

# Modélisation et commande d'un pendule inversé sur un robot X-Y par l'asservissement visuel

HAOPING WANG<sup>1</sup>, CHRISTIAN VASSEUR<sup>1</sup>, VLADAN KONCAR<sup>2</sup>, AFZAL CHAMROO<sup>3</sup>

<sup>1</sup> LAGIS - UMR CNRS 8146, USTL, UFR IEEA, Bâtiment P2, F-59655 Villeneuve d'Ascq cedex, France

<sup>2</sup> GEMTEX, ENSAIT, 9 rue de l'Ermitage, BP 30329, F-59056 Roubaix, France

<sup>3</sup> Université de Poitiers, LAII, 40 av. du recteur Pineau, F-86022, France

[haoping.wang@univ-lille1.fr](mailto:haoping.wang@univ-lille1.fr) - [christian.vasseur@univ-lille1.fr](mailto:christian.vasseur@univ-lille1.fr) - [vladan.koncar@ensait.fr](mailto:vladan.koncar@ensait.fr) - [afzal.chamroo@univ-poitiers.fr](mailto:afzal.chamroo@univ-poitiers.fr)

**Résumé** — Cet article présente une méthode de modélisation et de commande d'un pendule inversé sur un robot X-Y utilisant l'information fournie par une caméra CCD à bas coût. Les informations délivrées par le système de vision sont sous forme retardée et échantillonnée. La commande proposée en deux boucles nécessite la connaissance complète de l'état du pendule. L'idée principale est d'utiliser un observateur spécifique appelé Observateur Continu à fonctionnement par Morceaux (OCM). Cet observateur repose sur l'utilisation des Systèmes à Fonctionnement par Morceaux (SFM) qui représentent une classe particulière de systèmes hybrides et un Observateur Discret de Luenberger d'Ordre Réduit (ODLOR). Les simulations illustrent les performances de cette méthode.

**Mots clés** — Pendule inversé, robot X-Y, asservissement visuel, système continu à fonctionnement par morceaux, observateur continu à fonctionnement par morceaux.

## I. INTRODUCTION

Dans de nombreuses applications il peut s'avérer impossible de mettre le capteur en contact direct avec l'objet commandé. L'utilisation des techniques d'asservissement visuel fait, depuis plusieurs années, l'objet de nombreux travaux de recherche permettant de pallier cette contrainte physique. Par ailleurs, d'un point de vue économique, il peut s'avérer intéressant de considérer les cas où il n'est pas envisageable d'utiliser des capteurs de précision élevée, parce que de coût trop important. Ainsi, nous présentons ici une méthode de modélisation et de commande d'un pendule inversé sur un robot X-Y utilisant l'information fournie par une caméra CCD à bas coût.

En tant que système dynamique instable et fortement non linéaire, le système chariot-pendule inversé constitue un excellent support expérimental pour tester des algorithmes de commande. En littérature, les modèles les plus souvent considérés sont : le bras simple ou double [22], les pendules planaires simple, double, voire triple relié à un chariot par des liaisons rotoïdes [3, 6, 9, 19, 21, 25] et enfin, le pendule inversé sur un robot X-Y par un joint de cardan [7, 8, 17, 27]. Dans tous ces cas, les contrôleurs utilisent des codeurs angulaires continus de précision élevée fixés directement sur l'axe de rotation du pendule. Dans cet article, l'enjeu est d'utiliser une caméra CCD à bas coût comme capteur immatériel pour la commande. En contrepartie de son coût modeste, ce capteur induit des problèmes de précision, d'échantillonnage et de retard en raison du temps de calcul (traitement des images) et de transmission des signaux dans la boucle.

Dans [15], l'utilisation de la rétroaction visuelle en entrée des boucles de retour pour commander un robot est largement

décrite. Par exemple, des indices visuels (points, arêtes ou régions) obtenues par traitement d'image, peuvent être utilisées pour le guidage d'un robot. Dans ce sens, la vision artificielle est utilisée comme outil de rétroaction, pour la réalisation d'asservissements. Toutefois, le plus souvent, le système de vision est soit combiné à la commande dans une configuration en boucle ouverte du type « perception » puis « action » [23], soit simplement utilisé pour réaliser des visualisations et/ou des animations dans les systèmes de commande [2, 23]. Plus récemment, des tentatives d'insertion de la vision dans la boucle de commande pour stabiliser le système chariot-pendule ont été présentées [10, 26]. Toutefois, aucune réalisation convaincante de commandes simultanées de l'angle du pendule et de la position du chariot n'apparaît. Un contrôleur flou a été développé en [20], mais sur un pendule planaire et avec une caméra de grande précision. De plus, la stabilisation est limitée dans le temps et ne dure que quelques secondes. Nous trouvons [12] une autre commande qui simule le comportement humain pour un pendule planaire. Cette méthode stabilise le pendule limité dans le temps aussi ( $\leq 16$  s) et avec une grande oscillation de position angulaire  $\pm 26^\circ$ .

L'analyse des difficultés liées à la commande du pendule inversé par rétroaction visuelle fait apparaître que le signal de vision ne semble pas suffisamment bien exploité. C'est la raison pour laquelle, nous nous sommes efforcés de développer un observateur précis, rapide et robuste, afin de pallier les difficultés induites par la vision (échantillonnage et retard). Cet observateur, basé sur l'utilisation des Systèmes à Fonctionnement par Morceaux (SFM) [18], présente l'avantage de compenser les effets de retard et d'échantillonnage pour fournir des estimations continues et sans retard. Les travaux présentés ici constituent un prolongement de recherches précédentes [18, 4, 24], pour développer un Observateur Continu par Morceaux (OCM), à partir d'un Observateur Discret de Luenberger d'Ordre Réduit (ODLOR) et de deux SFM.

Le reste de l'article est organisé comme suit. La section II présente la description et la modélisation de la plateforme expérimentale. Dans la section III, le fonctionnement des SFM est détaillé. Ces derniers sont utilisés dans la construction de l'OCM et à la stabilisation du pendule inversé sur robot X-Y dans les sections IV et V. Les simulations sont analysées dans la section VI. Enfin, les conclusions sont exposées dans la section VII.

## II. DESCRIPTION ET MODELISATION

### A. Plateforme expérimentale

La plateforme expérimentale illustrée par la figure 1 est

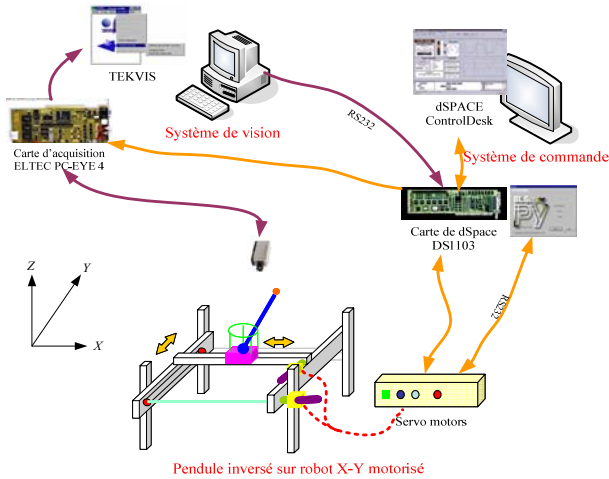


Fig.1: Plateforme expérimentale

composée principalement de trois parties:

#### 1. Système pendule inversé sur un robot X-Y motorisé

Ce système est composé d'un pendule inversé lié à un robot X-Y par un joint de cardan. Ce robot est motorisé à l'aide de deux servomoteurs SANYO DENKI PY2A015A3-A3 configurable par PY. L'entraînement du robot se fait par courroies crantées. De plus, deux codeurs (un pour chaque axe) mesurent la position du robot. Une LED infrarouge est fixée à l'extrémité haute du pendule, afin d'en faciliter le repérage par le système de vision. Enfin, une couronne circulaire limite le débattement du pendule en maintenant son angle d'inclinaison par rapport à la verticale inférieur à 50 degrés.

