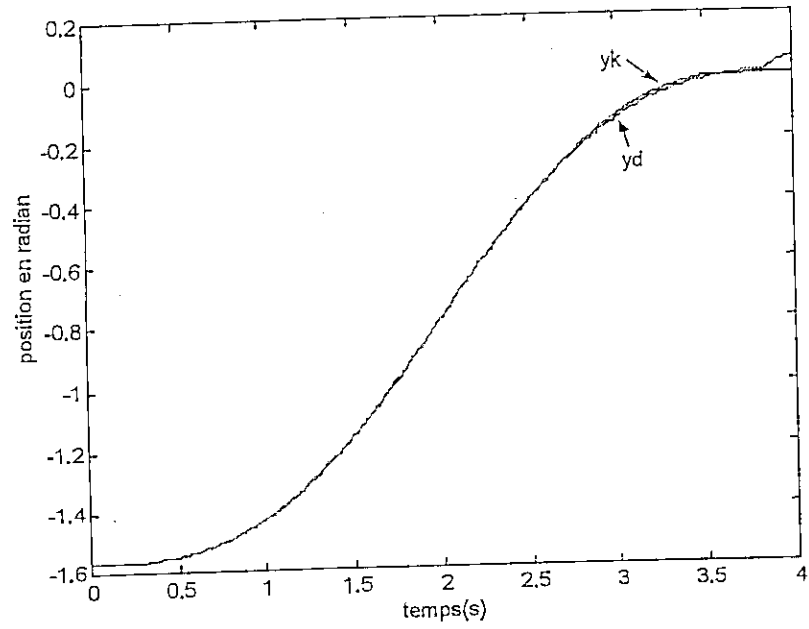


Fig. 4.3a position de la 1ère articulation pour $k=5$

