

# THÈSE

Présentée à

**L'UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE**

en vue de l'obtention du grade de

**DOCTEUR D'UNIVERSITÉ**

Spécialité

**Automatique et Informatique Industrielle**

Par

**Haoping WANG**

---

**OBSERVATION ET COMMANDE PAR  
SYSTÈMES À FONCTIONNEMENT PAR MORCEAUX :  
APPLICATION AUX ASSERVISSEMENTS VISUELS**

---

Soutenue le 19 Novembre 2008 devant le jury constitué de :

<i>Rapporteurs</i>	Jacques LOTTIN	Professeur <i>Ecole Polytechnique Universitaire de Savoie (EPUS)</i>
	Ahmed EL HAJJAJI	Professeur <i>Université de Picardie Jules Verne (UPJV)</i>
<i>Examineurs</i>	Nicolai CHRISTOV	Professeur <i>Université des Sciences &amp; Technologies de Lille (USTL)</i>
	Driss MEHDI	Professeur <i>Université de Poitiers (UP)</i>
	Afzal CHAMROO	Maître de Conférences <i>Université de Poitiers (UP)</i>
<i>Directeur</i>	Christian VASSEUR	Professeur <i>Université des Sciences &amp; Technologies de Lille (USTL)</i>
<i>Co-directeur</i>	Vladan KONCAR	Professeur <i>École Nationale Supérieure des Arts et Industries Textiles (ENSAIT), Roubaix</i>

*A ma femme*

*A mes parents et beaux parents*

*Aux tous les gens qui m'encouragent, à tous ceux qui comptent pour moi*

*Et aux futurs lecteurs.....*

*La théorie, c'est quand on sait tout et que rien ne fonctionne.*

*La pratique, c'est quand tout fonctionne et que personne ne sait pourquoi.*

*Ici, nous avons réuni THEORIE et PRATIQUE :*

*Rien ne fonctionne et personne ne sait pourquoi !*

*-----Albert Einstein*

# Table des matières

<b>Remerciements .....</b>	<b>ix</b>
<b>Avant-propos.....</b>	<b>xi</b>
<b>Abréviations et acronymes.....</b>	<b>xiii</b>

<b>Introduction générale .....</b>	<b>1</b>
0.1.    Motivations.....	2
0.2.    Problématique et approche .....	3
0.3.    Contributions .....	6
0.4.    Organisation du mémoire .....	6

## Chapitre 1

<b>Des systèmes hybrides aux systèmes à fonctionnement par morceaux .....</b>	<b>9</b>
<b>1.1.    Introduction .....</b>	<b>10</b>
<b>1.2.    Systèmes hybrides : présentation générale.....</b>	<b>11</b>
1.2.1.    Généralités sur les systèmes hybrides .....	11
1.2.2.    Définition de SANTIS .....	13
1.2.3.    Quelques classes particulières de systèmes hybrides .....	14
1.2.3.1.    Systèmes à Commutations (Swiched Systems).....	15
1.2.3.2.    Systèmes à Sauts (Jump Systems).....	15
1.2.3.3.    Systèmes Linéaires Affines Constants par Morceaux (Piecewise Affine Systems) .....	17

<b>1.3.</b>	<b>Systèmes à fonctionnement par morceaux.....</b>	<b>18</b>
1.3.1.	Introduction .....	18
1.3.2.	Formalisme général .....	18
1.3.3.	Systèmes continus à fonctionnement par morceaux .....	20
1.3.3.1.	Définitions et concepts de base .....	20
1.3.3.2.	Fonctionnement continu par morceaux .....	20
1.3.3.3.	Réalisation et exemple de fonctionnement.....	21
1.3.3.3.1	Schéma de réalisation.....	21
1.3.3.3.2	Réalisation technologique .....	23
1.3.3.3.3	Exemple de fonctionnement.....	24
1.3.3.4.	Cas remarquables .....	25
1.3.3.4.1	Bloqueur d'Ordre Zéro (Zero Order Hold) .....	25
1.3.3.4.2	Echantillonneur généralisé (Generalized Sampled data Hold Function (GSHF)) .....	25
1.3.3.4.3	Système continu à état initial imposé.....	26
1.3.3.4.4	Système continu à état initial propre .....	26
1.3.3.5.	Conclusion.....	27
1.3.4.	Systèmes bi-échantillonnés à fonctionnement par morceaux .....	27
1.3.4.1.	Deux échelles de temps échantillonnées .....	27
1.3.4.2.	Définition générale.....	28
1.3.4.3.	Réalisation et exemple de fonctionnement.....	29
1.3.4.3.1	Schéma de Réalisation .....	29
1.3.4.3.2	Réalisation technologique .....	30
1.3.4.3.3	Exemple de fonctionnement.....	31
<b>1.4.</b>	<b>Conclusion.....</b>	<b>31</b>

## Chapitre 2

### Poursuite échantillonnée par contrôleur à fonctionnement par morceaux .....33

<b>2.1.</b>	<b>Introduction .....</b>	<b>34</b>
<b>2.2.</b>	<b>Contexte de travail .....</b>	<b>35</b>
2.2.1.	Introduction .....	35
2.2.2.	Architecture de poursuite par CFM.....	36
2.2.3.	Stratégie de poursuite échantillonnée.....	38

<b>2.3. Contrôleur Continu par Morceaux (CCM)</b> .....	<b>38</b>
2.3.1. CCM non optimisé.....	38
2.3.1.1. Mise en équations .....	38
2.3.1.2. Architecture du CCM non optimisé.....	40
2.3.1.3. Conditions d'existence de $M(A, B, \gamma, \alpha, T_e)^{-1}$ .....	41
2.3.1.4. Validation expérimentale.....	43
2.3.2. CCM optimisé.....	46
2.3.2.1. Mise en équation.....	47
2.3.2.2. Identification des paramètres du CCM optimisé .....	47
2.3.2.3. Architecture du CCM optimisé.....	49
2.3.2.3.1. Calculs préliminaires .....	49
2.3.2.3.2. Structure du contrôleur optimisé .....	50
2.3.2.4. Condition d'existence de $\Theta_{12}^{-1}$ .....	50
2.3.2.4.1. Bloc-triangularisation de H .....	50
2.3.2.4.2. Interprétation .....	51
2.3.2.4.3. Calcul de $\tilde{\Theta}_{12}$ .....	52
2.3.2.5. Validations expérimentales.....	54
<b>2.4. Contrôleur bi-échantillonné</b> .....	<b>55</b>
2.4.1. CBE non optimisé.....	56
2.4.1.1. Mise en équation.....	56
2.4.1.2. Condition d'existence de $M^{-1}$ .....	57
2.4.1.3. Architecture du CBE non optimisé.....	57
2.4.1.4. Validation expérimentale.....	58
2.4.2. CBE optimisé.....	60
2.4.2.1. Mise en équation.....	60
2.4.2.2. Calcul de la condition initiale du CBE optimisé $\lambda_i^0$ .....	61
2.4.2.3. Condition d'existence de $\Theta_{12}^{-1}$ .....	62
2.4.2.4. Architecture du CBE optimisé.....	62
2.4.2.5. Validation expérimentale.....	63
2.4.3. Conclusion .....	64
<b>2.5. Poursuite de sortie : Contrôleur Récursif (CR)</b> .....	<b>65</b>
2.5.1. CCM non optimisé : adaptation à une consigne de sortie .....	65
2.5.2. Contrôleur Récursif (CR) .....	66
2.5.2.1. Mise en équation.....	66

2.5.2.2.	Interprétation algorithmique.....	67
2.5.2.3.	Architecture du CR.....	67
2.5.3.	CR avec amortisseur.....	68
2.5.3.1.	Amortisseur .....	68
2.5.3.2.	CR avec amortisseur intégré .....	69
2.5.3.2.1.	Stratégie de commande .....	69
2.5.3.2.2.	Architecture de commande.....	70
2.5.4.	Performances du CR avec amortisseur intégré .....	70
2.5.4.1.	Poursuite d'un processus linéaire ou changeant .....	70
2.5.4.1.1.	Poursuite d'un processus linéaire avec ou sans amortisseur.....	70
2.5.4.1.2.	Validation de la robustesse : poursuite d'un processus à paramètres changeants .....	72
2.5.4.2.	Linéarisation d'un processus non linéaire.....	73
2.5.4.2.1.	Architecture de linéarisation .....	74
2.5.4.2.2.	Résultats de simulation.....	74
2.5.4.3.	Application au système réel : chariot x-y motorisé.....	75
2.5.4.3.1.	Poursuite sans modèle .....	76
2.5.4.3.2.	Validation de la méthode de linéarisation.....	79
<b>2.6.</b>	<b>Conclusion.....</b>	<b>81</b>

## Chapitre 3

### Observateur continu par morceaux : application aux systèmes visuels.....83

<b>3.1.</b>	<b>Introduction .....</b>	<b>84</b>
<b>3.2.</b>	<b>Formulation du problème.....</b>	<b>86</b>
<b>3.3.</b>	<b>Synthèse d'un observateur linéaire.....</b>	<b>87</b>
3.3.1.	Introduction à l'observateur .....	87
3.3.2.	Observateur prédictif de Seuret [Seu 06] .....	88
<b>3.4.</b>	<b>OCM, à partir de la sortie retardée et échantillonnée .....</b>	<b>90</b>
3.4.1.	Mise en équation .....	90
3.4.1.1.	Première étape : utilisation du premier SLCM (I).....	90
3.4.1.2.	Deuxième étape : application de l'Observateur Discret de Luenberger d'Ordre Réduit (ODLOR) .....	90
3.4.1.3.	Troisième étape : utilisation du deuxième SLCM (II) .....	91
3.4.2.	Architecture de réalisation .....	91

3.4.2.1.	OCM à partir de la sortie retardé et échantillonnée $y_{i-1}^k$ .....	91
3.4.2.2.	OCM à partir de la sortie retardée $y(t-T_e)$ .....	92
3.4.2.3.	OCM à partir de l'état retardé et échantillonné $x_{i-1}^k$ .....	93
3.4.2.4.	OCM à partir de l'état retardé $x(t-T_e)$ .....	94
3.4.3.	Validations expérimentales.....	94
3.4.3.1.	Sans bruit de mesure.....	94
3.4.3.2.	Avec bruit de mesure (bruit blanc).....	96
3.4.3.3.	Avec bruit de mesure ( $a_1 \sin(bt) + a_0$ ).....	97
3.4.3.4.	Comparaison avec l'observateur de Seuret [Seu 06] .....	98
3.4.4.	Conclusion.....	99
<b>3.5.</b>	<b>Adaptation de l'OCM à la poursuite à partir de la sortie retardée et échantillonnée.....</b>	<b>99</b>
<b>3.6.</b>	<b>Application à la poursuite d'un chariot x-y motorisé observé par vision artificielle.....</b>	<b>100</b>
3.6.1.	Formulation du problème .....	100
3.6.2.	Adaptation de l'OCM dans le cas où $p = 1$ .....	101
3.6.3.	Architecture de commande.....	102
3.6.4.	Validation expérimentale.....	102
<b>3.7.</b>	<b>Conclusion .....</b>	<b>106</b>

## Chapitre 4

### Commande d'un pendule inversé 2D/3D par asservissement visuel ..... 107

<b>4.1.</b>	<b>Introduction .....</b>	<b>108</b>
<b>4.2.</b>	<b>Commande d'un pendule inversé 2D.....</b>	<b>110</b>
4.2.1.	Formulation du problème .....	110
4.2.2.	Architecture de commande.....	111
4.2.3.	Boucle d'observation du pendule inversé.....	112
4.2.3.1.	Linéarisation et stabilisation du pendule inversé.....	112
4.2.3.2.	Observateur continu par morceaux.....	113
4.2.4.	Boucle de stabilisation.....	115
4.2.5.	Validations expérimentales.....	116
4.2.5.1.	Résultats de simulation.....	116
4.2.5.1.1.	Sans perturbation .....	116

4.2.5.1.2.	Avec perturbations .....	118
4.2.5.2.	Application au système réel .....	120
4.2.5.2.1.	Algorithme de démarrage.....	120
4.2.5.2.2.	Résultats temps réel.....	121
<b>4.3.</b>	<b>Modélisation et commande d'un pendule inversé 3D .....</b>	<b>124</b>
4.3.1.	Méthodes de modélisation classiques [Yan 00] .....	124
4.3.2.	Modélisation et commande .....	126
4.3.2.1.	Modélisation.....	127
4.3.2.1.1.	Pendule inversé 3D par chariot x-y.....	127
4.3.2.1.2.	Traitement des signaux visuels .....	128
4.3.2.2.	Architecture de commande.....	129
4.3.3.	Validations expérimentales .....	130
4.3.3.1.	Sans perturbation.....	130
4.3.3.2.	Avec perturbations .....	133
<b>4.4.</b>	<b>Conclusion.....</b>	<b>135</b>

## **Conclusion générale et perspectives .....137**

<b>Conclusion générale .....</b>	<b>137</b>
<b>Perspectives de recherche.....</b>	<b>141</b>
Théoriques.....	141
Applicatives.....	141
Pédagogiques.....	141

## **Annexe I**

### **Plate-forme de pendule inversé 3D à retour visuel ....143**

<b>I.1. Plate-forme expérimentale de pendule inversé 3D à retour visuel .....</b>	<b>144</b>
I.1.1 Système mécanique chariot-pendule.....	144
I.1.2 Système de vision.....	145
I.1.3 Contrôleur.....	145
I.1.4 Actionneurs .....	145
<b>I.2. Modélisation du système de pendule inversé 3D .....</b>	<b>146</b>
I.2.1 Chariot motorisé x-y.....	146
I.2.2 Pendule inversé 2D.....	147

I.2.3	Pendule inversé 3D .....	148
I.2.4	Traitement des signaux visuels.....	148
I.2.5	Conclusion .....	149

## **Annexe II**

### **Systèmes mécaniques sous actionnés à déphasage non minimal et hypothèse de quasi-stationnarité..... 151**

II.1	Systèmes mécaniques sous actionnés à déphasage non minimal.....	151
II.2	Hypothèse de quasi-stationnarité .....	153

## **Annexe III**

### **Algorithme de démarrage ..... 155**

## **Annexe IV**

### **A. Bibliographie personnelle..... 157**

Revue internationale .....	157
Chapitre d'ouvrage .....	157
Conférences internationales avec actes & comité de lecture.....	157
Dissertation de mémoire .....	158

### **B. Bibliographie ..... 159**



# Remerciements

*Tous d'abord, j'exprime ma profonde reconnaissance à tous les membres du jury, pour l'attention qu'ils ont manifestée, particulièrement au Professeur Jacques LOTTIN (SYMME, EPUS) et au Professeur Ahmed EL HAJJAJI (UPJV, Amiens), en acceptant de rapporter sur ce mémoire de doctorat.*

*Je remercie le Professeur Christian VASSEUR (LAGIS, USTL), en tant que Directeur, que collaborateur et surtout à titre personnel, pour son accueil, son dévouement, ses conseils et son soutien tout au long de mon parcours de doctorant. Sa contribution m'a été très bénéfique dans le lancement de ma carrière de recherche et d'enseignement. Travailler avec lui est un réel plaisir. Je n'oublierai jamais les moments passés en sa compagnie que ce soit de nature professionnelle ou personnelle.*

*Je remercie également le Professeur Vladan KONCAR (GEMTEX, ENSAIT), en tant que Co-Directeur, qui a contribué énormément à l'avancement du travail scientifique comme collaborateur et qui m'a toujours encouragé lors de cette expérience de doctorat. Je n'oublierai jamais mes premières expériences d'enseignement passées sous sa direction pour les élèves ingénieurs apprentis.*

