

Un logiciel de commande non linéaire échantillonné

Valentin TANASA¹, Fernando TIEFENSEE²

¹Politehnica University of Bucarest,
Splaiul Independentei 313, 060042 Bucarest, Roumanie
valentin.tanasa@cti.pub.ro

²Laboratoire des Signaux et Systèmes, CNRS-Supelec
Plateau de Moulon, 91190 Gif-sur-Yvette, France
tiefensee@lss.supelec.fr

Résumé— Il s’agit de proposer un logiciel d’aide à la conception et au calcul de lois de commande non linéaire échantillonnées. La conception repose sur une méthode de reproduction, aux instants d’échantillonnage, de dynamiques cibles continues, dites idéales, développées dans plusieurs contextes [4], [5], [6] et [8]. Les performances qualitatives et en termes de faisabilité (temps de calcul) sont illustrées pour la stabilisation d’un corps rigide.

Mots-clés— Systèmes non linéaires échantillonnés, Commande non linéaire échantillonnée, logiciel de commande.

I. INTRODUCTION

Au cours des dernières décennies les exigences en performances de plus en plus fortes pour les systèmes de commande informatisés, les progrès constants du numérique en termes de coût, précision et rapidité, font de la recherche sur les systèmes échantillonnés une des thématiques phares de la communauté scientifique internationale. En sachant que, pour la classe de systèmes échantillonnés, deux opérations interagissent, échantillonnage des mesures d’un côté et reconstruction de signaux d’entrée continus du processus de l’autre, une question centrale est posée : est-il possible et comment peut-on préserver les performances d’un schéma continu par un schéma de commande numérique ?

Cette question a été largement traitée dans la littérature, en particulier dans un contexte linéaire où les solutions sont bien établies (voir par exemple [3], [1]). Toutefois, dans un contexte non linéaire, des difficultés supplémentaires font que la majorité des travaux disponibles utilisent des modèles échantillonnés au premier ordre - *schéma d’Euler*.

Le présent article repose sur la méthode de reproduction entrée/sortie sous échantillonnage ([4], [5], [6]). Il s’agit de concevoir une loi de commande constante par morceaux de façon à reproduire, sous échantillonnage, une trajectoire cible du système continu, dite solution idéale. Pour le calcul explicite de la commande, des méthodes d’approximation sont proposées. La solution est comparée à des solutions issues de méthodes classiques d’implantation et ses performances sont mises en évidence dans la première partie de l’article.

Un logiciel d’aide à la conception et au calcul, conçu à partir de méthodes algébriques formelles et d’algorithmes numériques stables de type Runge-Kutta [2], est présenté. Le calcul de la loi de commande se fait en deux temps. La conception exacte formelle et par approximations polynômiales de la loi de commande sont faites *hors ligne* avec le degré d’approximation souhaité par le concepteur.

Le calcul explicite de la solution polynômiale, à partir des mesures faites sur le processus réel, est réalisé *en ligne*. Le temps de calcul de cette évaluation numérique est donc négligeable et le retard possible compensé par l’aspect prédictif de la solution analytique, un avantage explicite de la méthode employée. Le logiciel propose également un choix de simulations comparatives sur les diverses variables du procédé, état, sortie mesurée ou non, entrée, fonctions significatives (Lyapunov, énergie, Hamiltonien), face au degré d’approximation choisi par le concepteur. Le logiciel développe des tests représentatifs de l’amélioration des performances face au degré d’approximation du contrôleur, de l’amplitude de la période d’échantillonnage et du gain du contrôleur.

L’article est organisé comme suit : la section II rappelle les méthodes formelles de calcul du modèle échantillonné exact d’un système continu. Dans la section III, la méthode de reproduction entrée/sortie sous échantillonnage est présentée et comparée d’un point de vue théorique à des méthodes classiques d’implantation numérique. Le logiciel est décrit dans la section IV. La stabilisation d’un corps rigide en rotation autour de son centre de masse illustre les performances de la commande dans la section V.

II. RAPPELS THÉORIQUES

Ces rappels proviennent de [4], [5] et [6].

A. La série de Lie exponentielle

Etant donné une fonction analytique $\lambda : \mathbb{R}^n \rightarrow \mathbb{R}$ et un champs de vecteur analytique θ sur \mathbb{R}^n , l’opérateur différentiel associé L_θ agit sur λ de sorte que $L_\theta \lambda(x) := \frac{\partial \lambda}{\partial x} \theta(x)$ exprime la dérivée de Lie de λ le long de θ , évaluée en x . L’action itérée p -fois de cet opérateur est définie $L_\theta^p := L_\theta \circ \dots \circ L_\theta$ pour $p \geq 0$, soit $L_\theta^p \lambda := \frac{\partial L_\theta^{p-1} \lambda}{\partial x} \theta$, avec $L_\theta^0 = 1$, l’opérateur identité. On note e^{L_θ} , indifféremment e^θ , la série exponentielle ou série de Lie associée à L_θ , $e^{L_\theta} = 1 + \sum_{p \geq 1} \frac{L_\theta^p}{p!}$. Etant donné un champs de vecteurs σ , le crochet de Lie est défini par $L_{[\theta, \sigma]} := L_\theta \circ L_\sigma - L_\sigma \circ L_\theta$. L’évaluation en x d’une fonctionnelle est notée indifféremment “ (x) ” ou “ $|_x$ ” ; l’évaluation d’un opérateur ou d’une composition d’opérateurs appliqué à la fonction identité $L_\theta Id|_x$, est notée $\theta(x)$ sans ambiguïté ; de même $L_\theta Id|_{e^\sigma(x)} := \theta(e^\sigma x)$.