#### 2. Contrôleur

Les logiciels de contrôle sont implantés sur une carte dSpace DS1103. Celle-ci permet de compiler directement les algorithmes de commande à partir de Matlab/Simulink et de superviser facilement les paramètres de contrôle à l'aide du module ControlDesk. Cela permet d'afficher en temps réel les variables de la carte. Les signaux de commande sont, quant à eux, envoyés aux amplificateurs de puissance par un CAN ( $\pm 10$  V).

#### 3. Système de vision

Le système de vision associé à une caméra est constitué d'une carte d'acquisition de type ELTEC PC-EYE 4 et du logiciel de traitement d'images TEKVIS. Ce logiciel calcule les coordonnées (x, y) (en pixels) de la LED fixée à l'extrémité haut du pendule et les transmet au système de commande par une liaison RS-232. La caméra fixée à environ 2 mètres au-dessus du centre du robot X-Y est une caméra ordinaire infrarouge de faible coût (Jai M50 IR), dont la cadence d'échantillonnage est de 25 images/sec et la résolution de  $640 \times 480$  pixels.

La prise d'image de la caméra est synchronisée avec une impulsion périodique externe générée par le système de commande. Lorsque le système de commande reçoit les coordonnées en pixels de la LED, il en fait la conversion en mètres, à partir d'une table résultant d'un calibrage effectué préalablement (méthode "TSAI" présentée dans [14]). Le calibrage permet, non seulement de convertir les coordonnées

de la LED en mètres, mais également de compenser les déformations résultant du système optique de la caméra. Dans ces conditions, compte tenu des temps de traitements, les coordonnées échantillonnées de la LED sont disponibles avec un retard de 40 ms.

### B. Modélisation

Dans [7, 8, 17, 27], les auteurs utilisent une modélisation pour laquelle les axes x et y sont découplés, ce qui revient à commander le pendule selon deux projections indépendantes. Une telle modélisation induit des approximations et n'est valide que pour des petits angles d'inclinaison du pendule. Nous développons ici une méthode de modélisation valable quel que soit l'angle d'inclinaison du pendule. L'idée principale consiste à travailler, à chaque instant, dans le plan du pendule. Ainsi, on transforme le problème 3D en un problème 2D.

#### 1. Mise en équations du pendule inversé sur robot X-Y

Comme l'illustre la figure 2a, on désigne par  $P$  le point de

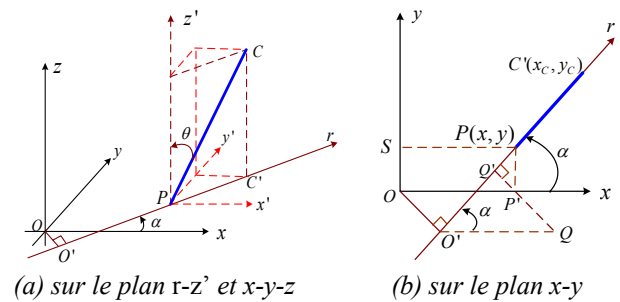


Fig.2: Projections

liaison du pendule avec le robot (cardan) et par  $O$  l'origine du plan x-y du robot X-Y. Dans un premier temps, on considère le repère  $Px'y'z'$  parallèle au repère  $Oxyz$  lié au robot. Le pendule inversé peut être projeté, à tout instant, sur deux plans orthogonaux: le plan x-y (plan du robot) contenant  $O$  et  $P$  et le plan r-z' passant par  $P$  et contenant le pendule en entier. Dans un second temps, le plan r-z' est choisi comme plan de stabilisation, de telle sorte que le problème du pendule inversé 3D devient un problème 2D dans ce plan. En se référant aux méthodes classiques [2, 9], le pendule inversé peut être modélisé comme suit, dans le plan r-z' :

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} - \omega_n^2 \sin \theta = -K \cos \theta \ddot{r} \quad (1)$$

où :  $\omega_n = \sqrt{mgl/(J + ml^2)}$ ,  $\zeta = B_r/[2\omega_n(J + ml^2)]$ ,  $K = ml/(J + ml^2)$ , avec:  $\theta$  angle du pendule avec l'axe  $Pz'$ ,  $l$  longueur du demi-pendule,  $m$  masse du pendule,  $B_r$  frottement visqueux entre le pendule et le robot,  $g$  accélération de la pesanteur,  $J = (ml^2)/3$  moment d'inertie du pendule et  $r$  position du robot sur l'axe  $r$ , l'origine de cet axe étant la projection  $O'$  de  $O$  sur le plan r-z'. Naturellement, le déplacement du robot sur l'axe  $r$  résulte d'une combinaison de mouvements selon les axes x et y.

Comme le montre la figure 2b, en considérant la projection du pendule sur le plan x-y, on peut trouver facilement la relation reliant  $x$ ,  $y$  et  $r$  :

$$r = x \cos \alpha + y \sin \alpha \quad (2)$$

où  $\alpha$  est l'angle entre l'axe  $O'r$  et l'axe  $Ox$ .

L'évolution de la coordonnée  $x$  du robot X-Y en axe  $x$  est décrite par :

$$\ddot{x} = (-\dot{x} + k_x u_x) / \tau_x \quad (3)$$

où :  $k_x$  et  $\tau_x$  sont le gain statique et la constante de temps du robot X-Y selon l'axe x et  $u_x$  sa tension d'entrée. Il en va de même pour y :

$$\ddot{y} = (-\dot{y} + k_y u_y) / \tau_y \quad (4)$$

## 2. Calcul des signaux visuels ( $\alpha$ et $\theta$ )

D'après la structure de notre plateforme, les informations qu'on peut utiliser sont les coordonnées  $(x_C, y_C)$  du haut du pendule obtenues par le système de vision. Les coordonnées  $(x, y)$  du pied du pendule sont mesurées par les codeurs. Ainsi selon les figures 2, les expressions de  $\alpha$  et  $\theta$  peuvent s'écrire :

$$\alpha = \tan^{-1}[(y_C - y)/(x_C - x)] \quad (5)$$

$$\theta = \sin^{-1}\{[(x_C - x) \cos \alpha + (y_C - y) \sin \alpha] / 2l\} \quad (6)$$

En réalité, pour l'extrémité du pendule, on ne dispose que d'informations retardées et échantillonnées :

$$(x_C(kt_e - qt_e), y_C(kt_e - qt_e)) = (x_{C,k-q}, y_{C,k-q}) \quad (7)$$

où  $t_e$  est la période d'échantillonnage du système de vision et  $qt_e$  ( $q$  entier positif) correspond au temps nécessaire pour le traitement d'image et pour la transmission des signaux.

Dans notre cas, on a  $t_e = 40$  ms et  $q = 1$ , de telle sorte qu'à chaque instant d'échantillonnage  $kt_e$ , les équations (5), (6) et (7) fournissent  $\alpha_{k-1}$  et  $\theta_{k-1}$ .