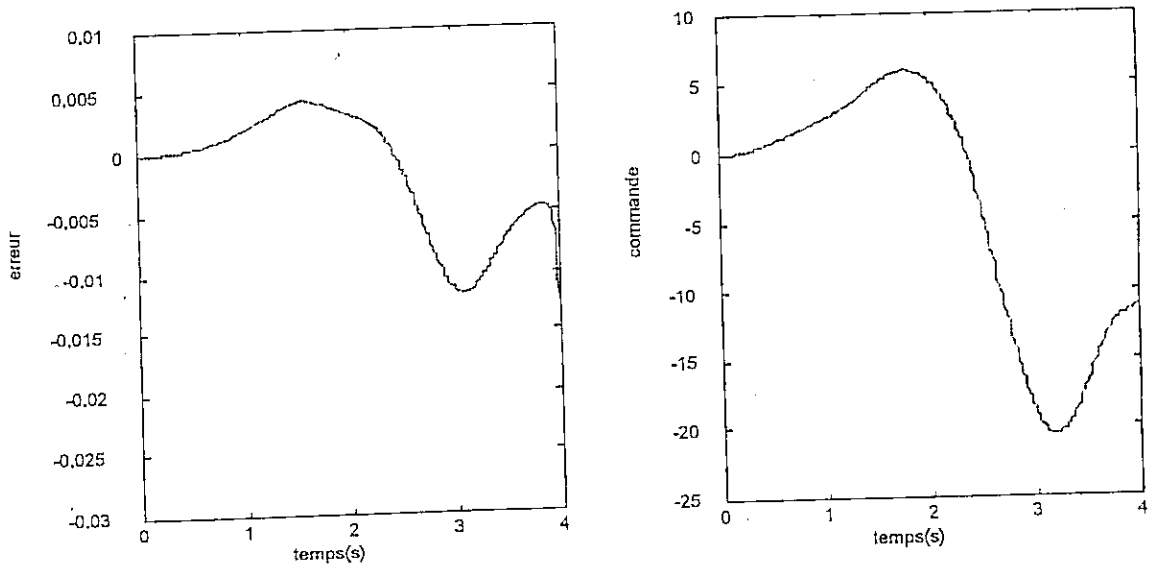
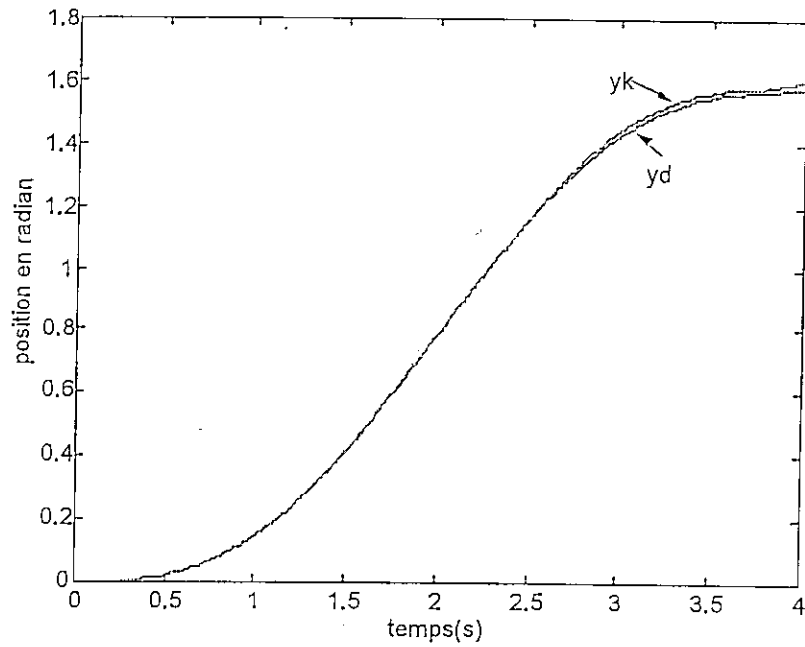
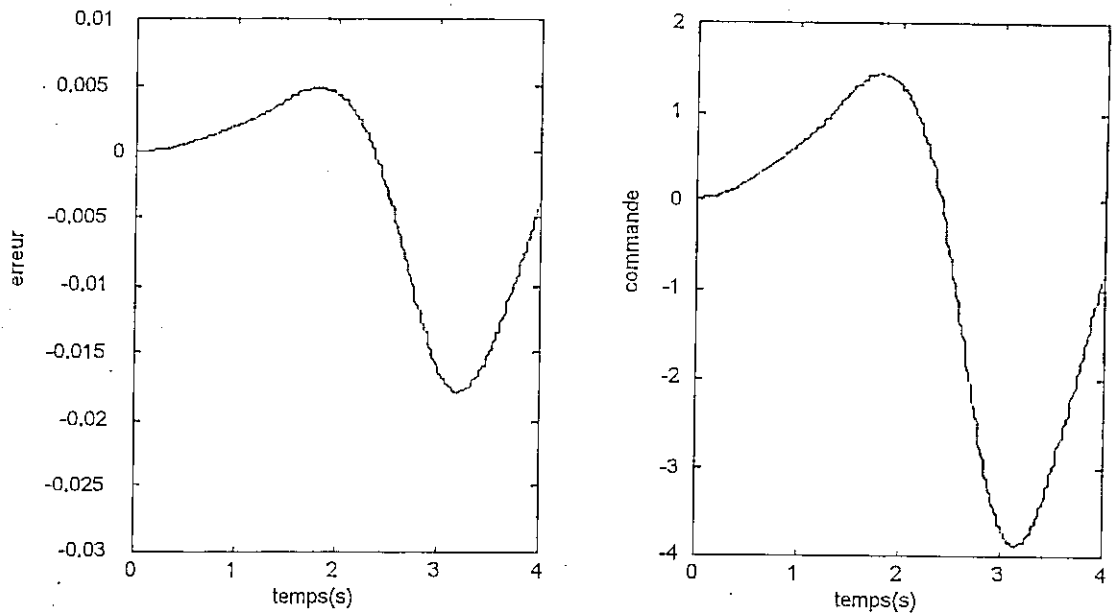