B. Modèle sous échantillonnage

Soit un système dynamique en temps continu à entrée-affine, Σ_c :

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \quad (1)$$

où f et g sont champs de vecteurs réels de dimension n et le signal d'entrée $u(t)$, une fonction régulière à valeurs réelles. Son équivalent sous échantillonnage, noté Σ^δ , soumis à une commande $u(t)$ constante sur des intervalles d'amplitude δ , c.à.d. $u(t + \tau) = u(t) = u_k$, pour $0 \leq \tau < \delta$, $t = k\delta$, $k \geq 0$, doit, pour une même condition initiale x_0 , avoir un comportement entrée-état identique à Σ_c , aux instants $t = k\delta$, $k \geq 0$. La solution traditionnelle est décrite par une équation aux différences non linéaire, paramétrée par la période d'échantillonnage :

$$x_{k+1} = F^\delta(x_k, u_k) = e^{\delta(L_f + u_k L_g)} x_k \quad (2)$$

où $e^{\delta(L_f + u_k L_g)}$, la série exponentielle de Lie associée au champs de vecteur $f + u_k g$, peut être calculée pour $\delta \in]0, T^*]$, T^* un intervalle de temps suffisamment petit, comme suit :

$$e^{\delta(L_f + u_k L_g)} = 1 + \delta(L_f + u_k L_g) + \frac{\delta^2}{2!} (L_f^2 + u_k(L_f L_g + L_g L_f) + u_k^2 L_g^2) + \dots + \frac{\delta^P}{P!} L_{f+u_k g}^P + \dots$$

Un modèle échantillonné approché à l'ordre P correspond à une approximation de la série asymptotique à l'ordre P en δ (erreur en $O(\delta^{P+1})$). Chaque coefficient de δ^P dans la série (2) n'est autre que la dérivée temporelle d'ordre P de l'état $x(t)$ de (1) soumis à la commande constante par morceaux u_k et calculée à l'instant $t = k\delta$ [7].

Remarque: Le modèle échantillonné approché à l'ordre 1 en δ correspond au schéma classique de discretization d'Euler, c.à.d. à l'approximation suivante de la dérivée $\dot{x}(t) = \lim_{\delta \rightarrow 0} \frac{x(t+\delta) - x(t)}{\delta}$, soit $x_{k+1} = x_k + \delta(f(x_k) + u_k g(x_k))$, pour $\delta \in]0; T^*]$, T^* suffisamment petit. \triangleleft

III. REPRODUCTION ENTRÉE/SORTIE PAR COMMANDES CONSTANTES PAR MORCEAUX

Etant donnée $h : \mathbb{R}^n \rightarrow \mathbb{R}$, une sortie ou une trajectoire cible associée au système (1), comment concevoir une commande constante par morceaux u_k permettant de la reproduire aux instants d'échantillonnage? Il est bien connu que la solution est liée au degré relatif du système défini pour (1) avec la sortie $h(x(t))$ comme le plus petit entier tel que :

$$L_g L_f^i h(x) = 0 \text{ pour } i < r - 1; \quad L_g L_f^{r-1} h(x) \neq 0$$

La sortie échantillonnée associée à (2) vérifie, aux instants $t = k\delta$:

$$y_{k+1} = h(x(t = (k+1)\delta)) = e^{\delta(L_f + u_k L_g)} h(x(k));$$

avec $L_g L_f^{r-1} h(x_k) \neq 0$ et le lien entrée/sortie échantillonné possède toujours un degré relatif égale à 1.

Remarque: Le concept de degré relatif, introduit dans le contexte non linéaire, correspond à la notion d'excès de pôles sur les zéros de la fonction transfert en linéaire. \triangleleft

Etant donnée une commande $u_c(x(t))$, il s'agit de calculer une loi de commande u_k telle que la sortie

du système échantillonné bouclé soit égale, aux instants d'échantillonnage, à la sortie du système en temps continu soumis à $u_c(x(t))$ pour une même initialisation $x(k\delta) = x_k$, c.à.d. vérifiant l'égalité :

$$e^{\delta(L_f + L_{u_c g})} h(x_k) = e^{\delta(L_f + u_k L_g)} h(x_k). \quad (3)$$

Théorème 1 : [6] Etant donné un système linéaire analytique en temps continu Σ_c associé à une sortie $h(x)$ de degré relatif r , et une loi de commande régulière $u_c(x(t)) \in \mathbb{R}$; alors pour tout $\delta \in]0, T^*]$, T^* suffisamment petit, il existe une loi de bouclage constante par morceaux $u_d^\delta(x) \in \mathbb{R}$, de telle sorte que le système bouclé sous échantillonnage Σ^δ reproduise, pas à pas, le comportement entrée/sortie de Σ_c .