## III. SYSTEMES A FONCTIONNEMENT PAR MORCEAUX

A partir des définitions données dans [18], on peut établir les équations d'un Système à Fonctionnement par Morceaux (SFM), comme suit :

- ❖ les commutations du système se font à période  $t_e$  constante de la même période d'échantillonnage du système de vision, ce qui équivaut à un ensemble de commutation de la forme  $S = \{kt_e, k = 0, 1, 2, \dots\}$ , deux instants de communications successives se délimitent un morceau  $\Phi_k = ]kt_e, (k+1)t_e]$ .

- ❖ les équations de fonctionnement du SFM sont :

$$\dot{\lambda}(t) = \alpha \lambda(t) + \beta \varphi(t), \quad \forall t \in \Phi_k, \quad (8a)$$

$$\lambda_k^+ = \delta \psi_k, \quad \forall k \in S, \quad (8b)$$

$$w(t) = \gamma \lambda(t), \quad \forall t. \quad (8c)$$

L'équation (8a) donne l'évolution continue de l'état  $\lambda(t) \in \Sigma^{\hat{n}}$  du SFM sur le morceau  $\Phi_k$ ,  $\alpha \in \mathbb{R}^{\hat{n} \times \hat{n}}$  et  $\beta \in \mathbb{R}^{\hat{n} \times \hat{r}}$  étant les matrices d'état du SFM et  $\varphi(t) \in U^{\hat{r}}$  étant son entrée continue.

L'équation (8b) définit l'état du SFM aux instants de commutation, à partir d'une entrée discrète bornée  $\psi(kT) \in V^s$ , selon une relation linéaire caractérisée par la matrice  $\delta \in \mathbb{R}^{\hat{n} \times s}$ . L'équation (8c) est l'équation de sortie du SFM caractérisée par la matrice  $\gamma \in \mathbb{R}^{\hat{m} \times \hat{n}}$ .

La figure 3a décrit la réalisation d'un SFM et sa représentation symbolique  $\Sigma(\{kt_e\}, \alpha, \beta, \delta, \gamma)$  est donnée figure 3b où chacune des entrées du bloc est représentée par un marqueur différent.

Considérons l'intégration de l'équation [8a] sur le morceau  $\Phi_k$ , on a :

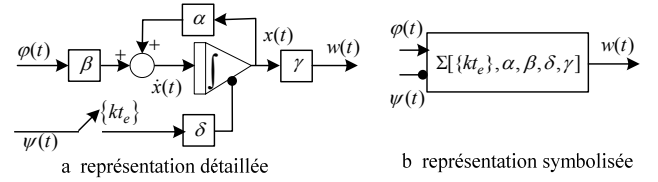


Fig.3: Réalisation d'un SFM

$$\lambda(t) = \exp(A(t - kt_e)) \lambda_k^+ + \int_{kt_e}^t \exp \alpha(t - \tau) \beta \varphi(\tau) d\tau, \quad \forall t \in \Phi_k$$

Soit en utilisant [8b] :

$$\lambda(t) = \exp(A(t - kt_e)) \delta \psi_k + \int_{kt_e}^t \exp \alpha(t - \tau) \beta \varphi(\tau) d\tau, \quad \forall t \in \Phi_k \quad (9)$$

La valeur de continuité de  $\lambda(t)$  à gauche de  $kt_e$ , notée  $\lambda_k^-$  est obtenu en écrivant l'équation (9) sur le morceau  $\Phi_{k-1}$  et en faisant tendre  $t$  vers  $kt_e$  par valeurs inférieures, soit :

$$\lambda_k^- = \exp(A t_e) \delta \psi_{k-1}^+ + \int_{(k-1)t_e}^{kt_e} \exp \alpha(t - \tau) \beta \varphi(\tau) d\tau, \quad \forall t \in \Phi_k$$

En général, on a  $\lambda_k^- \neq \lambda_k^+$ .

L'avantage de ce type de système se situe dans la réinitialisation d'état du SFM à chaque instant de commutation comme illustrée figure 4 dans le cas où la dimension d'état

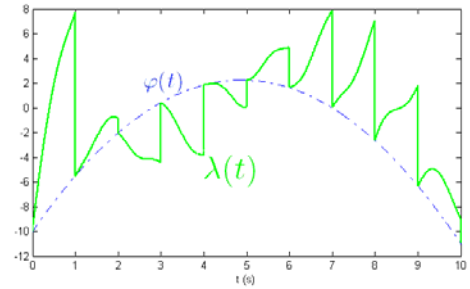


Fig.4: Evolution de l'état d'un SFM ( $t_e = 1$  s)

$\hat{n} = 1$  et  $\delta = 1$ . Dans la section suivante, on utilise le principe des SFM dans le développement d'un OCM et la construction de la commande afin de stabiliser l'ensemble du système.

## IV. COMMANDE EN DOUBLE BOUCLES

L'architecture de commande du pendule inversé 3D est présentée figure 5. Les retours fournissent les coordonnées continues  $(x, y)$  par les codeurs et la position angulaire retardée

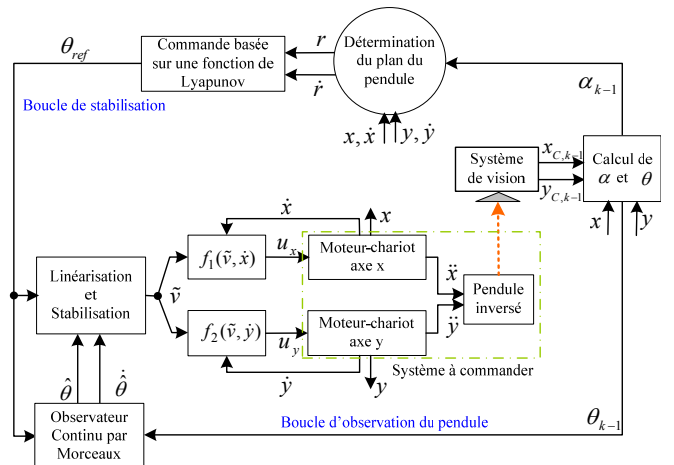


Fig.5: Architecture de commande

et échantillonnée  $\alpha_{k-1}$  et  $\theta_{k-1}$  par le système de vision. A chaque instant d'échantillonnage où à chaque période d'échantillonnage des commutation des SFM, le plan du pendule est repéré par  $\alpha_{k-1}$ . Ensuite, le système de commande agit dans ce plan en considérant deux boucles. La première est appelée « boucle d'observation du pendule ». Il vise à reconstruire l'état continu du pendule à partir des informations retardées et échantillonnées  $\theta_{k-1}$  à l'aide d'un Observateur Continu par Morceaux (OCM) constitué principalement deux SFM et un Observateur ODLOR. La seconde, appelée « boucle de stabilisation », est utilisée pour stabiliser l'ensemble robot X-Y + pendule, selon une commande construite  $\theta_{ref}$  constante par morceaux à partir d'une fonction de Lyapunov.