*Mes remerciements vont au Professeur Nicolai CHRISTOV (LAGIS, USTL), qui a toujours su trouver du temps à me consacrer pour partager mes problèmes et apporter ses conseils.*

*J'exprime ma gratitude au Docteur Afzal CHAMROO (Maître de Conférences, LAII, UP-ESIP) pour m'avoir accompagné et soutenu dès le début de mon doctorat et pour avoir toujours été quelqu'un sur qui je puisse compter.*

*Mes pensées vont aux doctorants du LAGIS avec qui j'ai partagé cette formidable aventure du doctorat. Je pense particulièrement à Ilaitisvery Jacques MADIOMANANA, Yanqing YANG, Yara BACHALANY, Samer RIACHY, Hexuan HU pour les bons moments passés en leur compagnie, sans oublier les Docteurs Habiboulaye BOUBACAR et Salah ZIDI pour leurs soutiens et encouragements réguliers.*

*Je remercie aussi tous mes collègues permanents du LAGIS avec qui je partage une vie sociale chaleureuse. Je pense notamment à Mesdames Annick PIGNON, Margot DESCAMPS et Patricia DEWYNTER, à Messieurs Frédéric DURAK, Philippe HENIN et Daniel FARGUE, aux responsables de la direction du laboratoire, particulièrement au Professeur Olivier COLOT et à tout le corps des enseignants–chercheurs.*

*Mes pensées vont également aux membres du conseil de laboratoire que je rencontre mensuellement depuis la deuxième année de ma thèse, aux étudiants et aux stagiaires qui m'ont permis d'affirmer mes compétences pédagogiques : un sujet est toujours mieux « maîtrisé » par soi quand on arrive à le transmettre à autrui avec succès.*

*Enfin, je souhaite exprimer toute ma gratitude à mes parents, beaux parents et quatre sœurs sans qui je n'aurais pas pu faire mes études universitaires en France, ... sans oublier, bien sûr, ma femme Yang, qui me soutient avec amour.*

# Avant-propos

*Ce mémoire représente l'aboutissement de trois ans de travail de recherche effectué au Laboratoire d'Automatique, Génie Informatique & Signal, LAGIS (UMR CNRS 8146) – Université des Sciences et Technologies de Lille, USTL, sous la direction de messieurs les Professeurs Christian VASSEUR et Vladan KONCAR. Les résultats scientifiques sont le fruit d'une collaboration avec ces derniers et plusieurs autres chercheurs et ont fait, par ailleurs, l'objet de plusieurs publications.*

*Je suis honoré de pouvoir présenter ce manuscrit et d'apporter ainsi, une modeste contribution à la recherche dans des domaines vivants de l'automatique et l'informatique industrielle. En effet, même si beaucoup de concepts et de théories fondamentales ont déjà été présentés par nos prédécesseurs, il y a toujours de nouvelles problématiques à considérer et des adaptations à apporter aux théories existantes afin de mieux répondre à des besoins d'actualité, tels que l'intégration de nouvelles technologies dans le domaine de la commande, du traitement signal, des asservissements visuels et surtout de développer de nouvelles méthodes de commande efficaces et robustes de complexité réduite pouvant intéresser le monde industriel.*

*Ce mémoire propose, en utilisant une classe de systèmes hybrides (Systèmes à Fonctionnement par Morceaux) de réaliser la linéarisation, l'observation et la commande de processus réels et en particulier de systèmes mécaniques non linéaires sous actionnés à déphasage non minimal par asservissement visuel. Les méthodes développées n'ont pas la prétention de révolutionner le monde de l'automatique et de l'informatique industrielle, mais je serais très heureux si elles pouvaient satisfaire quelques besoins, et surtout ouvrir de nouvelles perspectives de recherche.*

*D'un point de vue plus personnel, cette aventure du doctorat, comprenant la recherche, l'enseignement, les responsabilités au laboratoire et surtout la rédaction d'articles et de ce mémoire, m'a apporté rigueur, méthode et organisation qui jouent positivement sur mon quotidien. J'espère pouvoir cultiver ces qualités tout au long de ma vie et les transmettre à mon entourage.*



# Abréviations et acronymes

AViVA	Attelage Virtuel pour Véhicules Autonomes
BOZ	Bloqueur d'Ordre Zéro
CBE	Contrôleur Bi-Échantillonné
CCD	Charged Couple Device
CCM	Contrôleur Continu par Morceaux
CFM	Contrôleur à Fonctionnement par Morceaux
CI	Condition(s) Initiale(s)
CNRS	Centre National de la Recherche Scientifique
CR	Contrôleur Récursif
EPUS	Ecole Polytechnique Universitaire de Savoie
ENSAIT	Ecole Nationale Supérieure des Arts et Industries Textiles
ESIP	Ecole Supérieure d'Ingénieurs de Poitiers
GEMTEX	GENie & Matériaux TEXtiles
GRAISyHM	Groupe de Recherche en Automatisation Intégrée et Systèmes Hommes-Machines
ID	Ingénierie de la Décision
LAGIS	Laboratoire d'Automatique, Génie Informatique et Signal
LAII	Laboratoire d'Automatique et d'Informatique Industrielle
SBE	Systèmes Bi-Echantillonnés
SC	Systèmes à Commutations
SCFM	Systèmes Continus à Fonctionnement par Morceaux
SEFM	Systèmes Echantillonnés à Fonctionnement par Morceaux
SLACM	Systèmes Linéaires Affines Constants par Morceaux
SLS	Systèmes Linéaires à Saut
SS	Systèmes à Sauts
SYMME	Laboratoire SYstèmes et Matériaux pour la MEcatronique

TAT	Technologies Avancées pour les Transports
UMR	Unité Mixte de Recherche
UPJV	Université de Picardie Jules Vernes
UP	Université de Poitiers
USTL	Université des Sciences et Technologies de Lille

# Introduction générale

*Cette thèse a été préparée au sein de l'équipe ID (Ingénierie de la Décision) du Laboratoire d'Automatique, Génie Informatique & Signal (LAGIS, UMR CNRS 8146). Ce travail de doctorat s'est inscrit dans le cadre du projet AViVA<sup>1</sup> (Attelage Virtuel pour Véhicules Autonomes), se situant lui-même dans le programme CPER TAT 3.1 (Technologies Avancées pour les Transports) soutenu par la région Nord – Pas de Calais, l'état et la communauté européenne (contrat 15010/02Y0064/03-04 CAR/Presage N° 4605 Obj. 2-2004 : 2 - 4.1 - N° 160/4605).*

*La recherche est consacrée au développement de nouveaux algorithmes de complexité réduite pour la commande des systèmes réels utilisant un capteur « immatériel » (par exemple une caméra) dans la boucle de rétroaction. Une plateforme 2D<sup>2</sup> d'asservissement en position par retour immatériel a été utilisée pour valider les aspects théoriques ou pratiques de nos algorithmes. Ce type d'asservissement peut être adapté à l'« attelage virtuel » qui consiste à commander un véhicule « esclave » de sorte qu'il suive un véhicule « meneur » observé par vision artificielle. Plus généralement, nos algorithmes peuvent être intégrés dans des projets de robotique ou de production industrielle.*

---

<sup>1</sup> AViVA concerne le développement de concepts de routes et de véhicules intelligents afin d'optimiser la circulation routière, améliorer la sécurité, préserver l'environnement et proposer de nouveaux services aux usagers. Ce projet a été soutenu par 10 laboratoires et organismes de recherche de la région Nord – Pas de Calais appartenant à la fédération de recherche GRAISyHM (Groupement de Recherche en Automatisation Intégrée et Systèmes Hommes-Machines)<sup>o</sup>: <http://www.univ-valenciennes.fr/graisyhm/>.

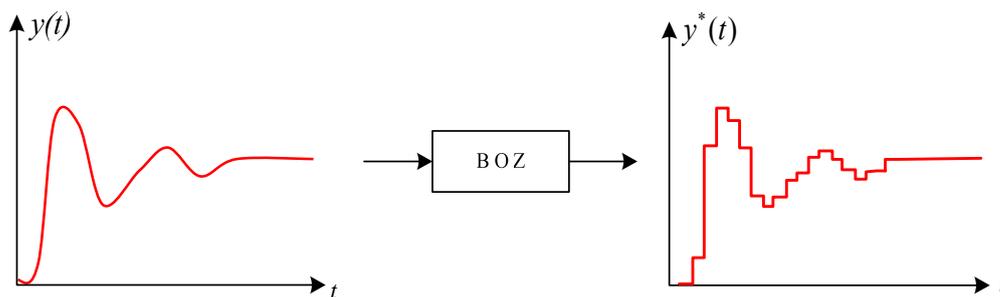
<sup>2</sup> Description en annexe I. Il s'agit d'un chariot x-y, qui peut être utilisé pour actionner un pendule inversé 2D ou un pendule inversé 3D.

## 0.1. Motivations

De nos jours, l'évolution des sciences et technologies de l'ingénieur a largement contribué au développement de capteurs numériques devenus incontournables dans les architectures de commande appliquées à de multiples domaines : transports, aéronautique, aide aux handicapés, commande embarquée, etc.

Bien souvent, les capteurs numériques sont gouvernés par des microprocesseurs, configurables en fonctions de traitement du signal. De tels capteurs dits « intelligents » introduisent, par nature, une discrétisation des données, pour laquelle on définit une période d'échantillonnage  $t_e$ . La figure 0.1 illustre l'effet d'un l'échantillonnage accompagné d'un blocage d'ordre zéro (BOZ).

Dans de nombreuses applications il peut s'avérer impossible de mettre le capteur en contact direct avec l'objet commandé. Les systèmes de vision artificielle<sup>3</sup> permettent de pallier cette contrainte physique. De plus, ce genre de capteur à distance permet de réaliser sans danger des tâches en environnement hostile ou très pénible pour l'homme



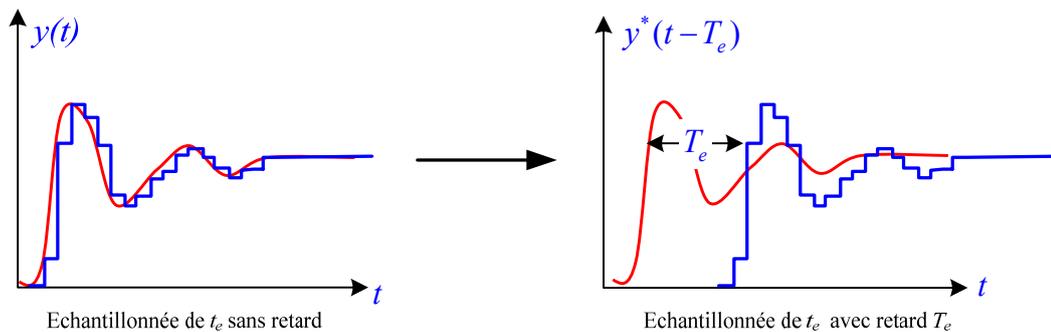
**Fig. 0.1.** Effet du Blocage d'Ordre Zéro sur un signal échantillonné à la période  $t_e$

(déminage, intervention dans un milieu radioactif, exploitation planétaire), d'admettre des utilisateurs délocalisés (enseignement, surveillance, chirurgie, etc..), de faire collaborer plusieurs applications d'un système informatique distribué ou encore de contrôler la qualité de fabrication dans les chaîne de production.

Dans [Hut 96, Khe 02, Mat 04, Chau 07], 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 un outil de rétroaction, pour la réalisation d'asservissements.

<sup>3</sup> La vision artificielle est un capteur permettant de mesurer une grandeur sur un objet sans entrer en contact avec celui-ci.

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 le coût est considéré comme prohibitif. Enfin, la vitesse de calcul du microprocesseur étant limitée, les temps cumulés de prise d'images et de traitement de ces images peuvent devenir non négligeables vis à vis de la dynamique du système à commander. Il convient alors de tenir compte du retard  $T_e$  induit. La figure 0.2 illustre cet aspect.



**Fig. 0.2.** Effet d'un retard sur la sortie discrétisée

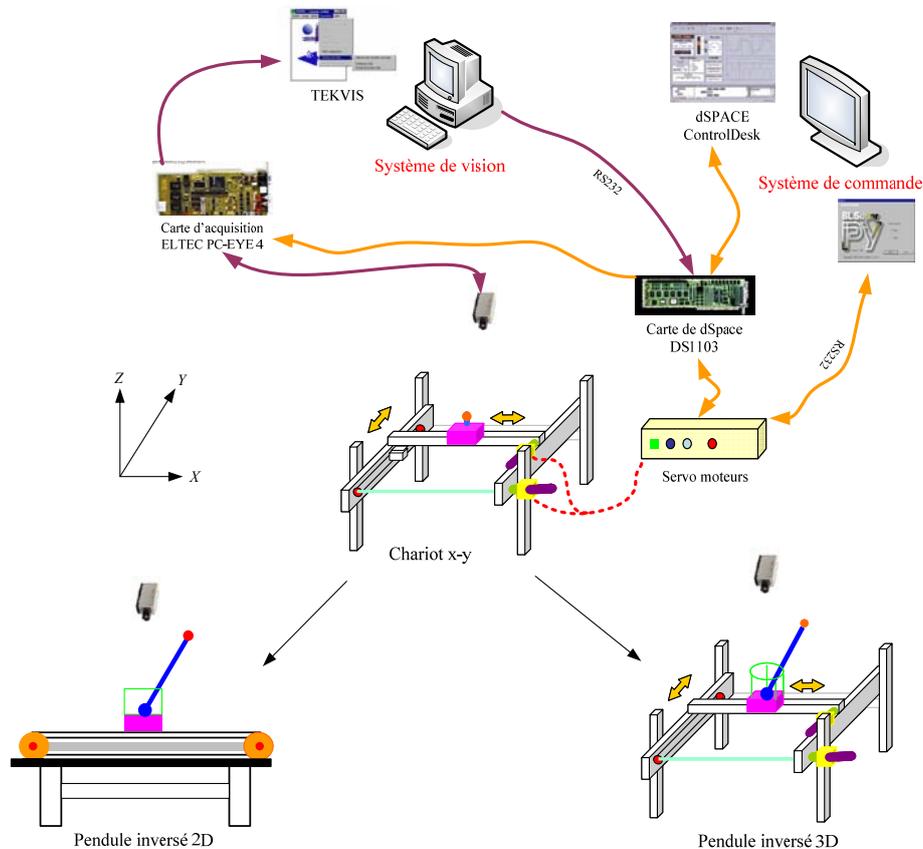
L'utilisation de la sortie retardée et échantillonnée  $y^*(t - T_e)$  peut poser des problèmes dans l'élaboration de la commande. De plus, intuitivement, on peut penser que plus on dispose d'informations sur le système, plus l'élaboration d'une loi de commande sera aisée. Une bonne commande passe donc, en général, par une bonne information sur le processus à commander. Les grandeurs mesurées ne couvrant généralement pas la totalité des grandeurs susceptibles de décrire complètement le comportement du processus, on peut se poser le problème de la reconstruction de l'information non directement mesurée à partir de celle mesurée : c'est le rôle de l'observateur ou estimateur d'état.

## 0.2. Problématique et approche

Dans ce mémoire, nous nous intéressons plus particulièrement aux capteurs intégrés dans des boucles de rétroaction sans contact (le système de vision dans notre cas). Dans ce contexte, deux verrous méthodologiques doivent être considérés :

- l'information est échantillonnée (durée de la trame de caméra par exemple),
- l'information subit un retard égal au cycle de calcul du capteur (traitement d'une séquence d'images) augmenté éventuellement du temps de transmission entre le système de vision et le système de commande.

Afin de valider les aspects théoriques et/ou pratiques de nos algorithmes, nous disposons d'un banc d'essai d'asservissement visuel illustré figure 0.3.