Preuve 1 : Récrivant (3) avec u_k défini comme

$$u_k = u_d^\delta(x_k) = u_{d0} + \sum_{i \geq 1} \frac{\delta^i}{(i+1)!} u_{di} \quad (4)$$

et $x(k\delta) = x_k$, la condition du degré relatif $L_g L_f^{r-1} h(x) \neq 0$, assure l'existence de u_d comme une série asymptotique en puissances de δ autour de u_{d0} . Les termes successifs de la loi de commande (4) sont calculés à partir de la comparaison terme à terme de l'égalité des séries formelles (3).

En supposant $r = 1$, une commande approchée du deuxième ordre :

$$u_d^{\delta 2}(x_k) = u_{d0} + \frac{\delta}{2!} u_{d1} + \frac{\delta^2}{3!} u_{d2}, \quad (5)$$

avec :

$$u_{d0} = u_c(x(t))|_{t=k\delta} \quad (6)$$

$$u_{d1} = \dot{u}_c(t)|_{t=k\delta} \quad (7)$$

$$u_{d2} = \left(\ddot{u}_c(t) + \frac{1}{2} \dot{u}_c(t) \frac{L_{[f,g]} h(x(t))}{L_g h(x(t))} \right) \Big|_{t=k\delta}, \quad (8)$$

assure l'égalité des séries (3) jusqu'à l'ordre 3 (erreur en $O(\delta^4)$).

Remarque: Le Théorème 1 est énoncé pour les systèmes mono-entrée/mono-sortie, mais l'extension aux systèmes multi-entrée multi/sortie carrés est immédiate. \triangleleft

Remarque: Dans le cas reproduction entrée/état traité en [6], l'égalité de n séries doit être assurée et on fait appel à des techniques d'échantillonnage multi-échelles [4]. \triangleleft

A. A propos d'autres méthodes d'implantation

Plusieurs techniques d'implantation numérique d'une commande conçue en temps continu, ne prenant pas en considération le modèle échantillonné équivalent, sont possibles. Elle sont analysées ci-dessous en comparant leurs expressions analytiques aux termes correcteurs décrits en (5) de la solution exacte proposée dans le Théorème 1.

A.1 Reproduction par bloqueur d'ordre zéro

La méthode la plus immédiate est celle du *bloqueur d'ordre zéro - ZOH - émulation de la commande continue*. Elle consiste à bloquer la commande conçue en temps continu pendant les intervalles d'échantillonnage; c.à.d. $u_k = u_c(x(t))|_{t=k\delta} = u_{d0}$. On retrouve donc uniquement le terme "dominant" (6). La validité d'une commande émulée est strictement limitée à des périodes d'échantillonnage d'amplitudes très faibles, et des dynamiques très lentes.

A.2 Reproduction par bloqueur d'ordre supérieur

La commande $u_c(x(t))$ peut être générée par un bloqueur d'ordre supérieur comme suit :

$$u_{sup}(t) = u_c(x(t))|_{t=k\delta} + \sum_{i \geq 1} \frac{(t - k\delta)^i}{i!} \frac{d^i u_c(x(t))}{dt^i} |_{t=k\delta}. \quad (9)$$

La commande u_{sup} est conçue de façon à reproduire la trajectoire de $u_c(x(t))$ au travers le calcul de toutes les dérivées temporelles, difficile en pratique. Des solutions approchées peuvent être adoptées, soit au deuxième ordre de (9) :

$$u_{sup}^{t2} = u_c(x(t))|_{t=k\delta} + (t - k\delta) \dot{u}_c|_{t=k\delta} + \frac{(t - k\delta)^2}{2!} \ddot{u}_c|_{t=k\delta},$$

qui diffère de (5) dès le premier terme correcteur ; l'égalité (3), avec $u_k = u_{sup}^{t2}$, n'est vérifiée que jusqu'à l'ordre δ (performance identique à celle de la commande émulée u_{d0}). Ceci confirme bien la différence conceptuelle des deux approches, l'une orientée à la reproduction d'une évolution désignée en boucle fermée améliorant aussi le comportement entre les instants d'échantillonnage, l'autre orientée à reproduire le signal continu.