#### A. Boucle d'observation du pendule

L'équation (1) peut se ramener au système d'équations suivant :

$$\ddot{\theta} = \tilde{v} \quad (10)$$

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} - \omega_n^2 \sin \theta = -K \cos \theta \tilde{v} \quad (11)$$

Afin de stabiliser ce système, une dynamique stable peut être imposée, en définissant une nouvelle commande  $v$  par la relation :

$$-2\zeta\omega_n\dot{\theta} + \omega_n^2 \sin \theta - K \cos \theta \tilde{v} = -2Z\Omega_n\dot{\theta} - \Omega_n^2\theta + K_2v \quad (12)$$

ce qui conduit à :

$$\tilde{v} = \left[ -2(\zeta\omega_n - Z\Omega_n)\dot{\theta} + \Omega_n^2\theta + \omega_n^2 \sin \theta - K_2v \right] / K \cos \theta, |\theta| < \pi/2 \quad (13)$$

où  $\Omega_n$ ,  $Z$  et  $K_2$  représentent respectivement la pulsation naturelle, l'amortissement et le gain du modèle stable.

Après transformation, le modèle devient :

$$\ddot{\theta} = \left[ -2(\zeta\omega_n - Z\Omega_n)\dot{\theta} + \Omega_n^2\theta + \omega_n^2 \sin \theta - K_2v \right] / K \cos \theta \quad (14)$$

$$\ddot{\theta} + 2Z\Omega_n\dot{\theta} + \Omega_n^2\theta = K_2v \quad (15)$$

En adoptant pour (15) une représentation d'état, il vient :

$$\dot{\Theta}(t) = A\Theta(t) + Bv(t) \quad (16)$$

$$\text{avec : } A = \begin{bmatrix} 0 & 1 \\ -\Omega_n^2 & -2Z\Omega_n \end{bmatrix}, B = \begin{bmatrix} 0 \\ K_2 \end{bmatrix} \text{ et } \Theta(t) = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}.$$

Dans notre cas, on ne dispose ni de  $\theta(t)$ , ni de  $\dot{\theta}(t)$ , mais seulement de  $\theta_{k-1}$ . Afin d'obtenir des estimations  $\hat{\theta}(t)$  et  $\dot{\hat{\theta}}(t)$  de  $\theta(t)$  et  $\dot{\theta}(t)$ , on développe ci-après un OCM réalisant la combinaison d'un Observateur Discret de Luenberger d'Ordre Réduit (ODLOR) et de deux SFM tels que définis dans la section 2 noté sous la forme  $(\Sigma(\{kt_e\}, \alpha, \beta, \delta, \gamma))$ . Dans le but d'estimer  $\Theta(t)$ , on choisit :  $\alpha$  &  $\beta$  comme définies dans l'équation (16) de  $A$  &  $B$  et  $\delta = \gamma = I_2$ . Ainsi l'OCM présenté figure 6 est construit en trois étapes :

#### ❖ Première étape : SFM I

En utilisant le SFM I, avec  $\varphi(t)=v(t)$  et  $\psi(t)=0$ , on obtient :

$$M_{k-1}(t) = \int_{(k-1)t_e}^t \exp A(t-\tau)Bv(\tau)d\tau,$$

Ce qui conduit en faisant  $t = kt_e$ , à :

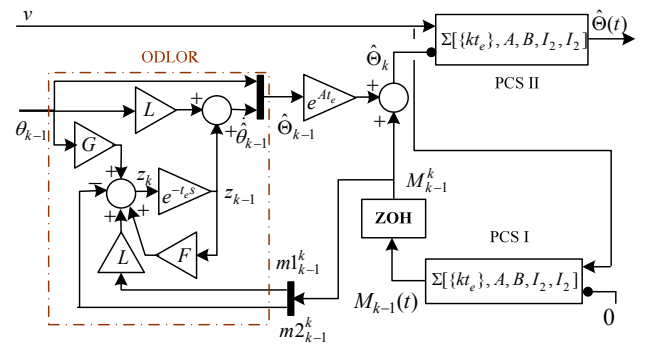


Fig.6: Observateur Continu par Morceaux

$$M_{k-1}^k = \int_{(k-1)t_e}^{kt_e} \exp A(kt_e - \tau)Bv(\tau)d\tau = \begin{bmatrix} m1_{k-1}^k \\ m2_{k-1}^k \end{bmatrix} \quad (17)$$

Le rôle du BOZ (Bloqueur d'Ordre Zéro) est donc de récupérer et de mémoriser cette valeur à  $kt_e$ .

#### ❖ Deuxième étape : ODLOR

$\Theta_{k-1}$  est estimé par l'ODLOR selon :

$$z_k = Fz_{k-1} + G\theta_{k-1} + (m2_{k-1}^k - Lm1_{k-1}^k) \quad (18a)$$

$$z_{k-1} = \hat{\theta}_{k-1} - L\theta_{k-1} \quad (18b)$$

où  $F$ ,  $G$  et  $L$  sont définies, à partir de  $\exp(At_e) = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}$ ,

comme suit :  $F = f_{22} - Lf_{12}$ ,  $G = FL + (f_{21} - Lf_{11})$ ,  $L = (f_{22}/f_{12}) \in R$  (Valeur imposée pour maximiser la convergence de l'ODLOR).

Ainsi, en estimant  $\dot{\theta}_{k-1}$  par  $\dot{\hat{\theta}}_{k-1} = z_{k-1} + L\theta_{k-1}$ , on obtient  $\hat{\Theta}_{k-1}$ , puis, par intégration de (16) sur morceau  $\Phi_{k-1}$ , on obtient  $\hat{\Theta}_k$  :

$$\hat{\Theta}_k = \exp(At_e)\hat{\Theta}_{k-1} + M_{k-1}^k \quad (19)$$

#### ❖ Troisième étape : SFM II

En utilisant le SFM III, avec  $\varphi(t)=v(t)$  et  $\psi(t)=\hat{\Theta}_k$ , on estime finalement  $\Theta(t)$  par :

$$\hat{\Theta}(t) = \exp(At)\hat{\Theta}_k + \int_{kt_e}^t \exp A(t-\tau)Bv(\tau)d\tau \quad (20)$$

En se réinitialisant à chaque période d'échantillonnage, l'OCM permet de compenser les effets de retard et discrétisation. Cela conduit à des estimations continues et sans retard,  $\hat{\theta}(t)$  et  $\dot{\hat{\theta}}(t)$ .

#### Analyse de l'erreur de l'OCM :

On peut discrétiser le système (16) sous la forme suivante :

$$\Theta_k = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \Theta_{k-1} + M_{k-1}^k$$

avec :  $M_{k-1}^k$  défini par (17). De la même façon, on peut écrire :

$$\Theta_{k-1} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \Theta_{k-2} + M_{k-2}^{k-1}$$

$$\text{avec : } M_{k-2}^{k-1} = \int_{(k-2)t_e}^{(k-1)t_e} \exp A(kt_e - \tau)Bv(\tau)d\tau = \begin{bmatrix} m1_{k-2}^{k-1} \\ m2_{k-2}^{k-1} \end{bmatrix}.$$



Comme  $\Theta(t) = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}$ , on a les équations équivalentes:

$$\theta_{k-1} = f_{11}\theta_{k-2} + f_{12}\dot{\theta}_{k-2} + m1_{k-2}^{k-1} \quad (21a)$$

$$\dot{\theta}_{k-1} = f_{21}\theta_{k-2} + f_{22}\dot{\theta}_{k-2} + m2_{k-2}^{k-1} \quad (21b)$$

En utilisant l'observateur ODLOR défini dans les équations (18), on a:

$$\dot{\hat{\theta}}_{k-1} = L\hat{\theta}_{k-1} + z_{k-1} \quad (22a)$$

$$z_{k-1} = Fz_{k-2} + G\theta_{k-2} + (m2_{k-2}^{k-1} - Lm1_{k-2}^{k-1}) \quad (22b)$$