**Fig. 0.3.** Banc d'essai d'asservissement visuel

Ce banc d'essai, qui découle de la configuration détaillée en annexe I, peut se décliner selon les trois applications suivantes :

- Chariot x-y pour lequel on dispose d'un retour d'information, soit par des codeurs de position/vitesse disponibles selon chaque axe, soit par un système de vision<sup>4</sup> placé au dessus du chariot et donnant les mesures de position sous forme échantillonnée et retardée,
- Pendule inversé 2D relié au chariot par une liaison en rotation à un degré de liberté. Dans ce cas, on dispose d'une combinaison hybride de retours d'informations : la position du chariot par un codeur selon l'axe utilisé et la position angulaire du pendule à partir du système de vision,
- Pendule inversé 3D relié au chariot par une liaison du type joint de Cardan. Dans ce cas la combinaison de retours d'informations est constituée des positions du chariot selon les axes x et y fournies par des codeurs et la position angulaire du pendule à partir du système de vision.

<sup>4</sup> Les mesures de vision sont considérées retardées et échantillonnées de 40 ms. Le système de vision est présenté en détail en annexe I.

Selon les applications, les différents types de retours disponibles sont les suivants :

- Retour par l'état continu (sans retard),
- Retour par la sortie continue (sans retard),
- Retour par l'état retardé et échantillonné,
- Retour par la sortie retardée et échantillonnée,
- Retour par la sortie hybride avec une partie continue sans retard (obtenue par les codeurs) et une partie retardée et échantillonnée (système de vision), comme indiqué dans les configurations correspondant aux pendules inversés 2D et 3D.

Nous proposons de développer de nouveaux algorithmes de linéarisation, de poursuite, d'observation et de stabilisation, dans le but, non seulement de réaliser, en temps réel, la poursuite d'une consigne (chariot x-y dans notre cas) avec comme seule information la sortie du capteur numérique (sortie retardée et échantillonnée), mais aussi de réaliser en temps réel la stabilisation d'un pendule inversé 2D et même 3D avec un retour visuel.

Il est évident qu'il existe déjà de nombreuses études qui traitent de la linéarisation, de l'observation et de la commande des procédés réels, surtout pour les systèmes mécaniques. Toutefois, le problème de la commande d'un système mécanique non linéaire sous actionné et à déphasage non minimal reste largement ouvert. Même si beaucoup de chercheurs tentent de stabiliser le pendule inversé par retour visuel depuis la fin des années 1990 [Mag 98, Wen 00, Esp 06, Fuk 06], ce problème reste toujours d'actualité.

La problématique introduite par la plateforme 2D d'asservissement visuel, qui fait apparaître une dynamique continue et des mesures discrètes, suggère d'utiliser une commande hybride. Dans ce sens, nous proposons d'exploiter une classe particulière de systèmes hybrides définis par Vasseur et Koncar dans plusieurs articles [Kon 00, Kon 01, Kon 02] : les Systèmes à Fonctionnement par Morceaux (SFM).

Ces systèmes, qui font appel à deux espaces temps et à deux espaces d'entrée, l'un discret l'autre continu, sont caractérisés par des sauts de leur état à des instants discrets dits instants de commutation et par une évolution continue entre ces instants. Les SFM sont, par nature, particulièrement bien adaptés à la technologie numérique, surtout quand il s'agit de traiter des données discrétisées comme dans le cas des capteurs présentés précédemment.

### 0.3. Contributions

Les contributions théoriques que nous proposons dans ce mémoire intéressent plus particulièrement :

- La commande temps réel par SFM,
- La linéarisation d'un système réel par SFM,
- L'observation de l'état continu d'un système, à partir de sa sortie échantillonnée et retardée (données de la vision artificielle). Pour cela nous avons défini un « Observateur Continu par Morceaux » construit à l'aide d'un observateur Luenberger d'ordre réduit couplé à deux SFM,
- La commande hybride par SFM de systèmes mécaniques sous actionnés à déphasage non minimal, avec application à la stabilisation d'un pendule inversé 3D.

Il est important de souligner que ce mémoire ne vise pas à étudier l'observabilité, l'identifiabilité ou la commandabilité des systèmes hybrides eux-mêmes [Vid 02], mais bien d'utiliser les SFM pour linéariser, estimer, prédire ou commander des procédés réels surtout dans le domaine des asservissements visuels. Les systèmes hybrides utilisés sont conçus pour être facilement implantés en temps réel sur calculateurs numériques, selon une procédure systématique, que nous avons développée.

### 0.4. Organisation du mémoire

L'applicabilité et les performances des algorithmes qui émanent de nos recherches sont mises en évidence sur des exemples de simulation à l'aide du logiciel Matlab<sup>®</sup>-Simulink et sur le banc d'essai décrit précédemment.

Le mémoire est organisé de la façon suivante :

#### **Chapitre 1 : Des systèmes hybrides aux systèmes à fonctionnement par morceaux**

Ce chapitre présente un état de l'art des systèmes hybrides. Il permet au lecteur non coutumier de cette thématique d'appréhender les principales définitions et classifications liées aux systèmes hybrides. Une attention particulière est apportée aux Systèmes à Fonctionnement par Morceaux (SFM). Ces systèmes qui se caractérisent par deux espaces temps et deux espaces de commande, sont en effet fondamentaux pour la compréhension et le développement des algorithmes présentés aux chapitres suivants.

## **Chapitre 2 : Poursuite échantillonnée par contrôleur à fonctionnement par morceaux**

Une architecture particulière de commande utilisant deux boucles de retour sur les SFM est présentée. Sur la base du formalisme des SFM et sur le principe de cette architecture, nous présentons d'abord les « Contrôleurs à Fonctionnement par Morceaux » (CFM), qui assurent la poursuite échantillonnée d'une trajectoire d'état, à partir d'un retour sur l'état du processus à commander. A partir des CFM, nous proposons un ensemble d'améliorations telles que : l'« Amortisseur », dont le rôle est d'éviter une commande excessive en cas de discontinuité de la consigne et le « Contrôleur Récursif » (CR) assurant la poursuite d'une consigne de sortie à partir d'un retour sur la sortie continue du processus à commander. L'association du « Contrôleur Récursif » et de l'« Amortisseur » révèle des propriétés très intéressantes, notamment la possibilité d'imposer simplement un modèle libre à un processus réel non linéaire. Des exemples de simulation et d'implantation temps réel sur la plateforme d'essai illustrent l'extrême robustesse du « Contrôleur Récursif » face aux perturbations et aux variations des paramètres du processus.

## **Chapitre 3 : Observateur continu par morceaux : application aux asservissements visuels**

Un des objectifs de cette thèse est de développer et de proposer des méthodes d'observation, à partir de la sortie retardée et échantillonnée. Pour cela, nous introduisons les « Observateurs Continus par Morceaux » (OCM). Les OCM sont construits en associant l'Observateur de Luenberger Discret d'Ordre Réduit (OLDOR) à des SFM. Des types d'OCM sont proposés :

- les OCM utilisant la sortie (ou l'état) échantillonné(e).
- les OCM utilisant la sortie (ou l'état) retardé(e) et échantillonné(e).

Les architectures d'OCM proposées sont validées sur la version « chariot x-y + caméra » du banc d'essai. L'OCM a également été utilisé pour observer les systèmes non linéaires à déphasage non minimal (cf. Chapitre 4).

## **Chapitre 4 : Commande d'un pendule inversé 2D/3D par asservissement visuel**

Ce chapitre présente les principales contributions de la thèse. Il commence par une présentation des difficultés liées à la stabilisation du pendule inversé 3D par retour visuel : système instable sous actionné, fortement non linéaire avec un retour hybride (échantillonné + continu) et retardé. Ensuite, nous effectuons l'analyse des commandes existantes pour des systèmes mécaniques similaires. Cette étude nous permet de justifier l'approche que nous proposons, selon les deux étapes suivantes :

- Une commande de stabilisation du pendule inversé 2D utilisant deux boucles de retour : la première est une boucle interne, qui traite les informations échantillonnées et retardées délivrées par le système de vision. Le traitement vise à réaliser un modèle stable et linéaire du pendule sans aucune approximation et à en reconstruire l'état continu à l'aide d'un OCM. La seconde est une boucle externe visant à stabiliser l'ensemble chariot-pendule, selon une commande utilisant une fonction de Lyapunov. Les simulations et les essais sur plate forme réelle illustrent les performances de cette méthode.
- L'extension au pendule inversé 3D se fait en considérant qu'à chaque instant d'échantillonnage de la vision, la commande agit dans le plan du pendule. De cette manière nous transformons le problème 3D en un problème 2D résolu précédemment.

**La Conclusion générale et perspectives** du mémoire expose une synthèse des contributions de la thèse et présente les travaux de recherche futurs en lien direct avec la problématique traitée dans ce manuscrit.

# Chapitre 1

## Des systèmes hybrides aux systèmes à fonctionnement par morceaux

*La problématique présentée dans l'introduction générale fait appel à des systèmes possédant des propriétés hybrides. Dans cet esprit, ce chapitre est consacré à l'introduction des systèmes hybrides dans un sens général et dans le but de mieux préparer à la présentation d'une classe particulière de tels systèmes : les « Systèmes à Fonctionnement par Morceau  $x$  » (SFM) développés par l'équipe. Ces systèmes hybrides se caractérisent par deux espaces temps et deux espaces commandes. De manière générale, selon que la dynamique du système est continue ou discrète, on les appelle : « Systèmes Continus à Fonctionnement par Morceaux » (SCFM) ou « Systèmes Echantillonnés à Fonctionnement par Morceaux » (plus simplement « Systèmes Bi-Echantillonnés » (SBE)).*

## 1.1. Introduction

En automatique, les approches classiques de modélisation sont basées sur :

- Les modèles fréquentiels (fonctions de transfert issues de l'utilisation de la transformée de Laplace) pour les systèmes continus,
- Les modèles différentiels ou aux différences (représentation d'état : états continus ou discrets, à temps continu ou discret),
- Modèles à événements discrets (automates, réseaux de Petri) et modèles markoviens pour les systèmes événementiels.

Les méthodes d'analyse classiques prennent en compte un seul aspect à la fois, l'aspect continu, l'aspect discret ou l'aspect événementiel. Mais, la plupart des systèmes réels sont composés de sous-processus continus (moteurs, procédés chimiques, systèmes de freinage) et à événements discret (arrêt, redémarrage, reconfiguration des chaînes de production industriels ou des systèmes ferroviaires).

Un autre exemple quotidien et plus ludique est celui de la balle de tennis (ou de football), au cours d'un match. Dans ce cas, la position de la balle et sa vitesse constituent l'état d'un système à impulsions, qui subit des sauts (changements de direction et de vitesse) aux instants de frappe par les joueurs ou de rebond sur le terrain et qui évolue selon une dynamique continue entre ces instants. L'évolution d'un tel système est à la fois continue et discrète ou événementielle.

De fait, de nombreux systèmes réels mélangent les comportements continu et discret. Ainsi, pour assurer une bonne représentation de ces systèmes, il est nécessaire de prendre en compte les aspects continus et événementiels de leur dynamique ainsi que les interactions entre ces aspects. Les systèmes hybrides ont été introduits pour répondre à cette nécessité.

Par ailleurs, les innovations technologiques ont un impact considérable sur l'apparition de processus dynamiques hybrides. C'est le cas, par exemple, de la voiture hybride<sup>1</sup> (essence-électricité), dont le choix du type de fonctionnement est confié à un ordinateur de bord. Ce type de motorisation diminue de 10 à 50 % la consommation des véhicules et permet également de limiter les émissions polluantes. La motorisation hybride est donc particulièrement intéressante pour les taxis et elle commence à être appliquée aux autobus aux USA, en Europe et au Japon.

---

<sup>1</sup> On parle généralement d'automobile hybride pour désigner plus spécifiquement un engin automobile disposant de deux types de motorisation : moteur thermique et moteur électrique.

Depuis le début des années 1990, les automaticiens s'intéressent de plus en plus aux systèmes hybrides, non seulement pour modéliser des systèmes hybrides existants, mais aussi pour concevoir et construire des systèmes hybrides artificiels pouvant eux-mêmes être utilisés pour identifier, commander ou superviser des procédés industriels complexes.

Des nombreux problèmes mal traités par les approches homogènes sont résolus par l'approche hybride :

- Variations de structure liées aux différents modes de marche,
- Variations des paramètres lors de grandes transitions dans les régimes de fonctionnement,
- Commandes discontinues (par exemple par relais ou impulsions),
- Modélisation de phénomènes transitoires rapides par des commutations de modèles.

Les paragraphes suivants fournissent une présentation générale sur les systèmes hybrides existants (structures, définitions et classification) avant d'aborder une classe particulière : les « Systèmes à Fonctionnement par Morceaux » décrits en détail à la fin du chapitre. Toutes les méthodes proposées dans ce mémoire sont basées sur ce type de systèmes hybrides.

## **1.2. Systèmes hybrides : présentation générale**

### **1.2.1. Généralités sur les systèmes hybrides**

Dans la littérature, plusieurs méthodes de modélisation des systèmes hybrides ont été proposées. [Zay 01, Bir 06] envisagent trois classes de méthodes :

- Les méthodes basées sur une extension des techniques de modélisation des systèmes continus par le formalisme bond-graph mixte,
- Les méthodes basées sur une extension des techniques de modélisation des systèmes à événements discrets par les réseaux de Petri hybrides, les automates etc.,
- Les méthodes mixtes combinant à la fois le continu et le discret dans une structure unique. Chaque aspect est décrit sous une forme classique et l'aspect hybride est pris en compte dans l'interface entre les deux aspects.

Plus précisément, les systèmes hybrides présentés figure 1.1 sont caractérisés par l'interaction entre une partie continue décrite par des équations différentielles et une

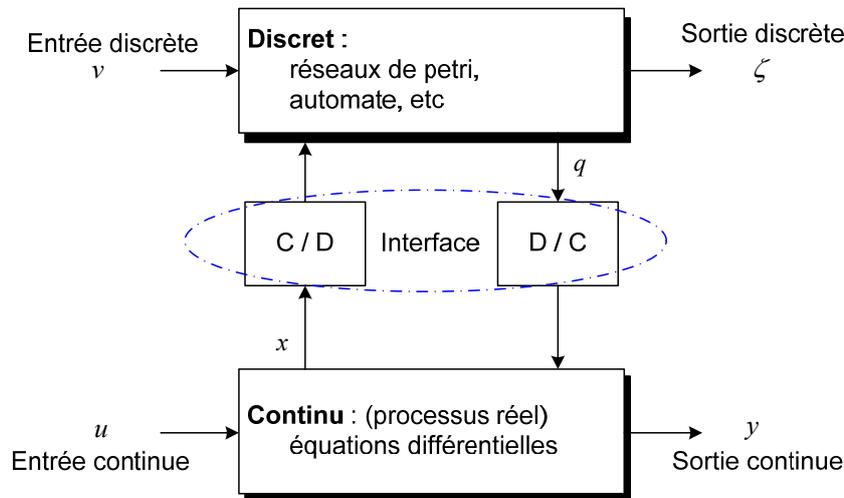


Fig. 1.1. Système hybride : structure générale

partie discrète décrite par des réseaux de Petri ou automates associés aux événements discrets.

Cette approche est inspirée du concept de systèmes hybrides développé par le pionnier Witsenhausen [Wit 66] et puis largement étudié par [Alu 93, Bac 93, Ner 93, Bra 94, Bra 95, Bra 98, Ant 98, Cur 98, Ye 98, Bem 99, Mic 99, Dec 00, Zay 01, Lib 03, Bir 06].