Fig.4.3b Erreur et commande de la 1ère articulation pour $k=5$

Fig. 4.6a position de la 2ème articulation pour $k=1$ Fig.4.6b Erreur et commande de la 2ème articulation pour $k=1$

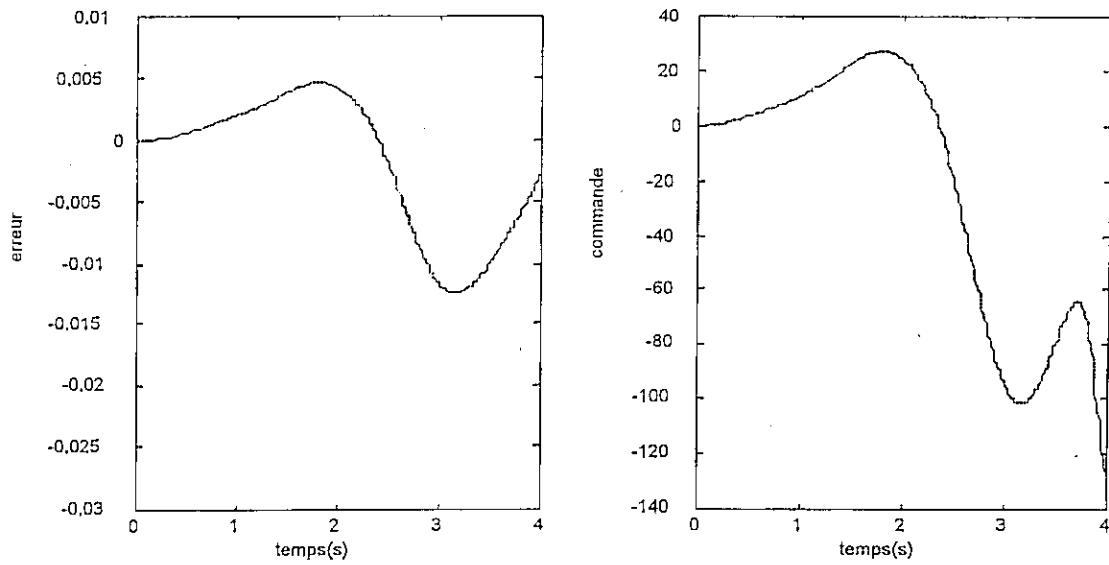


Fig.4.8a Erreur et commande de la 2ème articulation pour k=20

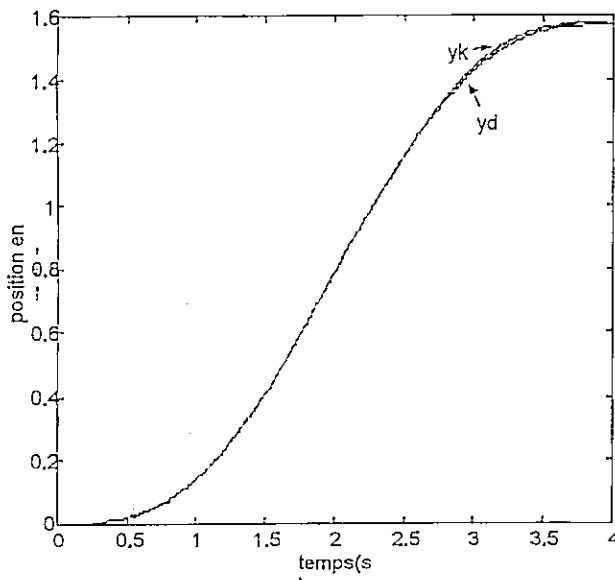


Fig. 4.8b Position de la 2ème articulation pour k=20

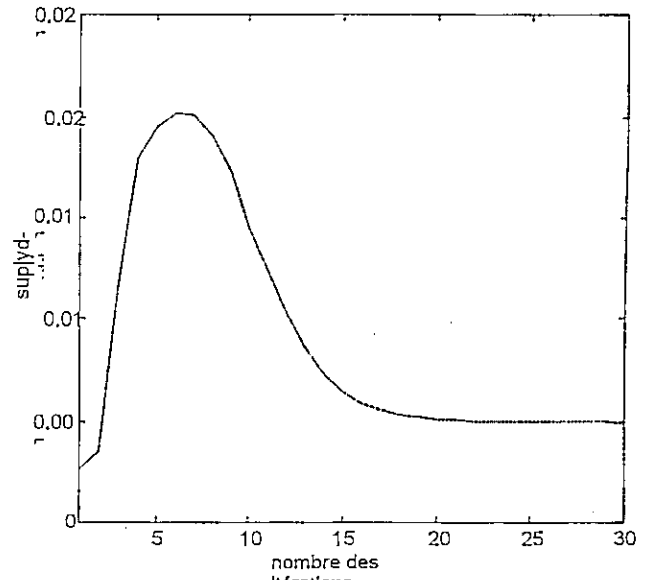


Fig.4.9 Sup |yd-yk|

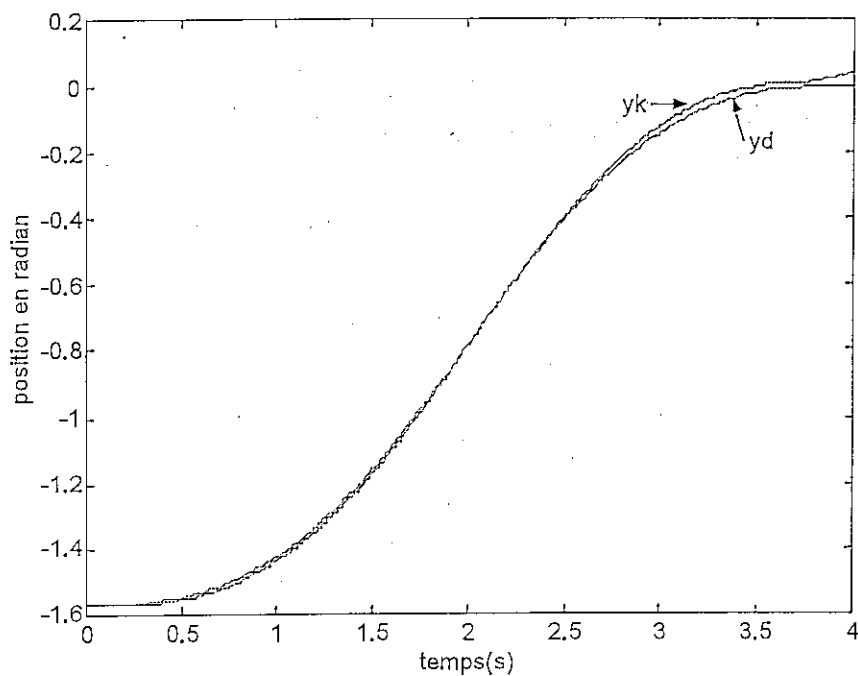


Fig. 4.11a Position de la 3^{ème} articulation pour $k=5$

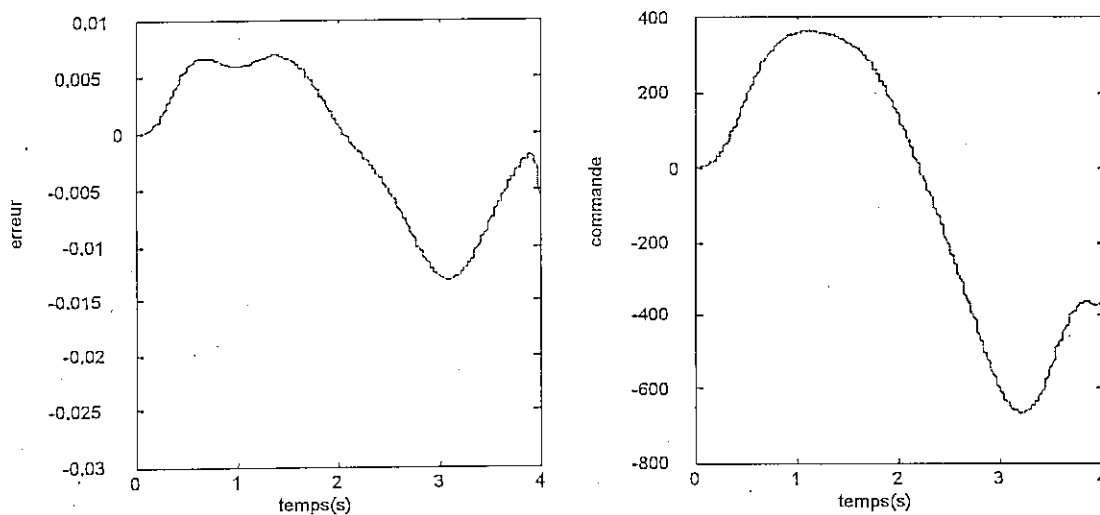


Fig.4.11b Erreur et commande de la 3^{ème} articulation pour $k=5$

Remarque : Nous remarquons qu'à travers les itérations, la commande devienne grande.

4.5 Conclusion

Une loi de commande par apprentissage itératif est appliquée au robot manipulateur de type PUMA 560, nous avons développé un type de calcul pour tirer les conditions suffisantes pour garantir la convergence. Il est montré que sous ces conditions, la sortie du système tend vers la trajectoire désirée avec une erreur de poursuite uniformément bornée. les résultats de simulations prouvent clairement l'amélioration des performances du système.