Pour le morceau  $\Phi_k$ , on dispose des valeurs réelles  $\theta_{k-1}$  et  $\theta_{k-2}$ . On remplace donc  $\hat{\theta}_{k-1}$  dans l'équation (22a) par  $\theta_{k-1}$ :

$$\begin{aligned} \dot{\hat{\theta}}_{k-1} &= L\theta_{k-1} + z_{k-1} \\ &= L(f_{11}\theta_{k-2} + f_{12}\dot{\theta}_{k-2} + m1_{k-2}^{k-1}) \\ &\quad + [Fz_{k-2} + G\theta_{k-2} + (m2_{k-2}^{k-1} - Lm1_{k-2}^{k-1})] \end{aligned} \quad (23)$$

On évalue les erreurs de mesure :

$$e_{k-1} = \Theta_{k-1} - \hat{\Theta}_{k-1} = \begin{bmatrix} e1_{k-1} \\ e2_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ e2_{k-1} \end{bmatrix},$$

dans lequel:

$$\begin{aligned} e2_{k-1} &= \dot{\theta}_{k-1} - \dot{\hat{\theta}}_{k-1} \\ &= f_{21}\theta_{k-2} + f_{22}\dot{\theta}_{k-2} + m2_{k-2}^{k-1} \\ &\quad - L(f_{11}\theta_{k-2} + f_{12}\dot{\theta}_{k-2} + m1_{k-2}^{k-1}) \\ &\quad - [Fz_{k-2} + G\theta_{k-2} + (m2_{k-2}^{k-1} - Lm1_{k-2}^{k-1})] \end{aligned} \quad (24)$$

A partir de (22a), on a :

$$z_{k-1} = \dot{\hat{\theta}}_{k-1} - l\theta_{k-1} = \dot{\theta}_{k-1} - e2_{k-1} - l\theta_{k-1},$$

de la même façon, on peut écrire:

$$z_{k-2} = \dot{\theta}_{k-2} - e2_{k-2} - l\theta_{k-2},$$

on le remplace dans (24):

$$\begin{aligned} e2_{k-1} &= \dot{\theta}_{k-1} - \dot{\hat{\theta}}_{k-1} \\ &= f_{21}\theta_{k-2} + f_{22}\dot{\theta}_{k-2} \\ &\quad - Lf_{11}\theta_{k-2} - Lf_{12}\dot{\theta}_{k-2} \\ &\quad - F(\dot{\theta}_{k-2} - e2_{k-2} - L\theta_{k-2}) - G\theta_{k-2} \end{aligned}$$

Finalement, on a:

$$\begin{aligned} e2_{k-1} &= F e2_{k-2} \\ &\quad + (f_{22} - Lf_{12} - F)\dot{\theta}_{k-2} \\ &\quad + (f_{21} - Lf_{11} + FL - G)\theta_{k-2} \end{aligned} \quad (25)$$

Pour que  $e2_{k-1}$  dans le système (25) tende vers zéro quand  $k \rightarrow +\infty$ , d'après les équations de (18), on a choisi  $F = f_{22} - Lf_{12}$  et  $G = FL + f_{21} - Lf_{11}$ . Ainsi, (25) devient :

$$e2_{k-1} = F e2_{k-2} \quad (26)$$

De plus comme  $F = f_{22} - Lf_{12}$ , on peut choisir la valeur de  $L$  pour rendre  $F$  stable.

Cela conduit à :

$$\lim_{k \rightarrow +\infty} e2_{k-1} = 0 \quad (27)$$

c-à-d,  $\hat{\theta}_{k-1}$  converge vers  $\dot{\theta}_{k-1}$ . Ainsi, d'après les équations de (19-20), on obtient :

$$\lim_{t \rightarrow +\infty} \hat{\Theta}(t) = \Theta(t).$$

### B. Boucle de stabilisation

Pour stabiliser le robot, on définit la fonction de Lyapunov ci-après :

$$V(r, \dot{r}) = (\chi r^2 + \delta \dot{r}^2)/2, \text{ avec: } \chi > 0 \text{ et } \delta > 0$$

ayant pour dérivée :  $\dot{V}(r, \dot{r}) = \dot{r}(\chi r + \delta \ddot{r}) = \dot{r}(\chi r + \delta \ddot{v})$ , d'après l'équation (10).

Pour garantir la stabilisation du robot, on impose que cette dérivée soit négative, par la relation :

$$(\chi r + \delta \ddot{v}) = -\mu[1 - e^{-(\chi r^2 + \delta \dot{r}^2)/2}] \dot{r}, \quad \mu > 0$$

ce qui conduit à la commande  $\ddot{v}$  :

$$\ddot{v} = -[\chi r + \mu(1 - e^{-(\chi r^2 + \delta \dot{r}^2)/2})\dot{r}]/\delta \quad (28)$$

cette commande  $\ddot{v}$  définie par l'équation (28) sous forme de PD ramène donc  $(r, \dot{r})$  à  $(0,0)$ . Il reste à garantir la stabilisation de  $\theta$ .

Pour cela, à partir des équations (14-15), Si l'on applique à l'équation (15) une commande  $v$  constante par morceaux (fonction escalier) et si chaque morceau est suffisamment long pour atteindre le régime stationnaire de  $\theta$  ( $\ddot{\theta} = \dot{\theta} = 0$ ), nous pouvons réécrire les équations (14-15) sous la forme suivante :

$$\ddot{r} = \ddot{v} = \omega_n^2 \tan(\theta_{ref})/K \quad (29)$$

$$v = \Omega_n^2 \theta_{ref}/K_2 \quad (30)$$

Dans lesquels  $\theta_{ref}$  désigne la valeur de  $\theta$  en régime stationnaire et proportionnel à la commande  $v$  constante par morceaux (en raison de simplicité, dans les parties suivantes, on choisit  $K_2 = \Omega_n^2$  pour que  $v = \theta_{ref}$ )

La figure 7 ci-après illustre la construction de  $\theta_{ref}$  :

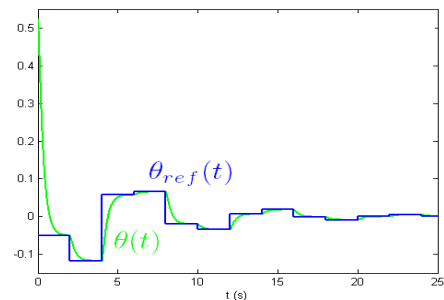


Fig.7: Commande constante par morceaux avec une période de 2 s,  $v = \theta_{ref}$  amène la stabilité de  $\theta$

Si les morceaux de  $v = \theta_{ref}$  deviennent infiniment courts (le pas de calcul de l'ordinateur),  $v$  tend vers un signal continu et les variations d'amplitude de  $v$  sur chaque morceau tendent vers zéro. Dans ces conditions, les équations donnant  $\theta_{ref}$  restent valides (hypothèse de quasi-stationnarité de  $\theta$  [13, 16]). La figure 8 ci-après illustre cet aspect :

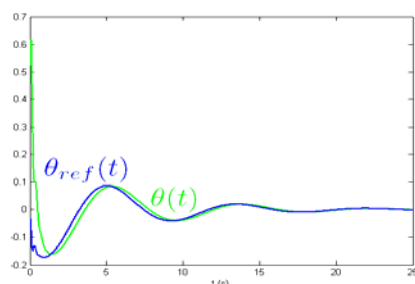


Fig.8 Commande devient continue avec une période de 0.001 s,  $\theta_{ref}$  amène la stabilité de  $\theta$

Finalement, en reportant l'équation (18) dans l'équation (17), on obtient l'expression de  $\theta_{ref}$ , en fonction de  $r$  et  $\dot{r}$  :

$$\theta_{ref}(r, \dot{r}) = \tan^{-1} \left\{ -K[\chi r + \mu(1 - e^{-(\chi r^2 + \delta \dot{r}^2)/2})\dot{r}] / (\delta \omega_n^2) \right\} \quad (31)$$

Par construction d'une fonction de Lyapunov à dérivée négative, on sait que la commande (28) stabilise le robot ( $r = \dot{r} = 0$ ). Dès lors, d'après (31), il apparaît que la stabilisation du robot conduit à  $\theta_{ref}(0,0) = 0$  et donc à  $v=0$ .