A.3 Intégration et valeur moyenne de $u_c(x(t))$

Implanter la valeur moyenne de la commande $u_c(x(t))$ sur un intervalle d'amplitude la période d'échantillonnage représente une approche prédictive ; cette approche conduit au calcul suivant

$$\begin{aligned} u_{moy} &:= \frac{1}{\delta} \int_{k\delta}^{(k+1)\delta} u_c(x(t)) dt \\ &= u_c(x(t)) + \sum_{i \geq 1} \frac{\delta^i}{(i+1)!} \frac{d^i u_c(x(t))}{dt} \Big|_{t=k\delta} \end{aligned}$$

et son approximation du deuxième ordre :

$$u_{moy}^{\delta 2} = u_c(x(t)) + \frac{\delta}{2!} \dot{u}_c(x(t)) + \frac{\delta^2}{3!} \ddot{u}_c(x(t)) \Big|_{t=k\delta}.$$

On constate qu'il y a coïncidence des termes avec (5) jusqu'au deuxième ordre, conférant à l'approche ici proposée un aspect prédictif.

A.4 Calcul de $u_c(x(t))$ à partir de l'état estimé au temps $t + \tau$

Afin de tenir compte d'un retard possible de calcul, l'implantation de la commande $u_c(x(t))$ à l'instant $t = k\delta$ peut être faite sur la base de l'état estimé à l'instant $t = k\delta + \tau$. On obtient :

$$\begin{aligned} u_{est}^\tau(k) &= u_c(\hat{x}(k\delta + \tau)) = e^{\tau(f+u_{cg})} u_c(x_k) \\ &= u_{d0} + \sum_{i \geq 1} \frac{\tau^i}{i!} L_{(f+u_{cg})}^{(i)} u_c(x_k). \end{aligned}$$

Si l'on pose $\tau = \frac{\delta}{2}$, on obtient l'approximation du deuxième ordre suivante :

$$u_{est}^{\delta 2} = u_{d0} + \left(\frac{\delta}{2!} \dot{u}_c + \frac{\delta^2}{3!} \ddot{u}_c \right) \Big|_{x_k}.$$

On constate qu'il y a coïncidence des termes avec (5) jusqu'au deuxième ordre, conférant à l'approche ici proposée de la robustesse quant au retard de calcul.

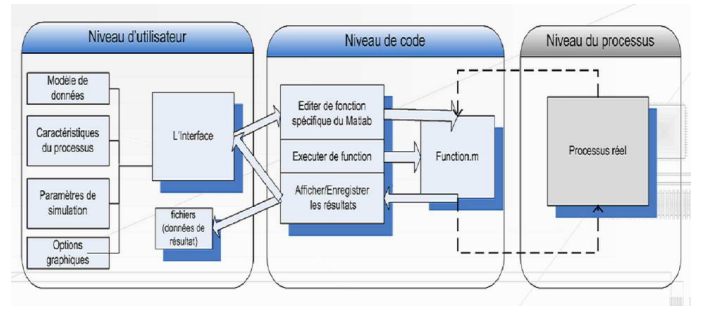


Fig. 1. La structure du logiciel

En conclusion, l'approche proposée tient compte dans sa conception des aspects relatifs au comportement entre les instants d'échantillonnage, possède des propriétés de prédiction et prend en compte un possible retard de calcul.

IV. LE LOGICIEL

Le logiciel -SimNLSys- a pour objectifs :

- l'intégration du modèle continu et la conception des systèmes discrets équivalents ;
- la construction de lois de commande constantes par morceaux assurant un comportement entrée/sortie désiré - *comportement cible* ;
- la simulation de tests représentatifs de l'influence du degré d'approximation.

La structure de l'application, illustrée dans la figure 1, est divisée en trois niveaux :

1. *Le niveau utilisateur*, dédié à l'entrée des données à la présentation des résultats ;
2. *Le niveau code*, ayant un niveau amont dédié aux algorithmes de conception ; un niveau de simulation ; un niveau aval, dédié à gérer les résultats ;
3. *Le niveau processus*, permettant la communication avec un processus réel ou avec d'autres logiciels de simulation (*Simulink*, par exemple). Ce niveau reste en cours de développement.

A. Fonctionnement du logiciel

L'interface de communication entre l'utilisateur et le logiciel est implantée à partir d'une fenêtre principale, permettant l'entrée des données du modèle, de la simulation et de l'analyse. Il faut remarquer que toutes les fonctions qui sont introduites doivent respecter les syntaxes imposées par Matlab, les états sont exprimés par x_1, \dots, x_n et le temps par t . Cette fenêtre principale comporte les quatre fonctions du logiciel :

A.1 Connaissance du modèle

Le logiciel étant conçu pour la simulation des systèmes non-linéaires à entrée-affine, comme en (1), l'utilisateur doit entrer les expressions analytiques de l'évolution libre $f(x) \in \mathbb{R}^n$, de la dynamique forcée $g(x) \in \mathbb{R}^n$, de la sortie $h(x)$, $h : \mathbb{R}^n \rightarrow \mathbb{R}$ et les conditions initiales du système. En option, l'expression analytique d'une commande $u(x, t)$, pouvant avoir une dépendance temporelle, peut être aussi spécifiée par l'utilisateur (le logiciel simule aussi des systèmes en boucle-ouverte).