Conclusion générale

L'objectif de ce travail a été de synthétiser des lois de commande par apprentissage itératif pour des systèmes répétitifs, puis appliquer cette technique de commande au robot manipulateur de type PUMA 560. Notre choix de la commande par apprentissage itératif a été motivé par la répétition des tâches exécutées par le robot manipulateur. Cette méthode de commande consiste à trouver un mécanisme itératif adéquat permettant d'apprendre les erreurs des cycles précédents et exécuter progressivement mieux le nouveaux cycle, et par la suite améliorer la poursuite à travers les itérations.

Deux lois de commande par apprentissage itératif ont été présenté pour des systèmes non-linéaires avec une loi itérative pour l'état initial, la loi de commande par apprentissage itératif autonome "*off-line*" qui utilise l'erreur précédente, et la loi de commande par apprentissage itératif connectée "*on-line*" qui utilise l'erreur présente. Nous avons proposé une loi de commande par apprentissage itératif en tenant compte de l'erreur précédente et l'erreur présente, pour résoudre le problème de la poursuite pour une classe de systèmes non-linéaires à temps continu. Ainsi l'erreur de poursuite converge vers zéro. Les résultats de simulation ont prouvé l'efficacité de cette loi de commande et l'amélioration des performances du système. Pour les systèmes non-linéaires perturbés, les limites pour les erreurs de poursuite sont obtenues, et dépendent des perturbations du système et des erreurs initiales.

Comme le degré relatif a un grand intérêt dans la commande des systèmes non-linéaires, notamment dans la commande par apprentissage itératif, dans ce travail, nous avons proposé une loi de commande par apprentissage itératif avec observateur d'état pour des systèmes non-linéaires avec degré relatif arbitraire, avec l'évaluation de l'erreur de poursuite. Cette loi de commande nous a permis d'améliorer les performances du système.

La commande par apprentissage itératif a été appliquée avec succès sur les robots manipulateur, notamment, au robot manipulateur de 2 degré de liberté, au robot industriel

Annexe

A.1 Présentation du robot PUMA 560

Dans cette étude, nous avons utilisé un robot de type PUMA 560. On a considéré uniquement les trois premières articulations rotationnelles θ_1, θ_2 et θ_3 présentées sur la figure.