D'après l'équation (15),  $v=0$  permet d'obtenir  $\theta = \dot{\theta} = 0$ . L'introduction de  $\theta_{ref}$ , tel que défini par les équations (29) et (30), assure donc la stabilité de l'ensemble robot X-Y + pendule.

Finalement, il reste à calculer  $u_x$  et  $u_y$  en fonction de  $\tilde{v}$  d'après l'architecture de commande. Pour cela, en considérant (3) et (4) et en réalisant des boucles locales appropriées, on obtient  $k_x = k_y = k$  et  $\tau_x = \tau_y = \tau$ . Ainsi, pour équilibrer le pendule inversé dans son plan de stabilisation (plan r-z'), il suffit d'imposer que le rapport des accélérations selon les axes x et y soit égal à  $\tan \alpha$  :

$$\ddot{y}/\ddot{x} = \tan \alpha \quad (32)$$

Cette relation supplémentaire permet de calculer les relations entre  $\tilde{v}$ ,  $u_x$  et  $u_y$  illustré sur la figure 5 :

$$u_x = f_1(\tilde{v}, \dot{x}) = (\dot{x} + \tau \cos \alpha \tilde{v})/k \quad (33)$$

$$u_y = f_2(\tilde{v}, \dot{y}) = (\dot{y} + \tau \sin \alpha \tilde{v})/k \quad (34)$$

## V. RESULTATS DE SIMULATION

### A. Paramètres du système

Pour le système réel illustré figure 9, le pendule en laiton a

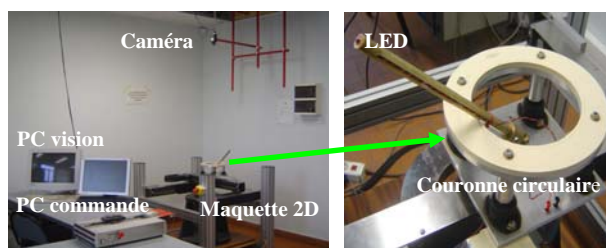


Fig.9: Vue de la plateforme expérimentale

une masse de 560 g et une longueur de 31,5 cm. Le frottement visqueux entre le pendule et le robot X-Y est estimé à  $B_r = 0.001$  kg/s. Cela donne :  $\zeta = 0.0039$  et  $\omega_n = 6.781$  rad/s.

Les paramètres du système robot sont imposés par les boucles locales à  $k = 2.92$  et  $\tau = 0.008$ , selon les axes x et y. Enfin les valeurs des différents paramètres introduits précédemment sont :  $K_2 = \omega_n^2$ ,  $Z = 1.2$ ,  $\Omega_n = \omega_n$ ,  $\chi = 2.5$ ,  $\delta = 10$  et  $\mu = 8$ .

### B. Résultats de simulation

La figure 10 illustre la stabilisation dans un plan quelconque. Cette figure fait apparaître : (a) la projection du plan du pendule sur le plan x-y dans lequel le point d'étoile représente les coordonnées initiales du robot  $(x_0, y_0) = (1, -1)$  et le point de cercle représente la projection d'origine de  $O$  sur l'axe  $r$   $(x_{cle}, y_{cle}) = (0.317, 0.183)$ , (b) les déplacements selon x, y et r, (c) les positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ ), (d) les vitesses angulaires du pendule ( $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$ ), (e) les commandes d'axes ( $u_x$ ,  $u_y$ ) et finalement (f)  $\theta_{ref}$  et  $\theta(t)$ .

La stabilisation de l'ensemble robot X-Y - pendule, à l'intérieur de la zone autorisée, se fait bien à l'origine  $O'$  de l'axe r. Enfin, il faut souligner que l'application de l'hypothèse de quasi-stationnarité n'apporte aucune restriction. Ainsi, aucune oscillation permanente n'apparaît, ni sur le déplacement du robot, ni sur le pendule. Les résultats de simulation illustrent parfaitement que  $\theta_{ref}(r, \dot{r})$  assure la stabilité du système global.

Afin d'illustrer la performance du contrôleur de Lyapunov dans la boucle de stabilisation, on le compare avec la méthode directe de PD sans l'utilisation de la fonction de Lyapunov, pour l'équation (18), on construit :

$$\ddot{r} = \tilde{v} = -a_1 \dot{r} - a_0 r$$

Dans laquelle  $a_0$  et  $a_1$  sont les paramètres choisies d'après le critère de Routh. Même si l'on choisit les valeurs propres du robot très négatives, le système diverge assez rapidement.

Comme dans tout système mécanique, les incertitudes d'identification et les variations des paramètres (notamment le coefficient de frottement  $B_r$ ) en fonction de  $\theta(t)$ ,  $\dot{\theta}(t)$  sont inévitables. Dans ce sens, pour étudier la robustesse de la stratégie de commande proposée, on a ajouté simultanément : un terme de variation sur le paramètre  $B_r$  de la forme  $B_r \sin 2\pi t$ , une erreur de mesure sur  $\alpha_{k-1}$  et  $\theta_{k-1}$  de la forme du bruit [white noise] illustré par figure 11. La simulation correspondante apparaît figure 12. Il apparaît que, même dans ce cas, la méthode proposée assure la stabilité du système de robot X-Y + pendule.

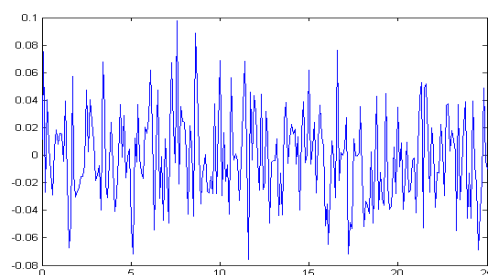
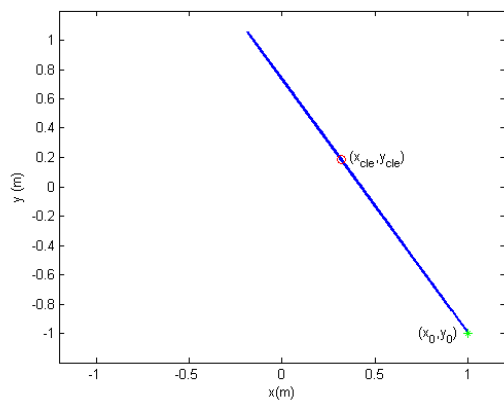
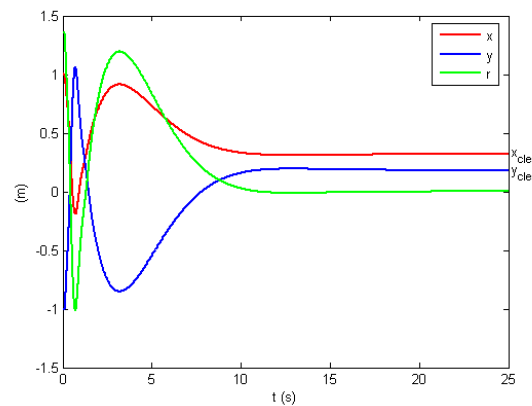