A.2 Outils de conception de la commande

Dans un contexte échantillonné, étant défini un schéma de commande en temps continu $\Sigma_c(f(x), g(x), h(x), u(x))$, après avoir vérifié le degré relatif du système, le logiciel calcule une loi de commande constante par morceaux du type (4), assurant la reproduction de la sortie $h(x)$.

Le degré d'approximation peut être choisi par l'utilisateur, ≤ 4 dans la version ici développée.

La conception de commandes à partir de critères de Lyapunov est aussi prise en compte par le logiciel. Des techniques telles que l'injection d'amortissement et la reproduction d'un certain critère énergétique sous échantillonnage (voir [8]), peuvent être calculées et simulées.

A.3 Simulation

Le logiciel simule les trois classes de comportements dynamiques non linéaires : temps continu, temps discret et sous échantillonnage.

Pour simuler le schéma en temps continu, on définit d'abord le pas d'intégration d du modèle. Ainsi, l'intégration du système (1) peut être effectuée soit par une méthode fonctionnelle, c.à.d.

$$x_{k+1} = x((k+1)d) = e^{d(f+u_c g)} x_k,$$

soit par la méthode numérique de *Dormand-Prince 5* à un pas fixe d (approche dérivée de la *formule* de Runge-Kutta, voir [2]).

Le schéma de commande échantillonné, caractérisé par un système en temps continu soumis à une loi de commande constante par morceaux, calculée à partir de données échantillonnées, met en jeu des dynamiques discrètes et continues. La simulation d'une telle classe de systèmes est effectuée par deux boucles de calcul : la première intègre le système en temps continu à un pas d et la deuxième l'échantillonne et impose une commande constante par morceaux d'amplitude δ .

Les paramètres d (pas d'intégration du système) et δ (pas d'échantillonnage) doivent être choisis par l'utilisateur avec certains critères selon la simulation désirée. Pour simuler un schéma de commande échantillonné il est nécessaire choisir $d \ll \delta$; pour un schéma de commande en temps continu on pose $d = \delta$ très petits par rapport aux dynamiques du système; et pour un schéma de commande purement en temps discret, on pose $d = \delta$ le pas de discrétisation.

A.4 Algorithme

L'algorithme utilisé est organisé comme suit :

```

INITIALISATION ( $x(0), u_d(x_k)$ )
for  $k = 1 : \delta : t_f$ 
  BLOCAGE  $u_d(x_k)$ 
  for  $i = k\delta : d : (k+1)\delta - d$ 
     $x(i+1)d = F^d(x(id), u(k\delta))$ 
  end for
  ECHANTILLONNAGE ( $x((k+1)\delta), h(x((k+1)\delta))$ )
end for

```

Dans l'étape d'initialisation, les conditions initiales sont chargées et le calcul de l'expression analytique de $u_d(x_k)$

est réalisé avec le degré d'approximation désiré. Dans la première boucle de simulation, concernant les opérations de blocage et d'échantillonnage à un pas δ , la valeur de $u_d(x_k)$ est actualisée selon la mesure des états au pas antérieur (ou selon les conditions initiales au premier pas) et ensuite injectée dans le procédé. Dans la deuxième boucle, le procédé soumis à une commande constante u_d est intégré sur un pas, à chaque pas d , selon l'approche choisie par l'utilisateur. Quand $i = (k+1)\delta$, l'algorithme quitte la deuxième boucle, les échantillons des états et de la sortie sont pris (dans l'étape d'*actualisation des mesures*) et le processus se répète jusqu'à $t = t_f$ (le paramètre t_f définit le temps total de simulation).

La conception de la commande consiste à égaliser terme à terme les séries (3) jusqu'à un certain ordre d'approximation. Son temps de réalisation dépend de la complexité du système. Pour éviter que le temps de calcul soit trop important par rapport à l'amplitude de δ , ce processus est placé dans l'étape d'*initialisation* et est limité à des approximations de quatrième ordre en δ . La conception de la commande est ainsi faite *hors-ligne*. Par contre l'actualisation de ces valeurs en fonction de l'évolution des états du système est faite *en ligne* et à chaque instant d'échantillonnage $t = k\delta$.

A.5 Les indicateurs de performance

Le logiciel permet d'imprimer les trajectoires des états, de la sortie et de la commande et enregistre ces données sur le disque sous la forme de vecteurs pour des manipulations supplémentaires sur Matlab.

Deux indicateurs de performances du schéma de commande échantillonné sont définis dans le cadre de la reproduction entrée/sortie.

Le premier est l'*erreur moyenne* entre la trajectoire cible en temps continu $h(x(t))$ et la trajectoire équivalente du système échantillonné, commandé par u_d pendant le temps de simulation. L'état du système continu soumis à u_c et l'état du système échantillonné soumis à u_d , étant notés respectivement par x_c et x_d , l'erreur est définie par la différence des carrés des sorties aux instants d'échantillonnage, c.à.d. :

$$E := \sqrt{\sum_{k=1}^n \frac{(h(x_c(t))|_{t=k\delta} - h(x_d(k\delta)))^2}{n}}. \quad (10)$$

où le produit $n\delta$ indique la durée du régime transitoire du système (l'erreur en régime permanent est toujours nulle).