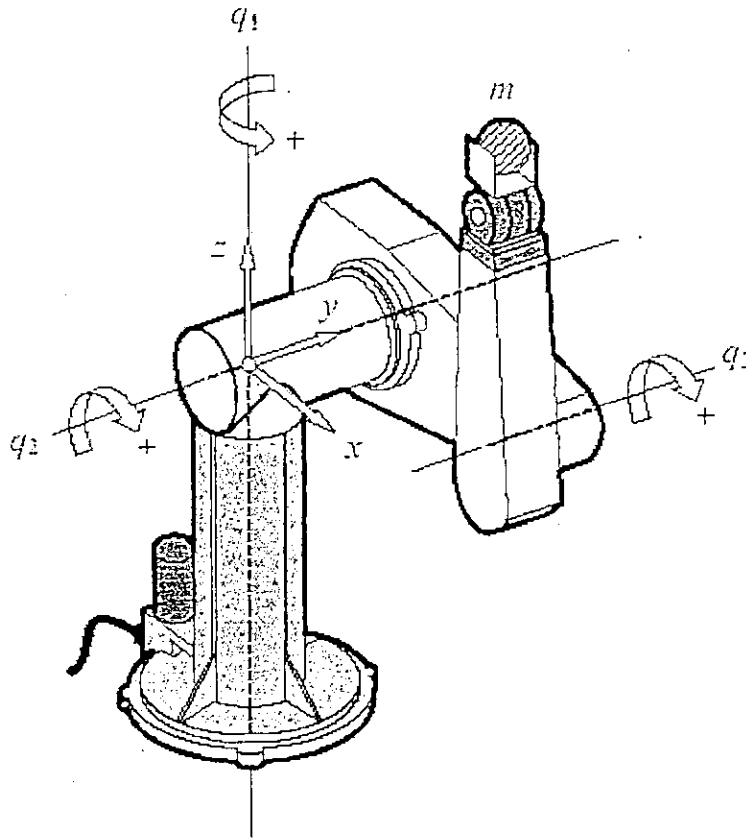


Fig.A.1 Présentation du robot PUMA 560

A.2 Modèle dynamique du robot manipulateur 560

Le modèle dynamique de ce robot est donné par l'équation suivante :

$$M(\theta)\ddot{\theta} + V_m(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau$$

avec

$$M(\theta) = \begin{bmatrix} I_1 + I_2 c_{23}^2 + I_3 c_2^2 + I_4 c_2 c_{23} & I_5 s_{23} + I_6 s_2 & I_5 s_{23} \\ I_5 s_{23} + I_6 s_2 & I_7 + I_4 c_3 & I_8 + 0.5 I_4 c_3 \\ I_5 s_{23} & I_8 + 0.5 I_4 c_3 & I_9 \end{bmatrix}$$

Effet de la masse de la charge

La masse de la charge a une importance capitale lors des simulations sur ordinateur, et pour modéliser son effet sur la dynamique du système, l'approche de la matrice jacobienne est utilisée.

L'expression de la matrice jacobienne est la suivante:

$$J(\theta) = \begin{bmatrix} -s_1(l_2c_2 + l_3c_{23}) - d_2c_1 & -c_1(l_2s_2 + l_3s_{23}) & -c_1(l_3s_{23}) \\ c_1(l_2c_2 + l_3c_{23}) - d_2s_1 & -s_1(l_2s_2 + l_3s_{23}) & -s_1(l_3s_{23}) \\ 0 & -(l_2c_2 + l_3c_{23}) & -(l_3c_{23}) \end{bmatrix}$$

Il s'ensuit que le couple additif dû à l'effet de la masse est donné par:

$$\tau_{masse} = m \cdot J^T (J \cdot \ddot{q} + \dot{J}(q, \dot{q}) \cdot \dot{q} + g)$$

avec m : masse de la charge.

\dot{J} : la matrice jacobienne dérivée par rapport au temps.

$$g = [0 \quad 0 \quad 9.81]^T :$$

Paramètres réels du robot PUMA 560

Parties	Poids(kg)
Liaison 1	
Liaison 2	17.40
Liaison 3	4.80
Liaison 4*	0.82
Liaison 5*	0.34
Liaison 6*	0.09
Poignet	1.24

* valeurs avec tolérance (kilogrammes $\pm 25\%$).

$$\triangleright m_t = m_4 + m_5 + m_6$$

Tableau A.1 : Poids des différentes liaisons (kilogrammes : $\pm 0.01=1\%$).