Chaque Partie du système hybride dispose d'entrées (continue  $u$ , discrète  $v$ ) et de sorties (continue  $y$ , discrète  $\zeta$ ) hybrides. Par l'intermédiaire de l'interface Continu-Discret (C/D), les signaux provenant de la partie continue déterminent des événements discrets provoquant des transitions d'un état discret à un autre. Inversement, l'état discret  $q$  détermine la dynamique spécifique de la partie continue du système hybride par l'interface Discret-Continu (D/C). Les interfaces C/D et D/C permettent donc d'interagir entre des variables continues et des variables discrètes à des instants dits de commutations, lorsque les variables continues atteignent des valeurs spécifiques prédéfinies.

Par exemple, un système non linéaire peut être modélisé par un ensemble de systèmes linéaires évoluant chacun dans une zone de fonctionnement spécifique. Il suffit alors de commuter judicieusement par des tests logiques entre ces modèles linéaires pour atteindre un modèle global pouvant mieux approcher la dynamique du système non linéaire [Ant 98].

Il est important de souligner qu'il existe de nombreuses commandes de type hybride qui, généralement, combinent le contrôle de variables continues avec des tests logiques de la forme "si-alors-sinon" et des propositions logiques permettant de mieux cerner le

comportement du processus à commander. Cette approche induit des propriétés nouvelles conduisant, par exemple, à un plus grand domaine de stabilité et/ou à une meilleure qualité des performances de commande.

### 1.2.2. Définition de SANTIS

Dans [San 03, Bir 06], Santis donne une définition formelle des systèmes hybrides. Cette définition nous semble suffisamment générale pour englober une très large classe de Système Hybrides. Malgré sa complexité, elle permet de mieux comprendre les classes existantes de Systèmes Hybrides et, notamment, les Systèmes à Fonctionnement par Morceaux définis par l'équipe et ainsi de mieux positionner nos recherches dans le contexte des Systèmes Hybrides.

**Définition 1.1.** [San 03, Bir 06] *Un système hybride en temps continu (respectivement en temps discret) est défini comme :*

$$H = (Q, V, \zeta, X, U, Y, Init, \mathfrak{S}_c, \mathfrak{S}_m, T, E, Inv, G, Rst), \quad (1.1)$$

(Respectivement  $H = (Q, V, \zeta, X, U, Y, Init, S_d, S_m, T, E, Inv, G, Rst)$ )

où :

- $Q = \{q_i, i \in J\}$  est un ensemble fini d'états discrets,  $J \in \mathbb{N}$ .
- $V = V_{Ext} \cup V_{Con} \cup \{\varepsilon_v\}$  est un ensemble fini d'entrées discrètes, avec  $\varepsilon_v$  l'événement d'entrée "vide".
- $\zeta = \{\zeta_i, i \in J\} \cup \{\varepsilon_z\}$  est un ensemble fini de sorties discrètes, avec  $\varepsilon_z$  la sortie discrète "vide" (utilisée pour désigner les événements de sortie non observable).
- $X, U, Y$ , sont des vecteurs de sous-espaces de dimensions finis. Ils représentent respectivement l'état continu, l'entrée continue et à la sortie continue du système. Pour chaque  $x \in X$ ,  $u \in U$  et un  $q \in Q$ , la sortie continue est telle que  $y = h(x, q, u) \in Y$ , avec  $h: X \times U \times Q \rightarrow Y$ .
- $Init \in X \times Q$  est l'ensemble des états initiaux.
- $\mathfrak{S}_c$  est une sous-classe de systèmes dynamiques en temps continu (respectivement  $S_d$  en temps discret).  $\mathfrak{S}_i \in \mathfrak{S}_c$  est définie par l'équation en temps continu comme :

$$\dot{x}(t) = f_i(x(t), u(t)) \quad i \in J,$$

où :  $t \in \mathbb{R}$ ,  $x \in X$  et  $f_i(\cdot)$  est une fonction telle que  $\forall u \in U \quad i \in J$ , sa solution  $x(t)$  existe et est unique pour chaque  $t \in \mathbb{R}$ .

(Respectivement,  $S_i \in S_d$  est définie par l'équation suivante :

$$x(k+1) = f_i(x(k), u(k)) \quad i \in J$$

où :  $k \in \mathbb{Z}$ ,  $x \in X$ )

- $\mathfrak{S}_m : Q \rightarrow \mathfrak{S}_c$  (respectivement  $S_m : Q \rightarrow S_d$ ) est une application qui associe à chaque état discret  $q_i \in Q$  une dynamique d'état en temps continu (respectivement en temps discret).
- $T \in Q \times V \times Q$  est l'ensemble de transitions discrètes. Nous notons par  $(q_i, v, q_j)$  une transition de l'état discret  $q_i$  à l'état discret  $q_j$  provoquée par l'arrivée de l'entrée discrète  $v$ .
- $E : \mathfrak{S}_m \rightarrow \zeta$  est une fonction qui associe à chaque transition une sortie discrète.
- $Inv : Q \rightarrow 2^{X \times V \times U}$  est une collection de domaines d'invariant.
- $G : T \rightarrow 2^{X \times V \times U}$  est une collection de domaine de conditions de garde.
- $Rst : T \times X \times U \rightarrow 2^X$  une collection de fonction d'actualisation (où d'initialisation) qui permettent, lors du franchissement d'une transition discrète, d'actualiser la variable d'état continu à  $x = Rst(\cdot, x)$ .

Les transitions discrètes définies par  $T$  peuvent être de différentes natures selon les trois types d'entrées discrètes :

- Des transitions de commutation contrôlables : elles sont déterminées par des entrées externes contrôlables  $v \in V_{Con}$  : par exemple, une loi de commande générée par un opérateur, un ordinateur ou une horloge (ajustement de la période de commutation),
- Des transitions de commutation non contrôlables, qui sont forcées par des entrées discrètes externes  $v \in V_{Ext}$  non contrôlables : par exemple, une perturbation externe,
- Des transitions de commutation d'invariant, qui se produisent lorsque la condition d'invariant n'est plus respectée  $v \in \{\varepsilon_v\}$  : par exemple, quand la valeur d'état atteint un certain seuil.

Dans ce mémoire, nous ne considérons que les transitions non contrôlables. Par ailleurs, les transitions d'invariant sont considérées comme des transitions autonomes.

### 1.2.3. Quelques classes particulières de systèmes hybrides

Pour simplifier la présentation, nous nous limitons aux systèmes hybrides dynamiques en temps continu. Pour les systèmes hybrides dynamiques en temps discret, il est facile d'extrapoler en remplaçant simplement les équations différentielles par des équations aux différences.

### 1.2.3.1. Systèmes à Commutations (Swiched Systems)

Les Systèmes à Commutations (SC) sont caractérisés par une évolution continue associée à un ensemble de sous-systèmes et une évolution discrète associée à des entrées contrôlables  $V_{Con}$ . Ces entrées sont souvent le résultat d'une stratégie de commande discrète.

Ainsi, selon la définition (2.1), un SC peut être défini par :

$$H_{SC} = (Q, V_{Con}, \{\varepsilon_\zeta\}, X, U, Y, Init, \mathfrak{T}_c, \mathfrak{T}_m, T, \{\varepsilon\}, Inv, G, Rst) \quad (1.2)$$

avec  $Q = \{1, 2, \dots, J\}$  ensemble fini d'états discrets.

- L'ensemble des transitions  $T$  est composé uniquement de transitions de type contrôlable : donc  $V_{Ext} = \emptyset$  et il n'y a aucune transition d'invariant ( $Inv = R^n$ ).
- Lors du franchissement d'une transition aucune sortie discrète n'est accessible à la mesure  $\zeta = \{\varepsilon_\zeta\}$  et  $E(T) = \{\varepsilon\}$ .

Les espaces d'état, de sortie et d'entrée sont tel que  $X \in R^n$ ,  $Y \in R^m$  et  $U \in R^r$ .

Pour chaque  $i \in Q$  un sous-système dynamique indexé par  $i$  est défini comme suit :

$$\begin{cases} \dot{x}(t) = f_i(x(t), u(t)) \\ y(t) = g_i(x(t), u(t)) \end{cases} \quad (1.3)$$

La valeur de la variable d'état continu  $x$  n'est pas réactualisée lors d'une commutation  $Rst(\cdot, x) = x$ .

Dans le cas linéaire, ces systèmes sont appelés Systèmes Linéaires à Commutations (SLC) et le système (1.3) est remplacé par :

$$\begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{cases} \quad (1.4)$$

avec :  $A_i \in R^{n \times n}$ ,  $B_i \in R^{n \times r}$  et  $C_i \in R^{m \times n}$ .

### 1.2.3.2. Systèmes à Sauts (Jump Systems)

Les Systèmes à Sauts (SS) se caractérisent par un état hybride  $(q, x)$  composé d'un état discret  $q$  dont l'évolution est régie par des entrées discrètes non contrôlables (inconnues) et d'un état continu  $x$  dont l'évolution est régie par une équation différentielle sans entrée

(système autonome). Les transitions associées aux SS sont uniquement de type non contrôlable  $V_{Ext}$ .

Dans le cadre de la définition générale, un système hybride SS est donc défini par :

$$H_{SS} = (Q, V_{Ext}, \{\varepsilon_\zeta\}, X, \{\varepsilon_U\}, Y, Init, \mathfrak{S}_c, \mathfrak{S}_m, T, \{\varepsilon\}, Inv, G, Rst) \quad (1.5)$$

avec  $Q = \{1, 2, \dots, J\}$  ensemble fini d'états discret,  $J$  étant le nombre de sous-systèmes. Cet indice est utilisé pour désigner le sous-système en cours d'évolution.

- Les entrées discrètes sont caractérisées par  $V_{Con} = \emptyset$ , car aucune entrée discrète n'est contrôlable.
- L'ensemble des sorties discrètes  $\zeta = \{\varepsilon_\zeta\}$  se résume à l'événement non observable parce qu'aucune sortie discrète n'est accessible à la mesure.
- Les espaces d'état, de sortie et d'entrée sont tels que  $X \in R^n$ ,  $Y \in R^m$  et  $U = \{\varepsilon_U\} = 0$ .

Pour chaque  $i \in Q$  un sous-système dynamique linéaire indexé par  $i$  est défini comme suit :

$$\begin{cases} \dot{x}(t) = f_i(x(t)) \\ y(t) = g_i(x(t)) \end{cases} \quad (1.6)$$

Les transitions  $T$  étant toutes de type non contrôlable, les domaines d'invariant et de garde sont caractérisés par  $Inv(\cdot) = R^n$ ,  $G(\cdot) = R^n$ . Lors du franchissement des transitions aucune sortie discrète mesurable n'est émise  $E = \{\varepsilon\}$  et aucune réactualisation du vecteur d'état n'est effectuée  $Rst(\cdot, x) = x$ .

De la même manière, on définit les Systèmes Linéaires à Saut (SLS), en remplaçant le système (1.5) par le suivant :

$$\begin{cases} \dot{x}(t) = A_i x(t) \\ y(t) = C_i x(t) \end{cases} \quad (1.7)$$

où  $A_i \in R^{n \times n}$ ,  $C_i \in R^{m \times n}$ .

### 1.2.3.3. Systèmes Linéaires Affines Constants par Morceaux (Piecewise Affine Systems)

Pour les Systèmes Linéaires Affines Constants par Morceaux (SLACM), l'évolution de l'état continu est régie par un système dynamique linéaire par morceaux et l'évolution de l'état discret est régie par des transitions de type invariant  $v \in \{\varepsilon_v\}$ . Il est important de souligner que, dans littérature, beaucoup de systèmes hybrides sont définis comme étant régi par des transitions de type invariant  $v \in \{\varepsilon_v\}$  [Wit 66, Zab 73, Ben 84, Bai 89, Bac 93, Bra 94, Bra 95, Bra 98, Tor 04, Bor 05].

En particulier Branicky [Bra 94, Bra 95, Bra 98, Bra 98a] a proposé de classer les systèmes hybrides à commutation autonome en trois classes : systèmes hybrides autonomes, contrôlés et à impulsion.

Selon la définition (2.1), un système SLACM peut être défini par :

$$H_{SLACM} = (Q, \{\varepsilon_v\}, \{\varepsilon_\zeta\}, X, U, Y, Init, \mathfrak{S}_c, \mathfrak{S}_m, T, \{\varepsilon\}, Inv, G, Rst) \quad (1.8)$$

avec  $Q = \{1, 2, \dots, J\}$  ensemble de modes d'évolution.

- Le franchissement des transitions ne produit aucune sortie discrète accessible à la mesure  $\zeta = \{\varepsilon_\zeta\}$  et  $E(T) = \{\varepsilon\}$ .
- L'invariant  $Inv = G$  est défini pour tout  $i, j \in Q$  comme :

$$Inv(i) \subset Q \times X, \text{ tel que } Inv(i) \cap Inv(j) = \emptyset$$

Pour chaque  $i \in Q$ , un sous-système linéaire indexé par  $i$  est actif :

$$\begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) + f_i \\ y(t) = C_i x(t) + g_i \end{cases} \quad (1.9)$$

où  $A_i \in R^{n \times n}$ ,  $B_i \in R^{n \times r}$ ,  $f_i \in R^n$ ,  $C_i \in R^{m \times n}$  et  $g_i \in R^m$ . Le vecteur d'état n'est pas réactualisé lors des commutations  $Rst(\cdot, x) = x$ .

Un SLACM évolue donc suivant plusieurs dynamiques différentes. Chaque dynamique est définie dans une région de l'espace d'état-entrée. Ces dernière sont obtenues à partir de la condition d'invariant  $Inv(i)$  qui permet de partitionner l'espace d'état-entrée en  $J$  régions  $\mathfrak{N}_i, i = 1, \dots, J$ . L'état initial est  $x_0 \in \mathfrak{N}_i$  et l'évolution de  $x$  est régie par le sous-système  $i$ . Lorsqu'à un instant donné  $x$  dépasse (ou atteint) la frontière séparant deux régions, l'évolution de  $x$  est régie par un nouveau sous système  $j$ .

On peut étendre l'équation (1.9) au cas non linéaire pour développer les Systèmes Hybrides Non Linéaire Affines en la Commande ou les Systèmes Hybrides Non Linéaire Affines en l'Etat.

## 1.3. Systèmes à fonctionnement par morceaux

### 1.3.1. Introduction

Depuis la fin des années 1970, des chercheurs lillois [Lau 72] [Vas 72] puis [Kon 00, Kon 01, Kon 02, Kon03] s'intéressent à une classe particulière de systèmes hybrides appelés Systèmes à Fonctionnement par Morceaux (SFM). Un SFM est un système strictement causal à dimension finie caractérisé par une commutation de son état en réponse à des impulsions contrôlées.

Un tel système possède deux espaces d'entrées ( $V^\sigma$  et  $U^r$ ) et se réfère à deux espaces temps :

- Le premier espace temps, nécessairement discret  $S = \{t_i, i = 0, 1, 2, \dots\}$  et  $S \subset \mathfrak{T}$  où  $\mathfrak{T}$  est un espace temps continu ou discret, constitue la référence des événements discrets choisis pour forcer l'état du SFM aux valeurs imposées par la première entrée  $V^\sigma$  de dimension  $\sigma$ .

$S$  est appelé *espace de commutation*.

- Le second espace temps  $\Phi_i = \{t \in \mathfrak{T} - S\}$ , constitue la référence de la dynamique du système en réponse au second espace d'entrée  $U^r$  de dimension  $r$ . Deux instants de commutation successifs de  $S$  délimitent un morceau  $q_i = ]t_i, t_{i+1}[$ .

### 1.3.2. Formalisme général

Selon que la dynamique du SFM est continue ou échantillonnée entre les instants de commutation, on parle de Systèmes Continus à Fonctionnement par Morceaux (SCFM) [Kon 01, Kon 03, Kon 06] ou de Systèmes Echantillonnés à Fonctionnement par Morceaux (SEFM) [Kon 02].