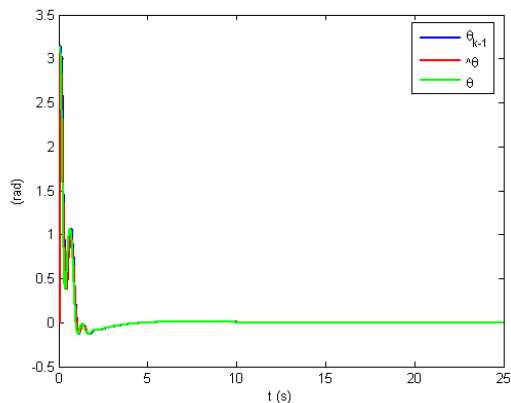
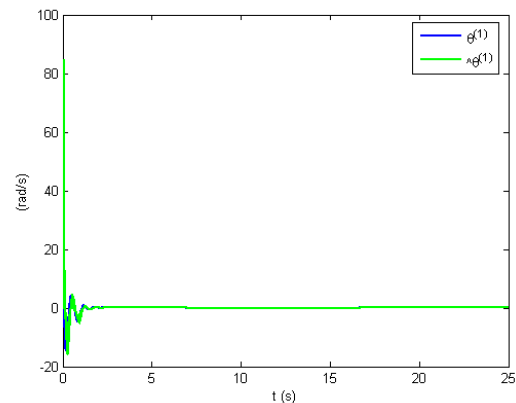
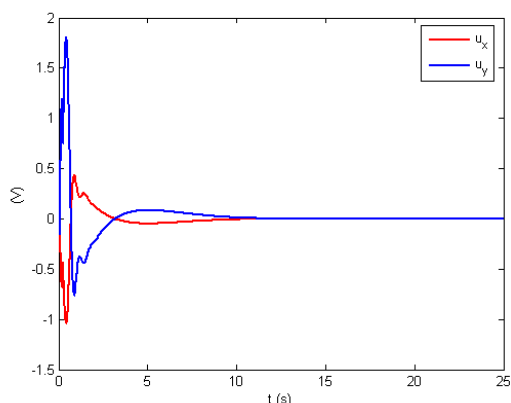
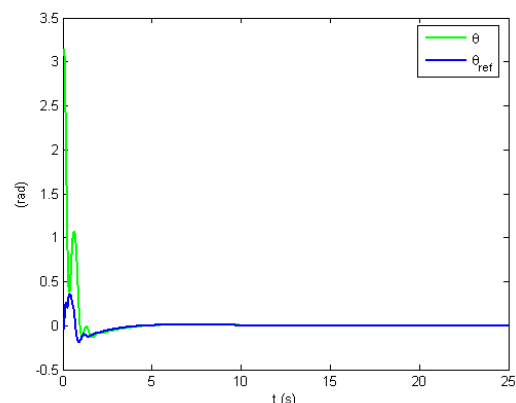
Fig.11: Bruit de mesure



(a) Plan du pendule sur plan x-y



(b) Déplacements selon x, y et r

(c)  $\theta_{k-1}$ ,  $\hat{\theta}(t)$  et  $\theta(t)$ (d)  $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$ (e) Commandes  $u_x$  et  $u_y$ (f)  $\theta(t)$  et  $\theta_{ref}(t)$ Fig.10: Résultats de Simulation avec  $\{\alpha_0 = 5\pi/3 \text{ rad}, \theta_0 = \pi \text{ rad}, x_0 = 1 \text{ m}, y_0 = -1 \text{ m}\}$ 

## VI. CONCLUSION

Cet article présente une méthode de modélisation et de stabilisation d'un pendule inversé sur un robot X-Y en utilisant un retour fourni par une caméra à bas coût et par deux codeurs incrémentaux. A chaque instant d'échantillonnage de la vision, la commande détermine le plan de stabilisation du pendule, ce qui permet de transformer le problème 3D en un problème 2D. Ensuite, le système de commande agit dans ce plan en considérant deux boucles. La première, appelée « boucle d'observation du pendule », linéarise, stabilise et observe le pendule, à partir des informations échantillonnées et retardées délivrées par le système de vision. La seconde, appelée « boucle de stabilisation », est utilisée pour stabiliser l'ensemble chariot-pendule, selon une commande construite à partir d'une fonction de Lyapunov.

Les résultats de simulation montrent l'efficacité et la robustesse de la méthode, en particulier pour un système mécanique sous actionné ayant une dynamique élevée à déphasage non minimal.

## VII. REFERENCES

- [1] Astrom K. J. and Furuta K., Swinging up a pendulum by energy control, *Automatica*, Vol. 36, pp. 287-295, 2000.
- [2] Armando A.R., Richard P.M., Oguzhan C. and Thanate D., Description of a modelling, simulation, animation, and real-time control (MoSART) environment for a class of electromechanical systems, *IEEE Trans. on Education*, Vol. 48, No. 3, pp. 359-374, 2005.