Paramètres géométriques :

$$\triangleright d_2 = 194.09mm$$

$$\triangleright l_2 = 431.8mm$$

$$\triangleright l_3 = 433.07mm$$

$$\begin{aligned}
n_{21} &\approx -0.5(-2.76s_2c_2 + 7.44 \times 10^{-1}c_{223} + 0.60s_2c_3 \\
&\quad - 2.13 \times 10^{-2}(1 - 2s_2s_3))\dot{q}_1 \\
n_{22} &\approx (2.20 \times 10^{-2}s_3 + 7.44 \times 10^{-1}c_3)\dot{q}_3 \\
n_{23} &\approx 0.5(2.20 \times 10^{-2}s_3 + 7.44 \times 10^{-1}c_3)\dot{q}_3 \\
n_{31} &\approx -0.5(7.44 \times 10^{-1}c_2c_{23} + 0.60s_2c_3 \\
&\quad + 2.20 \times 10^{-2}c_2s_{23} - 2.13 \times 10^{-2}(1 - 2s_2s_3))\dot{q}_1 \\
n_{32} &\approx -0.5(2.20 \times 10^{-2}s_3 + 7.44 \times 10^{-1}c_3)\dot{q}_3 \\
n_{33} &= 0
\end{aligned}$$

➤ Les éléments de $G(q)$:

$$\begin{aligned}
g_1 &= 0 \\
g_2 &\approx -37.2c_2 - 8.4s_{23} + 1.02s_2 \\
g_3 &\approx -8.4s_{23} + 0.25c_{23}
\end{aligned}$$

➤ Les éléments de la matrice Jacobienne :

$$\begin{aligned}
j_{11} &= -s_1(a_2c_2 + a_3c_{23}) - (d_2 + d_3)c_1 - d_4s_1s_{23} \\
j_{12} &= -c_1(a_2s_2 + a_3s_{23}) + d_4c_1c_{23} \\
j_{13} &= -a_3c_1s_{23} + d_4c_1c_{23} \\
j_{21} &= c_1(a_2c_2 + a_3c_{23}) - (d_2 + d_3)c_1 + d_4c_1s_{23} \\
j_{22} &= -s_1(a_2s_2 + a_3s_{23}) + d_4s_1c_{23} \\
j_{23} &= -a_3s_1s_{23} + d_4s_1c_{23} \\
j_{31} &= 0 \\
j_{32} &= -(a_2c_2 + a_3c_{23}) + d_4s_{23} \\
j_{33} &= -a_3c_{23} - d_4s_{23}
\end{aligned}$$

avec $a_2 = 0.4319m$, $a_3 = -0.0203m$, $d_2 = 0.2435m$, $d_3 = -0.0934m$, $d_4 = 0.4331m$.

Liste des principaux symboles

y_d	Sortie désirée du système.
x_d	Etat désiré du système.
u_d	Commande désirée du système.
k	Nombre d'itérations.
y_k	Sortie réelle du système.
x_k	Etat du système à la $k^{\text{ième}}$ itération.
u_k	Commande du système en feedforward à la $k^{\text{ième}}$ itération.
τ_k	Commande du système en feedback à la $k^{\text{ième}}$ itération.
\tilde{x}_k	Etat observé du système à la $k^{\text{ième}}$ itération.
\tilde{y}_k	Sortie observée du système à la $k^{\text{ième}}$ itération.
e_k	Erreur de poursuite à la $k^{\text{ième}}$ itération .
z_k	Erreur de poursuite filtrée à la $k^{\text{ième}}$ itération .
\tilde{u}	Erreur de commande.
μ	Degré relatif du système.
μ_q	Degré relatif de la $q^{\text{ième}}$ sortie d'un système MIMO.
$L_f g(x)$	Dérivée de Lie de $g(x)$ par rapport à f .
$h_x(x_k)$	Jacobien de la fonction $h(x)$
ϕ	Fonction du vecteur estimé.
λ	Nombre positif (largement grand).
$\ \cdot\ $	Désigne la Norme Euclidienne.