Dans les deux cas, la représentation d'état intra-morceau met en jeu les trois espaces vectoriels : l'espace d'état de dimension  $n$ , noté  $\Sigma^n$ , l'espace d'entrée de dimension  $r$ , noté  $U^r$  et l'espace de sortie de dimension  $m$ , noté  $Y^m$ . Si l'on note respectivement  $x(t) \in \Sigma^n$ ,  $u(t) \in U^r$  et  $y(t) \in Y^m$  l'état, l'entrée et la sortie en fonction du temps  $t \in \mathfrak{T}$ , alors la dynamique d'un SFM s'exprime selon les équations suivantes :

$$x(t_i^+) = s(t_i, v(t_i)), \quad \forall t_i \in S, \quad (1.10a)$$

$$x(t) = f(t, x(t_i^+), u_{]t_i, t_{i+1}[}), \quad \forall t \in q_i, \quad (1.10b)$$

$$y(t) = h(t, x(t), u(t)), \quad \forall t \in \Phi_t. \quad (1.10c)$$

dans lesquelles :

- L'équation (1.10a), appelée *équation de saut*, définit une transformation de la forme :

$$S \times V^\sigma \rightarrow \Sigma^n : \quad x(t_i^+) = s(t_i, v(t_i)), \quad \forall t_i \in S.$$

dans laquelle  $v(t_i) \in V^\sigma$  est la valeur de la première entrée à l'instant  $t_i$  et  $x(t_i^+)$  définit la valeur initiale de  $x(t)$ , au début du morceau  $q_i = ]t_i, t_{i+1}[$ .

- L'équation (1.10b), appelée *équation de transition d'état* traduit la causalité et constitue la solution d'une équation différentielle ou d'une équation de récurrence selon que l'espace temps utilisé est continu ou discret. Elle s'exprime dans chaque morceau  $q_i = ]t_i, t_{i+1}[$  par une transformation de la forme :

$$\mathfrak{S} \times \mathfrak{S} \times \Sigma^n \times U^r \rightarrow \Sigma^n : \quad x(t) = f(t, x(t_i^+), u_{]t_i, t_{i+1}[}), \quad \forall t \in q_i,$$

avec :  $u_{]t_i, t_{i+1}[}$  désignant l'entrée sur  $q_i = ]t_i, t_{i+1}[$ .

- L'équation (1.10c), appelée *équation de sortie* est, quant à elle, une équation sans mémoire donnée par la transformation suivante :

$$\mathfrak{S} \times \Sigma^n \times U^r \rightarrow Y^m : \quad y(t) = h(t, x(t), u(t)), \quad \forall t \in \Phi_t.$$

**Remarque 1.1.** Il est aussi possible d'envisager un changement de dynamique à chaque instant de commutation, selon une représentation multi modèle. Dans ce cas, les fonctions  $f(\cdot)$ ,  $s(\cdot)$  et  $h(\cdot)$  deviennent  $f_{q_i}(\cdot)$ ,  $s_{q_i}(\cdot)$  et  $h_{q_i}(\cdot)$  définies sur chaque morceau  $q_i$ .

**Remarque 1.2.** Par nature, les SFM font appel à des fonctions pouvant être discontinues aux instants de commutation  $t_i \in S$ . De ce fait, nous définissons, pour une fonction du temps  $g(t)$  :  $g(t_k^-) = g_k^-$  et  $g(t_k^+) = g_k^+$ . En général, quand une fonction  $g(t)$  est discontinue aux instants de commutation, on considère sa valeur juste après la commutation et simplifions l'écriture  $g_k^+$  par  $g_k$ . Bien entendu, en cas de continuité,  $g_k^- = g_k^+ = g_k$ . Enfin, dans le cas où l'on est amené à définir une commutation à période  $T_e$  constante, on a :  $S = \{ i.T_e, i = 0, 1, 2, \dots \}$  et  $g_k = g(kT_e)$ .

**Remarque 1.3.** *On peut également envisager le cas où les commutations sont générées par des transitions d'invariant, c-à-d qu'elles sont conditionnées par les valeurs d'état du système à commander (par exemple l'état atteint un certain seuil prédéfini).*

### 1.3.3. Systèmes continus à fonctionnement par morceaux

Pour faciliter la compréhension, dans le cas où  $\mathfrak{T}$  est un espace temps continu, on présente ci-après, le cas linéaire à modèle dynamique invariant correspondant aux Systèmes Linéaires Continus à fonctionnement par Morceaux (SLCM).

#### 1.3.3.1. Définitions et concepts de base

Un Système Linéaire Continu à fonctionnement par Morceaux (SLCM) est représenté par un quintuplet  $\Sigma_c(S, A, B_c, B_d, C)$  traduisant le système d'équations suivant :

$$x(t_i^+) = B_d v(t_i), \quad \forall t_i \in S \quad (1.11a)$$

$$\dot{x}(t) = Ax(t) + B_c u(t), \quad \forall t \in q_i \quad (1.11b)$$

$$y(t) = Cx(t), \quad \forall t \in \Phi_t \quad (1.11c)$$

dans lequel

- $S, v(t), q_i, \Phi_t, x(t_i^+), x(t), u(t)$  et  $y(t)$  ont été définis ci-dessus,
- $A \in \mathfrak{R}^{n \times n}, B_c \in \mathfrak{R}^{n \times r}, B_d \in \mathfrak{R}^{n \times \sigma}, C \in \mathfrak{R}^{m \times n}$  sont des matrices réelles,

L'équation (1.11a) donne la valeur de commutation de l'état à droite,  $\forall t_i \in S$ .

L'équation (1.11b) décrit, sous la forme d'une équation différentielle, la dynamique continue du SLCM sur chaque morceau  $q_i = ]t_i, t_{i+1}]$ .

L'équation (1.11c) représente, quant à elle, l'équation de sortie.

**Remarque 1.4.** *Le système d'équations (1.11) montre clairement le rôle des deux entrées du SLCM. L'entrée  $v(t)$  régit les sauts de l'état aux instants de commutation (1.11a), alors que l'entrée continue  $u(t)$  agit sur l'état entre ces instants (1.11b).*

**Remarque 1.5.** *Si on définit un espace de commutation régulière avec un période de commutation  $T_e$ , on a  $S = \{iT_e, i = 0, 1, 2, \dots\}$ . Dans ce cas, les instants de commutations sont indépendants de l'état du processus à commander.*

#### 1.3.3.2. Fonctionnement continu par morceaux

L'intégration du système (1.11) conduit à la solution :

$$x(t) = \exp(A(t-t_i))B_d v(t_i) + \int_{t_i}^t \exp(A(t-\tau))B_c u(\tau) d\tau, \quad \forall t \in q_i \quad (1.12)$$

La valeur de l'état  $x(t)$  à gauche de  $t_i$ , notée  $x(t_i^-)$ , est obtenue en écrivant l'équation (1.12) sur le morceau  $q_i = ]t_i, t_{i+1}]$  et en faisant tendre  $t$  vers  $t_i$  par valeurs inférieures, soit :

$$x(t_i^-) = \exp(A(t_i - t_{i-1}))B_d v(t_{i-1}) + \int_{t_{i-1}}^{t_i} \exp(A(t_i - \tau))B_c u(\tau) d\tau, \quad (1.13)$$

D'après l'équation (1.11a) aux instants de commutation  $\forall t_i \in S$  :

$$x(t_i^+) = B_d v(t_i), \quad (1.14)$$

En comparant l'équation (1.13) et l'équation (1.14), on a généralement :  $x(t_i^-) \neq x(t_i^+)$ .

À partir de ces équations et concepts de base, le comportement du fonctionnement d'un SLCM peut être illustré comme sur la figure 1.2 :

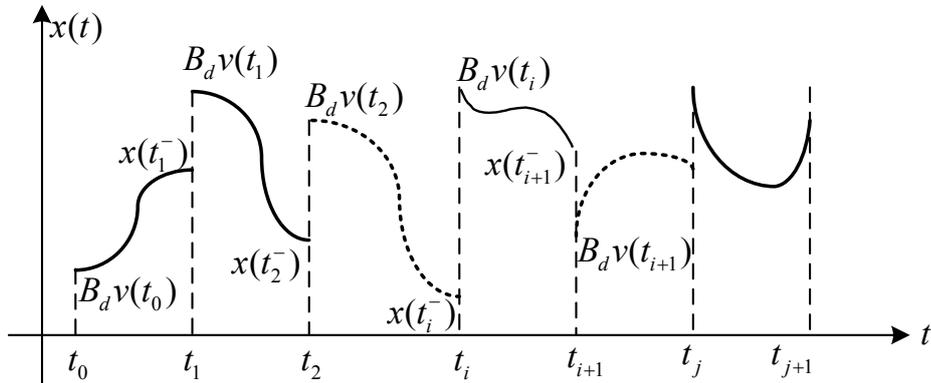


Fig. 1.2. Evolution de l'état d'un SLCM

### 1.3.3.3. Réalisation et exemple de fonctionnement

#### 1.3.3.3.1 Schéma de réalisation

Le schéma de réalisation d'un SLCM  $\Sigma_c(S, A, B_c, B_d, C)$  défini par le système d'équations (1.11) est directement interprétable sous la forme d'un schéma fonctionnel, ainsi que l'illustre la figure 1.3.

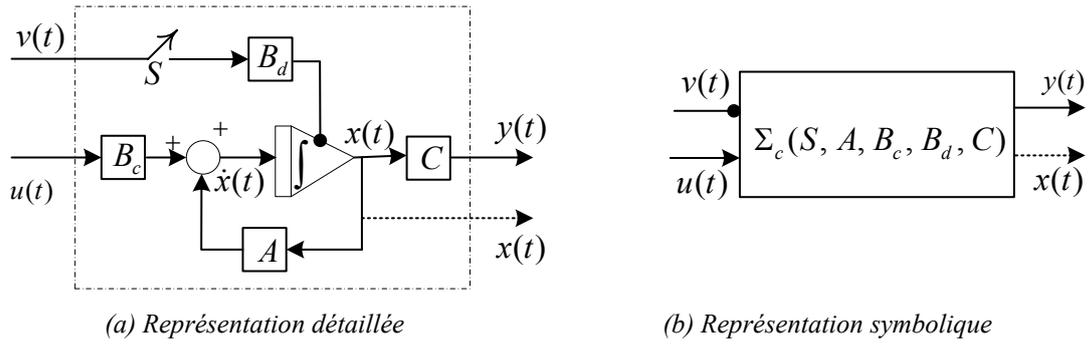


Fig. 1.3. Réalisation d'un SLCM

Sur la figure 1.3a, le schéma détaillé fait apparaître un intégrateur continu délivrant l'état  $x(t)$  et la sortie  $y(t)$ , selon la dynamique de l'équation (1.11b) : matrices  $A, B_c, C$ . La sortie de cet intégrateur peut être forcée à partir de la valeur fournie par la première entrée  $v(t)$  aux instants de commutation  $S$ , selon l'équation (1.14) :  $x(t_i^+) = B_d v(t_i)$ .

À partir du schéma détaillé de la figure 1.3a, on propose la représentation symbolique de la figure 1.3b pour représenter un SLCM définis par les équations (1.11).

En récapitulatif, le SLCM se caractérise donc par deux espaces de commandes selon deux espaces de temps :

- la première commande (entrée)  $v(t)$  repérée par un point marqueur, peut être continue. Elle est, de toute façon, discrétisée selon  $S$  (espace de commutation) avec  $S \subset \mathfrak{T}$  dans lequel  $\mathfrak{T}$  est un espace temps continu, afin d'obtenir sa commande discrète  $v(t_k)$ .
- la deuxième commande (entrée) continue  $u(t)$  repérée par une flèche agit entre deux instants de commutations successifs.

**Remarque 1.5.** *Le fonctionnement du SLCM n'est correct que si  $v(t)$  est définie bornée aux instants de communication  $S$  et  $u(t)$  est bornée entre les instants de commutation.*

*Pratiquement, cette condition est obtenue en imposant  $v(t)$  bornée et continue aux instants de commutation. Dans ces conditions, à chaque instant  $t_i$ , on commute forcément la sortie de l'intégrateur à la valeur  $x(t_i^+) = B_d v(t_i)$ , puis le système évolue à partir de  $x(t_i^+)$ , sous l'influence de la commande  $u(t)$  selon (1.12). En  $t_{i+1}$ , on commute à une nouvelle valeur  $x(t_{i+1}^+) = B_d v(t_{i+1})$ , et ainsi de suite (cf. figure 1.2.).*

*Comme dans les plupart des systèmes réels, pour des raisons économiques ou technologiques, on ne dispose que de la sortie  $y(t)$  du système et non pas de l'état  $x(t)$ .*

C'est la raison pour laquelle, le vecteur d'état  $x(t)$  est représenté par des pointillés sur la figure 1.3.

### 1.3.3.3.2 Réalisation technologique

Le schéma de la figure 1.3 est immédiatement implantable sur ordinateur. Par exemple à l'aide du logiciel Matlab/Simulink<sup>®</sup>, le schéma de réalisation technologique est illustré figure 1.4.

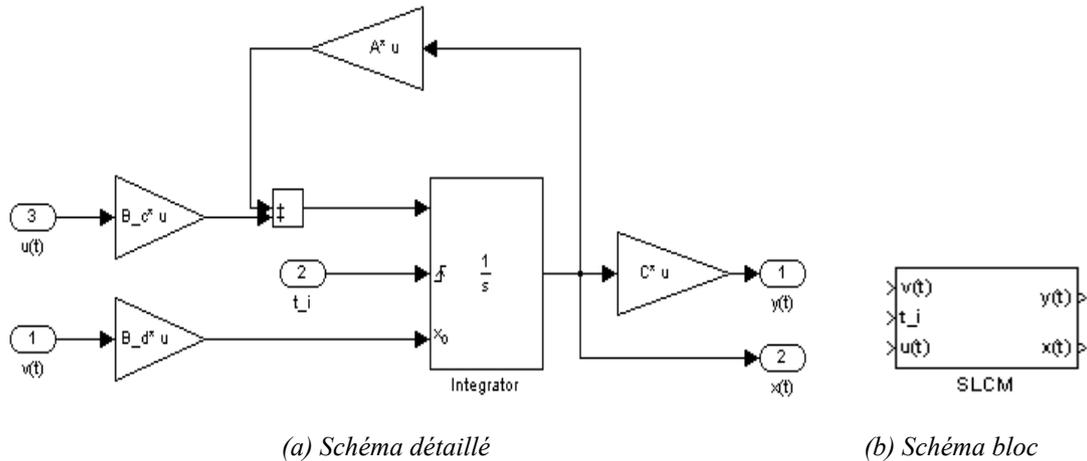


Fig. 1.4. Réalisation d'un SLCM sous Matlab/Simulink<sup>®</sup>

L'élément clé du SLCM est bien évidemment l'intégrateur qui doit être paramétré convenablement en mode *reset* pour que sa sortie puisse être réinitialisée à une valeur fournie par une entrée supplémentaire.

- le mode *reset* est déclenché sur le front montant/descendant (ou les deux) d'un signal d'horloge noté «  $t_i$  » sur la figure 1.4 pour forcer la sortie de l'intégrateur à une nouvelle condition initiale aux instants de commutation,
- le mode *externe* prend en compte cette nouvelle condition initiale à chaque *reset* par le biais d'une entrée supplémentaire.

Ainsi, ce bloc réalise l'intégration de l'entrée continue  $u(t)$  du SLCM à partir de la condition initiale<sup>2</sup> donnée par  $v(t)$  à chaque front du signal d'horloge.

Un signal carré de période constant  $T_e$  permet de générer les instants de commutation  $S = \{ iT_e, i = 0, 1, 2, \dots \}$ .