Le deuxième indicateur est la *valeur maximale* de la période d'échantillonnage, δ_{max} , permettant de stabiliser la trajectoire $h(x_d)$ du système échantillonné sur la trajectoire cible $h(x_c)$ quand le temps va vers l'infini, c.à.d. $\delta_{max} = \{max(\delta), t.q. \lim_{t,k \rightarrow \infty} h(x_c(t)) - h(x_d(k\delta)) = 0\}$

Un outil *multi-simulation* permet d'analyser les performances des systèmes à partir des variations de l'ordre P d'approximation sur u_d , de l'amplitude de la période d'échantillonnage δ et du gain K_c du contrôleur. Des graphiques en trois dimensions comportant l'évolution de l'erreur E en fonction de δ et K_c pour un P fixe, c.à.d. $(E \times \delta \times K_c)|_{P=cte}$, sont générés automatiquement. Un tel outil permet la conception d'un contrôleur numérique pour

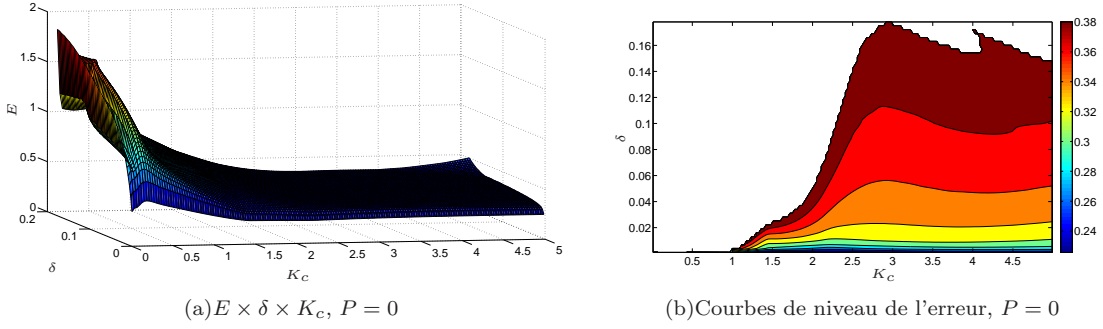


Fig. 2. Performances de la commande émulée, $P = 0$

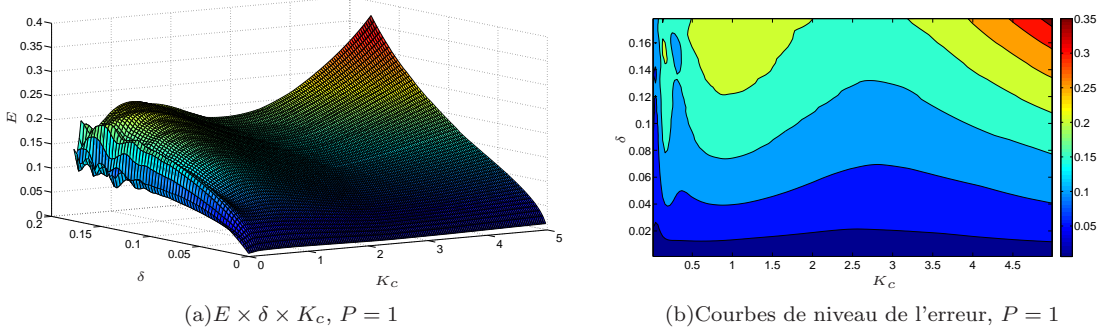


Fig. 3. Performances de la commande reproduction entrée/sortie, $P = 1$

une application réelle fixant les limites de δ , de K_c et de P pour une certaine erreur maximale acceptée sur la trajectoire à reproduire.

V. UN EXEMPLE SIMULÉ

L'objectif est d'utiliser les fonctions du logiciel pour mettre en évidence l'importance des paramètres δ , K_c et P dans le contrôleur échantillonné conçu pour la stabilisation d'un corps rigide en rotation autour de son centre de masse [9]. Son espace d'état est écrit en fonction des moments angulaires x_i , le long des trois principaux axes $i = 1, 2, 3$:

$$\begin{cases} \dot{x}_1(t) = x_2(t)x_3(t) \left(I_3^{-1} - I_2^{-1} \right) + b_1 u(t) \\ \dot{x}_2(t) = x_1(t)x_3(t) \left(I_1^{-1} - I_3^{-1} \right) + b_2 u(t) \\ \dot{x}_3(t) = x_1(t)x_2(t) \left(I_2^{-1} - I_1^{-1} \right) + b_3 u(t) \end{cases}$$

I_i , $i = 1, 2, 3$ sont les principaux moments d'inertie du système. La commande u représente un moment appliqué le long de l'axe $g = [b_1 \ b_2 \ b_3]^T$.