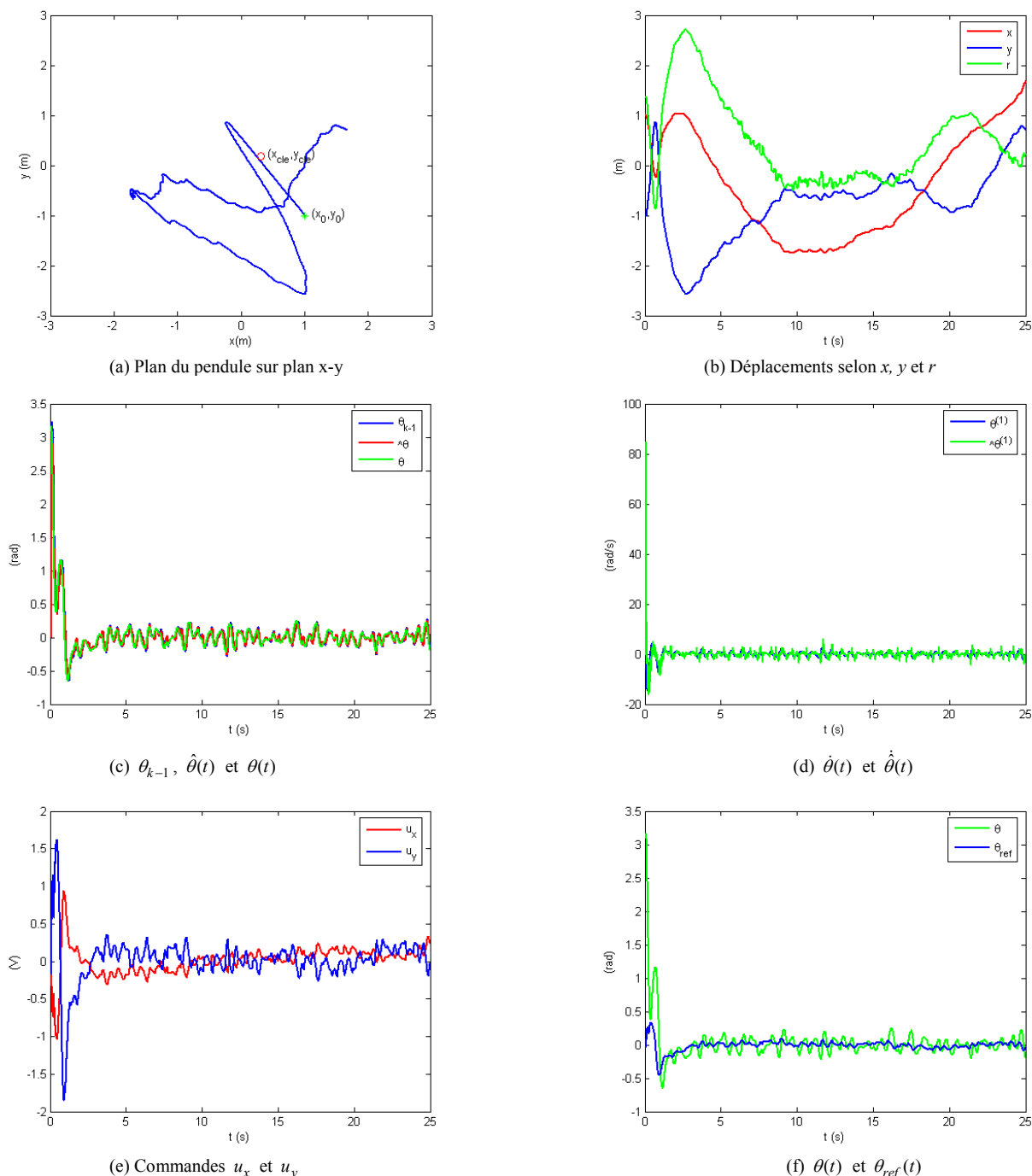


Fig.12: Résultats de simulation avec  $\{\alpha_0 = 5\pi/3 \text{ rad}, \theta_0 = \pi \text{ rad}, x_0 = 1 \text{ m}, y_0 = -1 \text{ m}\}$

- [3] Bugeja M., « Non-linear swing-up and stabilizing control of an inverted pendulum system », *EUROCON*, Ljubljana, Slovenia, 2003.
- [4] Chamroo A., Vasseur C. and Wang H.P., « Plant control using digital sensors that introduce a delayed and sampled output », *11<sup>th</sup> ELMA World Congress*, Vol. 1, pp. 119-124, Sofia, Bulgaria, Sep. 2005
- [5] Chatterjee D., Patra A. and Joglekar H., « Swing-up and stabilization of a cart-pendulum system under restricted cart track length », *System and Control Letters*, Vol. 47, No. 4, pp. 355-364, Nov 2002.
- [6] Chen L. and Smith R., « Closed-loop model validation for an inverted pendulum experiment via a linear matrix inequality approach », *Proc. Of the 36<sup>th</sup> CDC*, San Diego, USA, 1997.
- [7] Cho H.K. and Jung S., « Balancing and Position Tracking Control of An Inverted Pendulum on An X-Y Plane Using Decentralized Neural Networks », *Proc. of 2003 IEEE/ASME Inter. Conf. on Advanced Intelligent Mechatronics*, 2003,
- [8] Cho H.K., Jung S., « Neural Network Position Tracking Control of an Inverted Pendulum by an X-Y Table Robot », *Proc. of the 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, pp. 181-186, Oct 2003.
- [9] Deley D.W., « Controlling an inverted pendulum: example of a digital feedback control system », Jan 4, 2007, Accessible à : <http://members.cox.net/srice1/pendulum/index.htm>
- [10] Espinoza-Quesada E.S. and Ramos-Velasco L.E., « Vision based control of an underactuated system using a reduced observer », *Electronics, Robotics and Automotive Mechanics Conference*, Vol. 1, pp. 9-14, Sep, 2006,
- [11] Fierro R., Lewis F.L. and Lowe A., « Hybrid control for a class of underactuated mechanical systems », *IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 29, No. 6, pp. 649-654, 1999.
- [12] K.I Fukuda, S. Ushida and K. Deguchi, « Just-in-time control image-based inverted pendulums with a time-delay » *SICE-ICASE International Joint Conference*, pp. 4016-4021, Busan, Korea, Oct. 2006.



- [13] Guemghar K., Srinivasan B., Mullhaupt P. and Bonvin D., « Analysis of cascade structure with predictive control and feedback linearization », *IEE Control Theory & Applications*, Vol. 152, No. 3, pp. 317-324, 2005.
- [14] Heikkila J. and Silven O., « A four-step camera calibration procedure with implicit image correction », *1997 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 1106-1112, 1997.
- [15] Hutchinson S., Hager G.D. and Corke P.I., « A tutorial on visual servo control », *IEEE Trans. on Robotics and Automation*, Vol. 12, No. 5, pp. 651-670, 1996.
- [16] Isidori A., *Nonlinear control systems*, Springer, Third edition, 1994.
- [17] Jung S. and Cho H.K., « Decoupled neural network reference compensation technique for a PD controlled two degrees-of-freedom inverted pendulum », *Int. J. of Control, Automation and Systems*, Vol. 2, No. 1, pp.92-99, 2004.
- [18] Koncar V. and Vasseur C., « Systèmes à fonctionnement par morceaux et poursuite échantillonnée », *APII-JESA*, Vol. 35, N° 5, pp. 665 – 689, 2001
- [19] Lozano R., Fantoni I. and Block D.J., « Stabilization of the inverted pendulum around its homoclinic orbit », *Systems & Control Letters*, Vol. 40, pp.194-204, 2000.
- [20] Magana M.E. and Holzapfel F., « Fuzzy-logic control of an inverted pendulum with vision feedback », *IEEE Trans. on Education*, Vol. 41, No. 2, pp.165-170, 1998.
- [21] Ohsumi A. and Izumikawa T., « Nonlinear control of swing-up and stabilization of an inverted pendulum », *Proc. Of the 34th CDC*, New Orleans, pp. 3873-3880, Dec. 1995.
- [22] Olfati-Saber R., « Fixed point controllers and stabilization of the cart-pole system and the rotating pendulum », *Proc. of the 38th CDC*, Phoenix, p. 1174-1181, Dec 1999.
- [23] Sanchez J., Dormido S., Pastor R. and Morilla F., « A Java/Matlab-based environment for remote control system laboratories: illustrated with an inverted pendulum », *IEEE Trans. on Education*, Vol. 47, No. 3, pp. 321-329, Aug. 2004.
- [24] Wang H.P., Vasseur C., Chamroo and A., Koncar V., « Sampled tracking for delayed systems using piecewise functioning controller », *Proc. of the 4<sup>th</sup> IEEE conf. on Computational Engineering in Systems Applications*, Vol. 2, Beijing, China, pp.1326-1333, Oct. 2006.
- [25] Wei Q., Dayawansa W.P. and Levine W.S., « Nonlinear controller for an inverted pendulum having restricted travel », *Automatica*, Vol. 31, No. 6, pp.841-850, 1995.
- [26] Wenzel L., Vazquez N., Nair D. and Jamal R., « Computer vision based inverted pendulum », *Proc. of the 17th IEEE Instrumentation and Measurement Technology Conference*, Vol. 3, pp.1319-1323, 2000.
- [27] Yang R., Kuen Y. and Li Z., « Stabilization of A 2-DOF Spherical Pendulum on X-Y Table », *Proc. of the 2000 IEEE International Conference on Control Applications*, USA, pp.724-729, Sep 2000.