<sup>2</sup> Condition *initiale* définie à chaque commutation (équivalent à une *réinitialisation*).

**Remarque 1.5.** *On peut également déclencher les commutations, lorsqu'une condition logique est vérifiée : par exemple lorsque les données d'un capteur numérique sont prêtes. Cela permet de synchroniser l'évolution du système sur la prise d'information.*

1.3.3.3 Exemple de fonctionnement

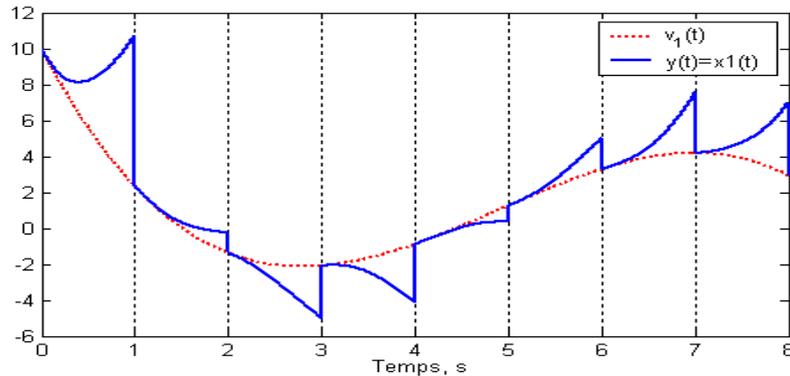
Soit le SLCM d'ordre 2 décrit par  $\Sigma_c(S, A, B_c, I_2, C)$  identique à celui décrit en [Kon 02] et [Wan 06a] :

$$A = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad C = [1 \quad 0] \quad \text{et} \quad S = \{iT_e, i=0,1,2,\dots\} \quad \text{où} \quad T_e = 1 \text{ s}$$

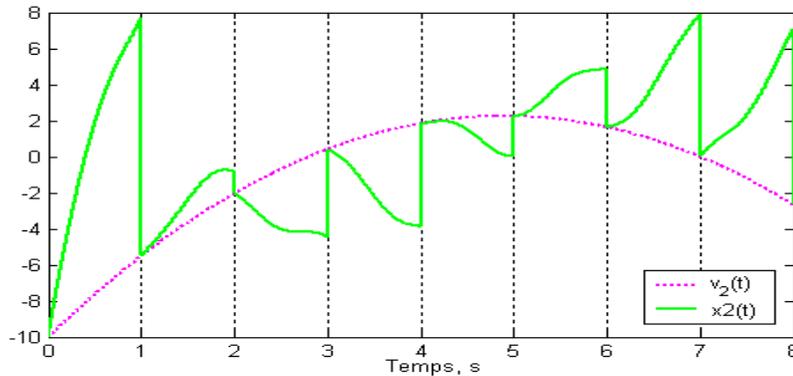
Aux instants de commutation on applique la première commande  $v(t)$  définie à partir de :

$$v(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} -0.17*t^3 + 2.5*t^2 - 10*t + 10 \\ -0.51*t^2 + 5*t - 10 \end{bmatrix}$$

La deuxième commande continue est définie par :  $u(t) = 2 \sin 5t$ .



(a) Evolution de la sortie  $y(t) = x_1(t)$  avec  $v_1(t)$



(b) Evolution de  $x_2(t)$  avec  $v_2(t)$

**Fig. 1.5.** Evolution de l'état du SLCM d'ordre 2 avec  $T_e=1s$

On peut constater sur la figure 1.5., que le système  $\Sigma_c(S, A, B_c, I_2, C)$  évolue de manière continue entre deux instants de commutation successif, à partir d'une condition initiale imposée par  $x(t_i^+) = B_d v(t_i)$ . L'état du système évolue par morceaux continus.

#### 1.3.3.4. Cas remarquables

Le SLCM standard étant défini par  $\Sigma_c(S, A, B_c, B_d, C)$ , nous donnons, dans ce paragraphe, quelques formes particulières à  $S$  et aux matrices  $A$ ,  $B_c$ ,  $B_d$  et  $C$ , afin de mettre en évidence quelques cas remarquables.

##### 1.3.3.4.1 Bloqueur d'Ordre Zéro (Zero Order Hold)

Le Bloqueur d'Ordre Zéro (BOZ) est réalisé en adoptant les valeurs suivantes :

$$A = 0, B_d = C = I_n \quad (\sigma = m = n).$$

De plus, on impose :  $u(t) = 0, \forall t \in \mathfrak{T}$ , ce qui est équivalent à  $B_c = 0$ . On n'utilise pas de commande continue, ainsi d'après les équations (1.11c) et (1.13), on obtient :

$$y(t) = v(t_i), \forall t \in q_i \cup t_i \in S.$$

Ainsi  $\Sigma_c(S, 0, 0, I_n, I_n)$  réalise l'échantillonnage blocage de  $v(t)$  selon les instants de commutation  $S$ . Si de plus, on définit  $S = \{ iT_e, i = 0, 1, 2, \dots \}$  où  $T_e$  est la période d'échantillonnage de commutation, alors  $\Sigma_c(S, 0, 0, I_n, I_n)$  est un BOZ à période constante.

##### 1.3.3.4.2 Echantillonneur généralisé (Generalized Sampled data Hold Function (GSHF))

Dans ce cas, on effectue le choix suivant :

$$S = \{ iT_e, i = 0, 1, 2, \dots \} \text{ où } T_e \text{ est la période de commutation,}$$

$$u(t) = 0, \forall t \in \mathfrak{T}, \text{ ce qui est équivalent à } B_c = 0,$$

En utilisant les équations (1.11c) et (1.13), on obtient  $y(t)$  sous la forme suivante :

$$y(t) = C \exp(A(t - iT_e)) B_d v_i, \quad \forall t, i$$

Afin d'interpréter ce résultat, posons :

$$F(t) = \sum_{i=0}^{\infty} \Gamma(t, i) \quad \forall t,$$

$$\text{avec } \Gamma(t, i) = \begin{cases} C \exp(A(t - iT_e)) B_d & \forall t \notin S \\ 0 & \text{ailleurs} \end{cases}$$

Il apparaît alors que  $F(t) = F(t + T_e)$  et que  $F(t)$  est intégrable.

Ainsi on a :

$$F(t) = \Gamma(t, i), \forall t \notin S$$

$$y(t) = \Gamma(t, i)v_i, \forall t \notin S$$

Donc  $F(t)$  est une matrice périodique, de période  $T_e$  et intégrable appartenant à  $R^{m \times \sigma}$ . Ceci implique  $\Sigma_c(S, A, 0, B_d, C)$  réalise un échantillonneur généralisé (Generalized Sampled data Hold Function (GSHF)) tel que défini par Kabamba [Kab 87].

#### 1.3.3.4.3 *Système continu à état initial imposé*

Dans ce cas, on définit un seul instant de commutation pour imposer l'état initial du système (instant  $t_0$ ). On a alors  $S = \{t_0\}$  et, en imposant  $B_d = I_n$  ( $\sigma = n$ ), on réalise un système continu évoluant à partir d'un état initial imposé  $x(t_0^+) = v(t_0)$ .

Selon les équations (1.11a) et (1.12), on a donc :

$$x(t) = \exp(A(t - t_0))v(t_0) + \int_{t_0}^t \exp(A(t - \tau))B_c u(\tau) d\tau, \quad \forall t > t_0 \quad (1.15)$$

Ainsi, un système continu à état initial imposé prend la forme  $\Sigma_c(\{t_0\}, A, B_c, I_n, C)$ .

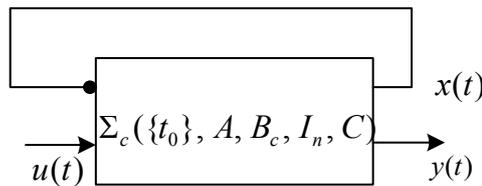
#### 1.3.3.4.4 *Système continu à état initial propre*

Dans le cas précédent, si l'on choisit :  $S = \{t_0\}$  et  $B_d = I_n$  ( $\sigma = n$ ), on a l'évolution d'état décrit par l'équation (1.15).

Si, en plus, on impose  $v(t) \equiv x(t)$ , on obtient l'évolution suivante :

$$x(t) = \exp(A(t - t_0))x(t_0) + \int_{t_0}^t \exp(A(t - \tau))B_c u(\tau) d\tau, \quad \forall t > t_0 \quad (1.16)$$

Ce cas particulier présenté figure 1.6, modélise un procédé continu réel.



**Fig. 1.6.** *Système continu à état initial propre*

### 1.3.3.5. Conclusion

La mise en œuvre, dans les SCFM, de deux espaces de commande ( $U^r$  et  $V^\sigma$ ) et de deux espaces temps ( $S$  et  $\Phi_t$ ) leur confère à la fois une nature échantillonnée et une nature continue.

En combinant les propriétés liées à cette doubles nature, il est possible, non seulement de modéliser des procédés réels, mais aussi de définir de nouvelles architectures de commande, que nous appelons *systèmes de commande continus (à fonctionnement) par morceaux* ou *contrôleurs continus (à fonctionnement) par morceaux*. Cet aspect sera abordé dans les chapitres suivants.

### 1.3.4. Systèmes bi-échantillonnés à fonctionnement par morceaux

Le second type des systèmes à fonctionnement par morceaux appelé Systèmes Echantillonnés à Fonctionnement par Morceaux (SEFM) ou simplement Systèmes Bi-Echantillonnés (SBE) a été proposé pour la première fois dans [Kon 02], à la suite des travaux sur les SCFM présentés dans [Kon 00] et [Kon 01].

#### 1.3.4.1. Deux échelles de temps échantillonnées

Pour ce type des systèmes, l'idée est d'utiliser deux échelles de temps discrètes, comme illustré figure 1.7.

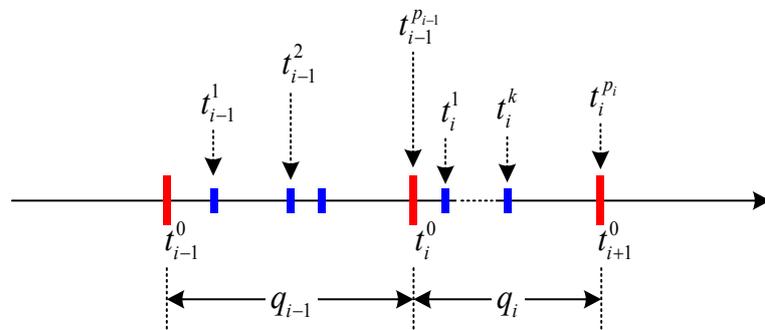


Fig. 1.7. Deux échelles de temps échantillonnées

Les instants discrets notés  $t_i^k$ , sont caractérisés par :

- $i$  : échelle de temps relative aux instants de commutation  $S = \{t_i^0, i = 0, 1, 2, \dots\}$ . Deux instants de commutation successifs tels que  $t_i^0$  et  $t_{i+1}^0$  délimitent un morceau noté  $q_i = \{t_i^0 \leq t_i^k < t_{i+1}^0, k = 1, 2, \dots, p_i - 1\}$ .
- $k$  : échelle de temps relative au fonctionnement du système entre deux instants de commutation, où le SBE évolue avec une période d'échantillonnage non fixée.

A l'intérieur du morceau  $q_i$ , l'indice  $i$  reste constant, tandis que l'exposant  $k$  croît de 0 à  $p_i - 1$ . Il existe, par conséquent,  $p_i$  échantillons de l'état dans un morceau  $q_i$ . Dans le cas général, la durée des morceaux n'est pas constante, de même que la période d'échantillonnage et le nombre d'instant d'échantillonnage à l'intérieur d'un morceau. Enfin, l'instant final du morceau  $q_{i-1}$  est confondu avec l'instant initial du morceau  $q_i$ . Ceci se traduit par la relation suivante :  $t_{i-1}^{p_{i-1}} = t_i^0$  (comme illustré figure 1.7).

Selon les deux échelles de temps échantillonnées, les SBE mettent en jeu deux espaces d'entrée : Le premier espace  $V^\sigma$  sert à définir la valeur à laquelle l'état est commuté  $\forall t \in S$ . Le second espace  $U^r$  nécessairement échantillonné selon les instants discrets  $t_i^k$ , détermine l'évolution de l'état.

#### 1.3.4.2. Définition générale

En s'inspirant de la définition des SCFM donnée section 1.3.3.3, on peut définir les systèmes non linéaires échantillonnés à fonctionnement par morceaux sur tout morceau  $q_i$  par le système d'équations suivant :

$$x(t_i^0) = s(t_i^0, v(t_i^0)), \quad \forall t_i^0 \in S \quad (1.17a)$$

$$x_i^{k+1} = f(t_i^k, x(t_i^k), u_i^k), \quad \forall t_i^k \in q_i \quad (1.17b)$$

$$y_i^k = h(t_i^k, x(t_i^k), u(t_i^k)), \quad \forall t_i^k. \quad (1.17c)$$

Dans lequel :

- $x_i^k \in R^n$  est le vecteur d'état avec  $k = 0, 1, 2 \dots p_i$ , et  $i = 0, 1, 2 \dots$
- $v(t) \in V^\sigma$  le vecteur de commande qui n'est pris en compte qu'aux instants de commutation  $S = \{t_i^0, i = 0, 1, 2 \dots\}$  pour définir l'état initial  $x_i^0$  pour chaque morceau.
- $u_i^k \in U^r$  est le vecteur de commande déterminant l'évolution de l'état à l'intérieur de chaque morceau.
- $y_i^k \in R^m$  est le vecteur de sortie,
- $s(\cdot)$ ,  $f(\cdot)$ ,  $g(\cdot)$  sont trois fonctions appropriées.

De manière plus générale, il est possible d'envisager un changement de modèle aux instants de commutation. Dans ce cas, il convient de modifier les équations précédentes en conséquence :

$$x(t_i^0) = s_{q_i}(t_i^0, v(t_i^0)), \quad \forall t_i^0 \in S \quad (1.18a)$$

$$x_i^{k+1} = f_{q_i}(t_i^k, x(t_i^0), u_i^k), \quad \forall t_i^k \in q_i \quad (1.18b)$$

$$y_i^k = h_{q_i}(t_i^k, x(t_i^k), u(t_i^k)), \quad \forall t_i^k. \quad (1.18c)$$

Si l'on remplace les trois fonctions  $f_{q_i}(\cdot)$ ,  $s_{q_i}(\cdot)$  et  $h_{q_i}(\cdot)$  par quatre matrices constantes  $F$ ,  $H$ ,  $H_d$  et  $G$  avec des dimensions appropriées, on réalise un système linéaire échantillonné à fonctionnement par morceaux (Simplement dit Systèmes Linéaires Bi-Echantillonnés (SLBE) présenté par  $\Sigma_d(S, F, H_c, H_d, G)$  décrit par les équations suivant dans un morceau  $q_i$  :

$$x_i^0 = H_d v_i^0, \quad \forall t_i^0 \in S \quad (1.19a)$$

$$x_i^{k+1} = F x_i^k + H_c u_i^k, \quad \forall t_i^k \in q_i \quad (1.19b)$$

$$y_i^k = G x_i^k, \quad \forall t_i^k. \quad (1.19c)$$

En générale,  $x_i^0 \neq x_{i-1}^{p_{i-1}}$ , ce qui traduit une discontinuité de fonctionnement lors de la transition d'un morceau à l'autre. L'évolution de l'état est illustrée figure 1.8.