La stabilisation en temps continu est développée en [9]. Considérant l'énergie cinétique comme la fonction de Lyapunov, c.à.d. $V(x) = \frac{1}{2} \left(\sum_{i=1}^3 \frac{x_i^2}{I_i} \right)$, t.q. $V(0) = 0$, le système est stabilisé asymptotiquement son point d'équilibre stable $[x_{1e} \ x_{2e} \ x_{3e}]^T = [0 \ 0 \ 0]$ avec une commande :

$$u_c(x(t)) = -K_c L_g V(x(t)) = -K_c \sum_{i=1}^3 \frac{b_i}{I_i} x_i(t), \quad K_c > 0.$$

Pour la conception de la commande sous échantillonnage, on choisit la vitesse du système en boucle fermée comme trajectoire cible, c.à.d. $y(t) = \left(\sum_{i=1}^3 \frac{b_i}{I_i} x_i(t) \right)$.

Les paramètres utilisés pour la simulation sont : $b_1 = I_1 = 3$, $b_2 = I_2 = 2$ et $b_3 = I_3 = 1$; $x(0) = [5, 5, 5]$; $d = 0.5ms$; le temps de simulation est de $2s$. Les plages de

variation des paramètres du contrôleur sont $K_c \in [0.01; 5]$, $\delta \in [1ms; 180ms]$ et $P = 0, 1, 2, 3$.

Le logiciel -SimNLSSys- génère les figures 2 à 5 où celles indexées par (a) décrivent l'erreur E , mesurée en m/s et calculée comme en (10), en fonction des variations de δ et K_c pour chaque $P = 0, 1, 2, 3$. Les figures indexées par (b) décrivent les courbes de niveaux de l'erreur E en fonction de δ et K_c pour chaque P .

La performance de chaque commande peut être déterminée à partir de l'analyse des courbes de niveau de l'erreur. Si l'on a une prédominance des niveaux à basse erreur sur les figures $K_c \times \delta$, la commande correspondante est robuste par rapport aux variations de δ et K_c . Si par contre, on observe la prédominance des niveaux à grande erreur, la plage des valeurs acceptables pour δ et K_c est restreinte et le contrôleur doit être calculé avec un meilleur degré d'approximation. Il est aussi intéressant d'avoir une région à faible erreur dans l'extrémité supérieure droite de la figure, ce qui indique qu'un contrôleur un grand gain peut être implanté pour des périodes d'échantillonnage grandes.

Le logiciel note les zones à faible erreur par des tonalités de bleu, les zones à erreur moyenne par des tonalités de vert et des zones à grande erreur par des tonalités entre le jaune et le rouge. Des zones blanches indiquent un erreur supérieure au seuil indiqué.

La commande émulée (figure 2) n'assure la reproduction entrée/sortie à faible erreur que pour des valeurs de δ très proches de $1ms$. Une petite augmentation de δ entraîne une forte majoration de l'erreur.

La commande corrigée du premier ordre, $P = 1$ (figure 3) assure un faible erreur pratiquement pour tout $K_c \in [0.01; 5]$ quand $\delta < 80ms$; pour un $K_c \in [2; 3.5]$ on peut en avoir un δ légèrement supérieur à $100ms$. Par contre, pour de grandes valeurs de δ et K_c l'erreur croît vite.