Références

- [1] Uchiyama.M: "Formulation of high-speed motion pattern of a mechanical arm by trial", *Trans.soc.Instrum.contr.eng.* (Japon), 1978, Vol. 14, No. 16, pp. 706-712.
- [2] Arimoto.S, Kwamura.S, Miyazaki.F : "Bettering operation of robots by learning", *Journal of robotic systems*, Vol.1, No. 2, 1984, pp. 123-140.
- [3] Casalino.G, et Bartolini.G : " A learning procedure of the control of mouvements of robotics manipulators", *In IASTED Symposium on robotics and Automation, Amsterdam, The Netherlands*, 1984, pp. 108-111.
- [4] Craig.J : "Adaptive control of manipulators through repeated trial", Proc. American Control Conference, San Diego, CA, June 1984, pp. 1566-1573
- [5] Jang.T, Choi.C, and Ahn.H : Iterative learning control in feedback systems. *Automatica*, 1995, Vol. 31, No. 2, pp. 243-248.
- [6] Chen.Y, Wen.C, and Sun.M : A robust high-order of P-type iterative learning controller using current iterative tracking error. *International journal of control*, 1997, Vol. 68, No.2, pp. 331-342.
- [7] Lee.H, and Bien.Z : Study on robustness of iterative learning control with non-zero initial error. *International journal of control*, 1996, Vol. 64, No.3, pp. 345-359.
- [8] Yan.X.G, and Zhang. S. Y : Iterative learning control for a class of nonlinear similar composit systems. *Control & Decision*, 1998, Vol. 14, pp. 361-363.
- [9] Lee.H, and Bien.Z : Initial condition problem of learning control. *IEEE Proc.D*, Nov. 1991, Vol. 138, No. 6, pp. 525-528.
- [10] Hauser,J.E : Learning control for a class of nonlinear systems. *In proceedings of the 26th IEEE Conf. Decision Control, Los Angeles, CA, USA, Dec. 1987*, pp. 859-60.
- [11] Ahn.H, Choi.C, Kim.K : "Iterative learning control for a class of nonlinear systems", *Automatica*, 1993, Vol. 29, No. 6, pp. 1575-1578.

- [23] Chen Y, Wen C, Gong Z, et Sun M. "An iterative learning controller with initial state learning", *IEEE Transaction on Automatic Control*, 1999, Vol. 44, No. 2, pp. 371-375.
- [24] Tayebi.A : Observer-based iterative learning control for a class of time-varying nonlinear system. *IEEE Transaction on Circuits and Systems I*, 2002.
- [25] Sun.D, Wang.D : Anticipatory iterative learning control for nonlinear systems with arbitrary relative degree. *IEE Transaction on Automatic control*, May 2001, Vol.46, No.5, pp. 783-788.
- [26] Isidori.A : Nonlinear control systems. *Berlin, Germany, Springer-Verlag, 1995*.
- [27] Rajamani.R : Observers for Lipschitz nonlinear systems. *IEEE Transaction on Automatic control*, 1998, Vol.43, No. 3, pp. 397-401.
- [28] Arimoto.S : Learning control theory for robotic motion. *Int. j. of Adaptive Control and Signal Processing*, 1990, 4, pp. 543-564.
- [29] Lewis.F.L, Abdellah.C.T, and Dawson.D.M : Control of robot manipulators. *Macmillan, New York, 1993*.
- [30] Dixon.W.E, Zergeroglu.E, Dawson.D.M, and Costic.B.T : Repetitive learning control: A Lyapunov-based approach. *Proceeding of the 2001 IEEE International Conference on Control Application, September 5-7, 2001, Mexico*, pp. 530-535.