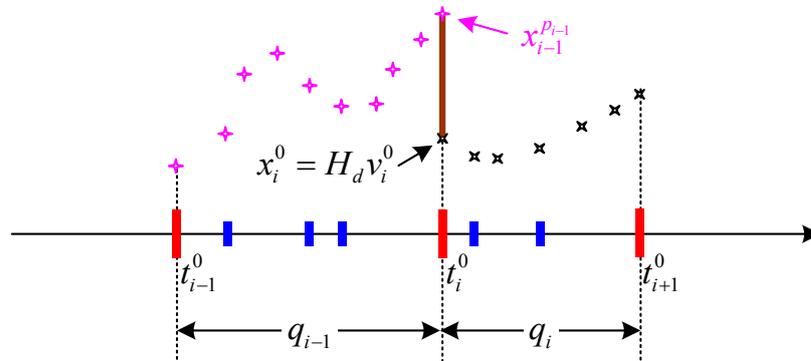


Fig. 1.8. Evolution d'état d'un SLBE

### 1.3.4.3. Réalisation et exemple de fonctionnement

#### 1.3.4.3.1 Schéma de Réalisation

Dans le cas particulier d'un SLBE défini par le système d'équations (1.19), on peut prendre la durée des morceaux constante  $T_e = t_{i+1}^0 - t_i^0, i = 0, 1, 2, \dots$ , de même que la période d'échantillonnage dans chaque morceau  $t_e = t_i^{k+1} - t_i^k, k = 0, 1, 2, \dots, p_i - 1$ . Dans ce cas, le nombre d'instant d'échantillonnage à l'intérieur d'un morceau est constant également,  $p_i = p$ . Ainsi la définition de ce SLBE particulier est directement interprétable sous la forme d'un schéma de réalisation, ainsi que l'illustre la figure 1.9.

Dans la figure 1.9a, nous introduisons un bloc «  $\frac{1}{z}$  » à sortie «réinitialisable » représentant un retard suivi de d'un BOZ. Le «  $\frac{1}{z}$  », cadencé selon l'échelle de temps  $i$ , est réinitialisé à une valeur fournie par une première entrée aux instants de commutation.

À partir du schéma détaillé de la figure 1.9a, nous proposons la représentation symbolique de la figure 1.9b, pour définir un SLBE particulier en choisissant  $S = \{iT_e, i = 0, 1, 2, \dots\}$  et  $q_i = \{t_i^k = iT_e + kt_e, k = 0, 1, 2, \dots, p-1\}$ .

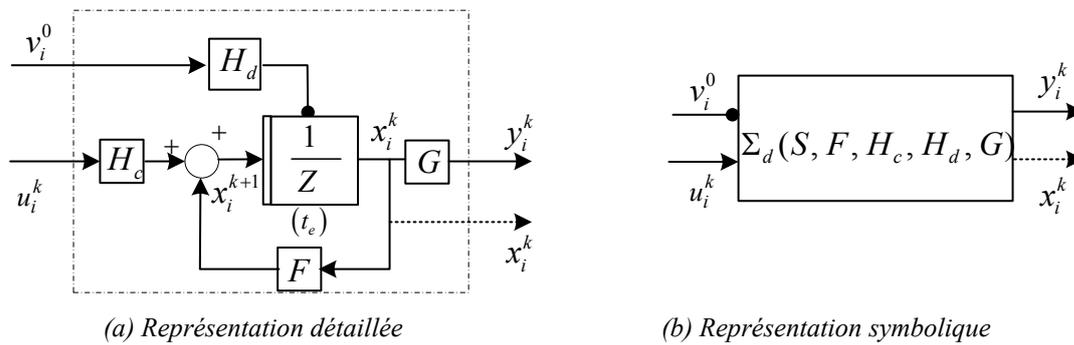


Fig. 1.9. Réalisation d'un SLBE

### 1.3.4.3.2 Réalisation technologique

Contrairement à l'intégrateur continu qui permet directement l'utilisation d'un mode *reset*, la bibliothèque de Simulink® ne dispose pas explicitement du bloc  $\frac{1}{z}$  en temps discret. Il faut donc construire ce bloc à partir d'un intégrateur discret qui admet le mode *reset* souhaité selon le montage donné figure 1.10.

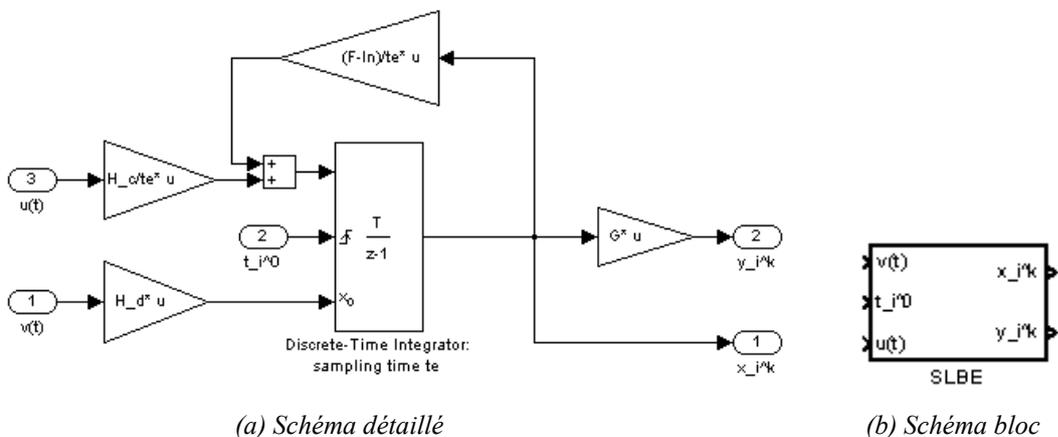


Fig. 1.10. Réalisation d'un SLBE sous Matlab/Simulink®

### 1.3.4.3.3 Exemple de fonctionnement

Reprenant l'exemple du SLCM défini de la section 1.3.3.3.3 par  $\Sigma_c(S, A, B_c, I_2, C)$  :

$$A = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, C = [1 \quad 0] \text{ et } S = \{iT_e, i=0,1,2,\dots\} \text{ où } T_e = 1 \text{ s}$$

On choisit  $\Phi_i = \{t_i^k = iT_e + kt_e, k=0,1,2,\dots,p-1\}$  et on réalise un SLBE défini par  $\Sigma_d(S, F, H_c, I_2, C)$  avec :  $F = \exp(AT_e)$ ,  $H_c = \int_0^{t_e} \exp(A(t_e - \tau))B_c d\tau$ ,  $t_e = 0.05 \text{ s}$  et  $S = \{i, i=0,1,2,\dots\}$ .

De plus, aux instants de commutation, on applique la même entrée  $v(t)$  définie par :

$$v(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} -0.17*t^3 + 2.5*t^2 - 10*t + 10 \\ -0.51*t^2 + 5*t - 10 \end{bmatrix} \text{ et l'entrée } u(t) = 2 \sin 5t.$$

On constate figure 1.11 que l'état évolue de manière échantillonnée selon (1.19b) sur chaque morceau  $q_i$  à partir d'une condition initiale imposée par  $v(t)$  aux instants de commutation. Des discontinuités apparaissent aux instants de commutation. Entre deux instants de commutation,  $p$  pas d'évolutions dans chaque morceau.

Trois valeurs différentes de  $p$  ont été testées (2, 4, 20). La figure montre clairement que quand  $p \rightarrow +\infty$ , l'évolution discrète de l'état d'un SLBE tend vers celle du SLCM correspondant.

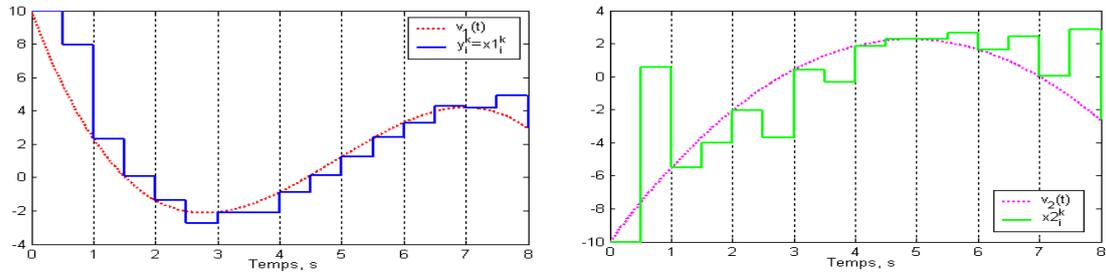
## 1.4. Conclusion

Dans ce chapitre, la définition de SANTIS nous a permis d'introduire la plupart des systèmes hybrides.

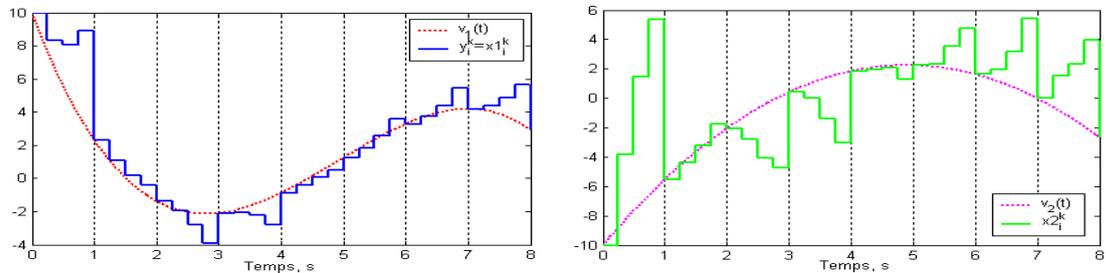
Nous avons ensuite introduit un type particulier de systèmes hybrides développé au LAGIS : les Systèmes à Fonctionnement par Morceaux (SFM) présentés dans [Kon 00] [Kon 01] [Kon 02] [Kon 03]. Notre étude se situe dans la continuité de ces travaux.

De manière générale selon que la dynamique du système est continue ou discrète, nous avons présentés deux types de SFM : les Systèmes Continus à Fonctionnement par Morceaux (SCFM) et les Systèmes Echantillonnés à Fonctionnement par Morceaux (plus simplement Systèmes Bi-Echantillonnés (SBE)).

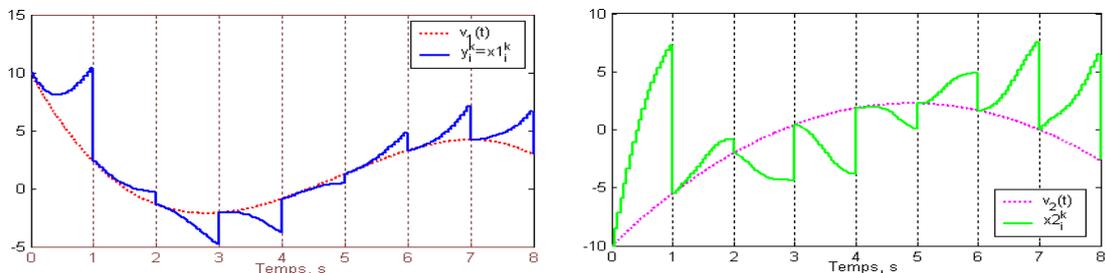
Dans les trois chapitres suivants, les SFM sont utilisés pour la modélisation et la linéarisation des systèmes, pour l'estimation de l'état et enfin pour la commande de systèmes mécaniques sous actionnés à déphasage non minimal.



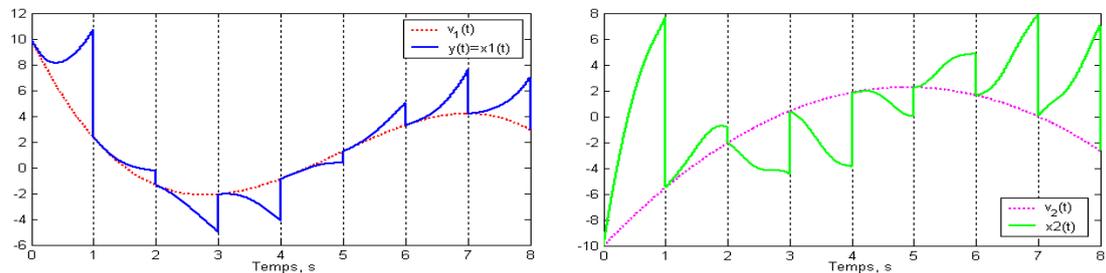
(a) SLBE ( $p=2$ )



(b) SLBE ( $p=4$ )



(d) SLBE ( $p=20$ )



(d) SLCM

Fig. 1.11. Evolution de l'état du SLBE particulier d'ordre 2 avec  $T_e=1s$  avec  $p$  différents

## Chapitre 2

# Poursuite échantillonnée par contrôleur à fonctionnement par morceaux

*Dans ce chapitre, une architecture particulière de commande utilisant deux boucles de retour à base de SFM est proposée. On réalise d'abord des Contrôleurs à Fonctionnement par Morceaux (CFM) qui garantissent la poursuite échantillonnée d'une trajectoire d'état à partir de l'état du processus à commander. Selon que la dynamique du CFM est continue ou échantillonnée entre les instants de commutation, on définit des Contrôleurs Continus à fonctionnement par Morceaux (CCM) ou des Contrôleurs Bi-Echantillonnés à fonctionnement par morceaux (CBE). Ensuite on développe un contrôleur spécifique appelé Contrôleur Récursif (CR) assurant la poursuite d'une trajectoire de sortie à partir de la sortie continue du processus à commander. Afin de prendre en compte des éventuelles discontinuités sur la consigne, on propose d'introduire un amortisseur qui permet d'éviter une commande excessive en cas de discontinuités sur la consigne. L'intégration de cet amortisseur dans l'architecture de commande CFM (surtout CCM), fait apparaître des propriétés intéressantes. Il peut être utilisé pour imposer à un système réel des paramètres libres choisis. Des exemples de simulation et d'implantation temps réel sur la plateforme de chariot x-y illustrent la robustesse des contrôleurs CR face aux perturbations et aux variations des paramètres du processus.*

## 2.1. Introduction

En pratique, si un système présente très peu d'imperfections (erreurs de modélisation, erreurs de mesures, ...) et s'il est très peu perturbé, la commande en boucle ouverte produit des résultats satisfaisants. Par contre, si ces imperfections et ces perturbations ne peuvent pas être négligées, même si elles sont petites, l'application d'une commande en boucle ouverte produit un écart important entre le résultat désiré et la réponse obtenue. Par exemple sur un robot mobile autonome, un léger glissement des roues sur le sol entraîne une dérive du véhicule pratiquement inévitable.

La commande en boucle fermée permet de rendre les systèmes moins sensibles aux perturbations, aux simplifications et approximations dans la modélisation, aux variations de paramètres du modèle ainsi qu'aux incertitudes sur les grandeurs physiques mesurées.

Ainsi la notion d'asservissement est très importante dans la commande des processus réels. On peut distinguer deux types d'asservissements :

- **La régulation** : dans ce cas, il s'agit d'amener le plus rapidement possible la sortie du processus à une valeur de consigne constante, tout en minimisant les dépassements et les oscillations transitoires et en maintenant cette valeur de consigne, même en présence de perturbations. La régulation doit également stabiliser le système. Les exemples de régulations sont nombreux : vitesse d'un véhicule, température d'un réfrigérateur, vitesse d'un ascenseur quel que soit le nombre de passagers, etc.
- **La poursuite** : dans ce cas, il s'agit de faire suivre à la sortie du système une consigne variable tout en assurant la stabilité. Exemple : faire suivre une cible mobile à un robot, le métro autonome de Lille, etc. Les exigences de la poursuite sont supérieures à celles de la régulation, surtout si la consigne admet des discontinuités et si les mesures des grandeurs associées au système subissent des retards importants résultant du fonctionnement des capteurs.

Plusieurs méthodes classiques d'asservissement garantissent la régulation, surtout pour les systèmes SISO invariants. La théorie du contrôle automatique fournit un éventail de solutions maintenant bien définies et maîtrisées, depuis les travaux de Bode [Bod 45], Horowitz [Hor 63], Fortmann [For 77], etc.