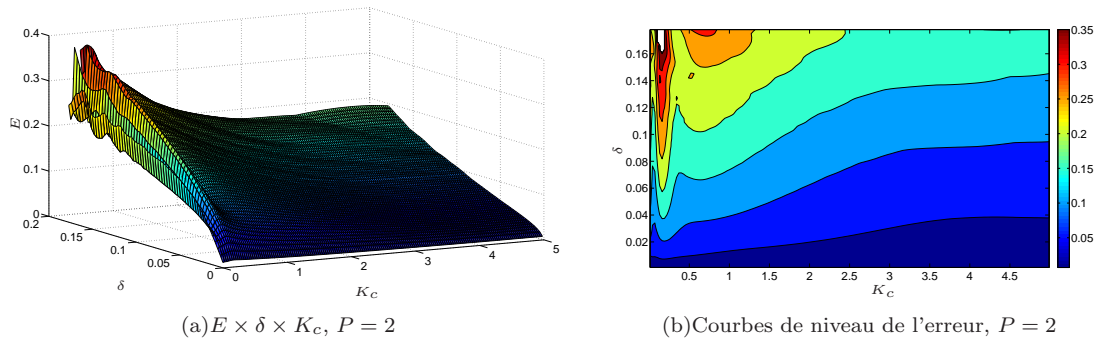


Fig. 4. Performances de la commande reproduction entrée/sortie, $P = 2$

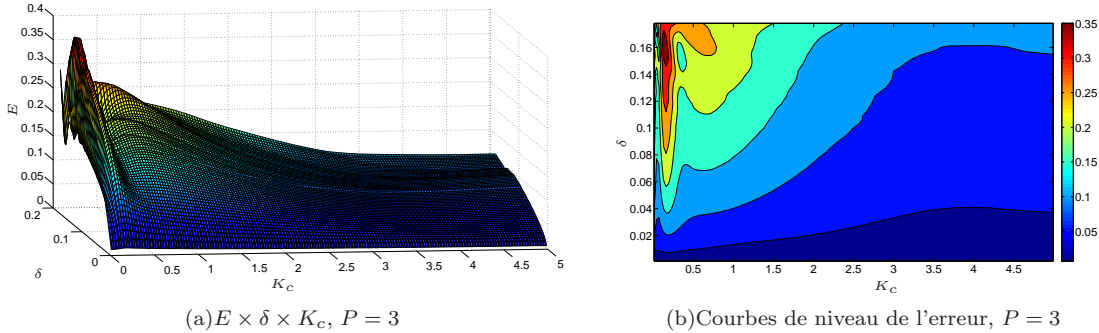


Fig. 5. Performances de la commande reproduction entrée/sortie, $P = 3$

Une commande corrigée du second ordre, $P = 2$ (figure 4) augmente les régions à faible erreur en diminuant surtout l'erreur dans le voisinage $(\delta, K_c) = (180ms, 5)$. La commande du troisième ordre, $P = 3$ (figure 5) améliore encore les performances englobant $(\delta, K_c) = (180ms, 5)$ dans une région à faible erreur, ce qui signifie que des commandes du troisième ordre à grand gain peuvent être implantées pour des grandes amplitudes de δ et assurent une erreur petite.

On conclut avec cet exemple, qu'une commande d'un ordre élevé assure une faible erreur dans la reproduction d'une trajectoire cible pour un grand choix des paramètres du contrôleur. Par contre il faut remarquer qu'à mesure que les régions de faible erreur grandissent avec P pour $K_c > 1$, elles se rétrécissent pour $K_c < 1$. Ainsi, le contrôleur du premier ordre est le plus indiqué pour des gains faibles.

VI. CONCLUSION

La technique de reproduction entrée/sortie sous échantillonnage a été analysée. Ses différences analytiques par rapport à des méthodes classiques d'implantation numérique, telles que l'estimation sous échantillonnage, la moyennisation et l'émulation de la commande conçue en temps continu par bloqueur d'ordre zéro ou par bloqueur d'ordre supérieur ont été montrées. Les performances de la commande de reproduction entrée/sortie, par rapport à des variations des paramètres du contrôleur ont été analysées avec le logiciel SimNLSys. Il est clairement mis en évidence que la robustesse par rapport à la période d'échantillonnage et au gain du contrôleur augmente considérablement avec l'ordre d'approximation de la loi de commande, ceci étant aussi fonction des non linéarités des dynamiques traitées.

Ce premier travail confirme l'avantage de disposer d'un outil logiciel pour la simulation de contrôleurs échantillonnés par un non spécialiste. Il sera ultérieurement

développé pour s'adapter à d'autres contrôleurs et simplifier toujours davantage son utilisation.

Remerciements : Les auteurs remercient D. Normand-Cyrot et S. Monaco qui ont encadré ce travail réalisé pendant le stage au L2S de V. Tanassa avec le support de l'Université Polytechnique de Bucarest.

RÉFÉRENCES

- [1] K. J. Aström and B. Wittenmark. *Computer controlled systems : theory and design*. Prentice-Hall, 1984.
- [2] J.R. Dormand and P.J.Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1) :19–26, 1980.
- [3] G.F. Franklin and J.D. Powell. *Digital control of dynamic systems*. Addison-Wesley, 1980.
- [4] S. Monaco and D. Normand-Cyrot. Issues on nonlinear digital systems. *European Journal of Control - Special Issue.*, 7(2-3) :160–178, 2001. Semi-Plenary Conference, ECC-01, Porto, Invited Paper.
- [5] S. Monaco and D. Normand-Cyrot. Advanced tools for nonlinear sampled-data systems analysis and control : Mini-tutorial. *European J. of Contr. - Special Issue , Fund. Issues in Contr.*, 13(2,3) :221–241, 2007. ECC 07, Kos, Invited Paper.
- [6] S. Monaco and D. Normand-Cyrot. Méthodologies d'analyse et de synthèse pour les systèmes échantillonnés. In *Conférence Plénière CIFA '08.*, pages 33–48, Bucarest, Roumanie, 3-5 sept 2008. Dans *Automatique Avancée et Informatique Appliquée* (P. Borne, M. Benrejeb, F. Filip, D. Popescu Eds), Editions de l'Académie de Roumanie.
- [7] D. Normand-Cyrot S. Monaco. *Quelques éléments sur la commande numérique non linéaire*, chapter Systèmes non linéaires, vol 3 - commande, pages 111–136. Masson, Paris, 1993.
- [8] F. Tiefensee, S. Monaco, and D. Normand-Cyrot. Lyapunov design under sampling for a synchronous machine. In *ECC09*, pages 2775–2780, Budapest, Hungary, 2009.
- [9] A. J. van der Schaft. *\mathcal{L}_2 -Gain and passivity techniques in nonlinear control*. Communication and Control Engineering. Springer Verlag, Heidelberg, 2nd edition, 1999.