La poursuite, aussi appelée suivi de trajectoire, peut être traitée par les méthodes classiques : par exemple le régulateur PID pour des systèmes SISO. Mais ce correcteur atteint vite ses limites : en effet, il obéit à la loi de Bode, pour laquelle les effets d'amplitude et de phase sont couplés et antagonistes. De même, selon Landau [Lan 88],

un régulateur PID numérique ne peut s'appliquer rigoureusement qu'aux procédés modélisables par un système continu caractérisé par une fonction de transfert de degré maximal égal à 2. Afin de pallier aux limitations des méthodes classiques et dans l'optique de généraliser un formalisme adaptable à des processus MIMO, les méthodes de contrôle développées dans ce mémoire s'appuient plutôt sur une modélisation d'état du processus. Dans ce contexte, les deux méthodes les plus utilisées sont l'optimisation quadratique et le placement de pôles [Kai 80].

Dans la littérature, la poursuite de trajectoire est également traitée par la commande en mode glissant [Zha 03, Sal 04, Def 07], la commande adaptative selon la méthode de back stepping [Fan 05], la commande neuronale [Jia 07] ou la commande floue [Bou 00]. Même si ces méthodes permettent d'assurer la robustesse vis à vis des incertitudes et des perturbations, elles complexifient la synthèse de la commande. Par exemple, pour la commande en mode glissant, les discontinuités au voisinage de la surface induisant des dynamiques de haute fréquence appelées "réticences" constituent un inconvénient majeur.

Le type d'asservissement que nous envisageons dans ce chapitre s'inscrit dans le cadre de la poursuite de trajectoire pour les systèmes linéaires MIMO. Il s'agit ici de faire suivre une consigne changeante à la sortie ou à l'état du processus, en assurant la concordance à des instants d'échantillonnage prédéfinis correspondant aux instants de commutation dans la théorie des SFM. Selon la nature continue ou discrète du processus, on parle soit de *poursuite échantillonnée* soit de *poursuite bi-échantillonnée*.

Il est bien connu que tout processus continu peut être modélisé par un système échantillonné. Cependant, il faut noter que, contrairement à un système purement discret, le processus continu évolue également entre les instants d'échantillonnage. Les contrôleurs à fonctionnement par morceaux que nous proposons tiennent compte de cette évolution. De plus, selon que l'information de retour est utilisée uniquement aux instants d'échantillonnage ou aussi entre ces instants, on décline les méthodes de poursuite en deux versions : *sans* et *avec optimisation*. Dans ce dernier cas, on minimise l'erreur de poursuite entre les instants de commutation.

## 2.2. Contexte de travail

### 2.2.1. Introduction

Le principe des contrôleurs à fonctionnement par morceaux a été introduit pour la première fois dans les années 1970 par Laurent [Lau 72] et Vasseur [Vas 72] sous la dénomination de « commande composite des systèmes échantillonnés par impulsions multi modulées ».

D'autres auteurs ont apporté leur contribution à ce domaine de recherche : Parmi eux, Kabamba [Kab 87] a proposé de généraliser la notion d'échantillonneur pour la réalisation du contrôle des systèmes linéaires. L'idée est de générer la commande à partir d'une matrice périodique agissant sur le vecteur de sortie échantillonné (GSHF : Generalized Sample data Hold Function). On a montré au chapitre précédent comment formaliser le GSHF en utilisant le formalisme des SLCM. Par ailleurs, [Kim 81, O'Re 81, Wol 83, Uri 87] ont montré comment la commande *deadbeat* permet d'annuler l'erreur entre la sortie du processus à commander et sa consigne au bout d'un temps fini. Classiquement, ce type de commandes n'utilise le retour qu'aux instants d'échantillonnage. Ainsi, des oscillations sur la variable commandée entre ces instants peuvent se produire. Urikura et Nagata [Uri 87] ont proposé, quant à eux, une commande échantillonnée avec réduction des oscillations entre les instants d'échantillonnage. Plus récemment, Yamamoto [Yam 94] a utilisé le concept de fonction définie par morceaux pour réaliser un contrôle qu'il qualifie d'hybride et pour lequel l'état est pris en compte, non seulement aux instants d'échantillonnage, mais également entre ces instants. Il fait ressortir qu'une architecture de commande échantillonnée appliquée à un processus continu offre des propriétés de stabilisation irréalisables par une commande classique.

### 2.2.2. Architecture de poursuite par CFM

Les contrôleurs développés dans ce chapitre sont basés sur le formalisme des SFM linéaires et sont particulièrement bien adaptés à la réalisation d'une architecture de poursuite en temps réel.

Le processus à commander est un système linéaire continu modélisé, selon les notations de la section 1.3.3.4.4. « Système continu à état initial propre » :

$$\Sigma_s(\{t_0\}, A, B, I_n, C)$$

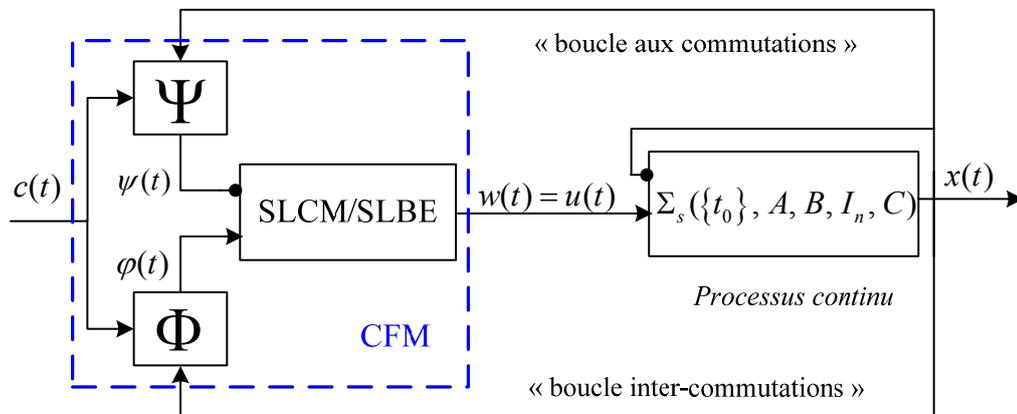
avec :  $A \in R^{n \times n}$ ,  $B \in R^{n \times r}$ ,  $B_d = I_n \in R^{n \times n}$ ,  $C \in R^{m \times n}$  : matrices usuelles de la représentation d'état.  $u(t) \in U^r$ ,  $x(t) \in \Sigma^n$  et  $y(t) \in Y^m$  désignent respectivement l'entrée, l'état et la sortie du processus.

**Définition 2.1.** Une architecture de commande CFM est définie par (figure 2.1) :

- un CFM linéaire (SLCM ou SLBE) d'espace de commutation  $S = \{t_i, i=0,1,2,\dots \mid t_{i+1} > t_i\}$ ,  $t_i \in \mathfrak{T}$ , avec  $\lambda(t)$ ,  $w(t)$ ,  $\psi(t)$  et  $\varphi(t)$  respectivement état, sortie, entrée à discrétiser (selon  $S$ ) et entrée continue. Les matrices et vecteurs apparaissant dans la définition du contrôleur sont de dimensions appropriées.
- deux boucles d'état :
  - la première boucle, appelée « boucle aux commutations », réalise une transformation  $\Psi(c(t), x(t))$  aux instants de commutations définis par  $S$ ;
  - la seconde boucle, appelée « boucle inter-commutations », réalise une transformation  $\Phi(c(t), x(t))$  entre les instants de commutations,

générant respectivement les entrées à discrétiser et continue du CFM linéaire, où  $c(t)$  et  $x(t)$  sont respectivement la consigne d'état et l'état du processus.

**Remarque 2.1.** Les consignes d'état d'un système dynamique à commander sont en réalité des points de passage imposés dans l'espace d'état de ce système.



**Fig. 2.1.** Architecture de poursuite par un CFM

Par ailleurs, selon que l'on utilise des SLCM ou des SLBE pour générer la commande, on parle de Contrôleurs Continus à fonctionnement par Morceaux (CCM) ou de Contrôleurs Bi-Echantillonnés à fonctionnement par morceaux (CBE).

De plus, selon que la « boucle inter-commutations » est utilisée ou non, on parle de CFM (CCM ou CBE) *optimisé* ou *non optimisé*. Dans ce dernier cas, le contrôleur fonctionne en *boucle ouverte* entre les instants de commutation.

**Hypothèse 2.1.** Nous supposons, pour la définition des contrôleurs, que le processus est commandable et observable.

### 2.2.3. Stratégie de poursuite échantillonnée

Dans cette partie nous imposons une stratégie de commande pour laquelle les CFM réalisent la poursuite échantillonnée selon l'équation suivante :

$$x((i+1)T_e) = c(iT_e), \quad \forall i = 0, 1, 2, \dots \quad (2.1)$$

avec  $T_e$  période d'échantillonnage constante correspondant aux instants de commutation du CFM.

En simplifiant, on note l'équation (2.1) :  $x_{i+1} = c_i$ .

L'équation (2.1) conduit à réaliser la poursuite échantillonnée de  $c(t)$  par  $x(t)$ , avec un retard d'une période d'échantillonnage  $T_e$ .

## 2.3. Contrôleur Continu par Morceaux (CCM)

Dans ce cas, le CFM est basé sur un SLCM. Un CCM standard représente un système associé dont l'état évolue de manière continue entre les instants de commutation. La sortie du contrôleur génère une commande continue par morceaux pour le processus à contrôler.

### 2.3.1. CCM non optimisé

Nous considérons ici, seulement la condition de poursuite présentée équation (2.1), sans imposer de conditions sur l'évolution de l'état du processus entre les instants d'échantillonnage. Dans ce cas, seule la boucle aux commutations est utilisée.

#### 2.3.1.1. Mise en équations

Afin de satisfaire la condition de poursuite (2.1), nous effectuons les choix suivants :

- Pour le contrôleur CCM, un SLCM  $\Sigma_c(S, \alpha, \beta_c, \beta_d, \gamma)$  avec :
  - des instants de commutation définis par  $S = \{iT_e, i = 0, 1, 2, \dots\}$  (période de commutation constante égale à  $T_e$ )
  - $\dim(\lambda(t)) = \dim(x(t)) = n$ ,
  - $\dim(w(t)) = \dim(u(t)) = r$ ,
  - $\dim(\alpha) = n \times n$ ,
  - $\dim(\gamma) = r \times n$  avec  $\gamma$  de rang plein,

- une entrée à discrétiser  $\psi(t)$  provenant de la fonctionnelle  $\Psi(c(t), x(t))$  définie précédemment,
- il n'y a pas de « boucle continue », soit  $\varphi(t) \equiv 0$  ou encore  $\beta_c = 0$ ;

Dans ces conditions, nous pouvons décrire le fonctionnement du système bouclé, sur le morceau  $q_i = ]t_i, t_{i+1}]$  avec  $i = 0, 1, 2, \dots$ , par le jeu d'équations suivant :

$$x'(t) = Ax(t) + Bu(t) \quad \forall t, \quad (2.2a)$$

$$\lambda_i^+ = \beta_d \psi_i \quad \forall i = 0, 1, 2, \dots, \quad (2.2b)$$

$$\lambda'(t) = \alpha \lambda(t) \quad \forall t \in q_i, \quad (2.2c)$$

$$u(t) = w(t) = \gamma \lambda(t) \quad \forall t \quad (2.2d)$$

L'évolution du processus continu est représentée par (2.2a) et celle du CCM est donnée par (2.2b et 2.2c). La connexion entre le CCM et le processus est donnée par (2.2d).

Selon (2.2a), nous pouvons décrire l'évolution du processus sur le morceau  $q_i$  à partir de sa condition initiale à l'instant  $t_i^+$  :

$$x(t) = \exp(A(t - t_i^+))x(t_i^+) + \int_{t_i^+}^t \exp(A(t - \tau))Bu(\tau)d\tau \quad \forall t \in q_i \quad (2.3)$$

Par ailleurs, selon (2.2c) l'évolution de l'état du contrôleur sur le même morceau  $q_i$  à partir de sa condition initiale à l'instant  $t_i^+$  est donnée par :

$$\lambda(t) = \exp(\alpha(t - t_i^+))\lambda(t_i^+) \quad \forall t \in q_i \quad (2.4)$$

Ainsi, en tenant en compte de l'expression de la commande  $u(t)$ , l'évolution du système bouclé conduit en  $t_{i+1}$  à :

$$x_{i+1} = \exp(AT_e)x_i + M(A, B, \gamma, \alpha, T_e)\lambda_i^+ \quad (2.5)$$

avec :

$$M(A, B, \gamma, \alpha, T_e) = \exp(AT_e) \int_0^{T_e} \exp(-A\tau)B\gamma \exp(\alpha\tau)d\tau, \quad M \in R^{n \times n} \quad (2.6)$$

À condition que  $M$  soit inversible, nous pouvons obtenir le fonctionnement désiré, soit  $x_{i+1} = c_i$ , en imposant, d'après l'équation (2.5) :

$$\lambda_i^+ = M^{-1} \{ c_i - \exp(AT_e)x_i \} \quad (2.7)$$

L'équation (2.7) suggère de choisir :

$$\beta_d = M^{-1}, \quad (2.8a)$$

$$\psi(t) = c(t) - \exp(AT_e)x(t). \quad (2.8b)$$

### 2.3.1.2. Architecture du CCM non optimisé

En observant que  $M^{-1}$  et  $\exp(AT_e)$  sont deux matrices constantes, on aboutit à l'architecture de CCM non optimisé présentée figure 2.2.

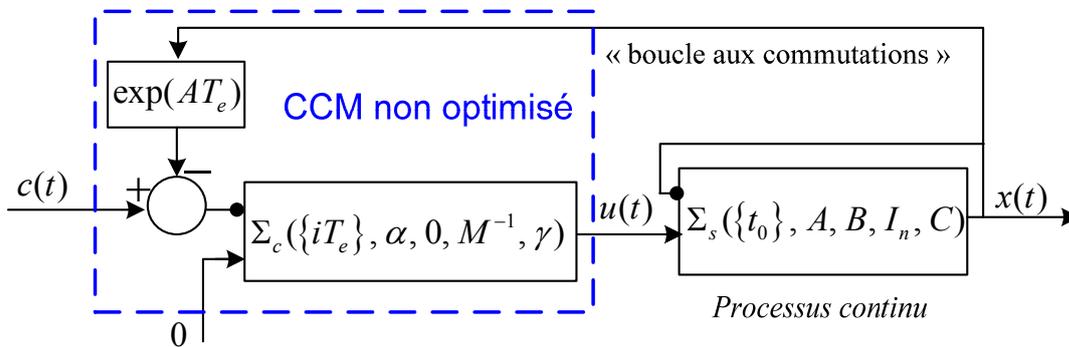


Fig. 2.2. Poursuite échantillonnée par un CCM non optimisé

Finalement, le CCM non optimisé est donc entièrement identifié par :

$$\Sigma_c(\{iT_e\}, \alpha, 0, M^{-1}, \gamma)$$

avec :

- $M$  définie par l'équation (2.6) à condition que  $M$  soit inversible. Quant aux matrices  $\alpha$  et  $\gamma$ , elles sont choisies arbitrairement dans ce mode de fonctionnement.
- Le bouclage discret est défini par (2.8),
- Le bouclage continu est absent de l'architecture.

Sans bouclage continu, le CCM permet de réaliser une poursuite échantillonnée garantissant  $x_{i+1} = c_i$  aux instants de commutation. Par contre, aucune contrainte n'étant

imposée sur la commande entre ces instants, il existe un risque d'oscillations entre ces instants de commutation, surtout si la période  $T_e$  est grande.

### 2.3.1.3. Conditions d'existence de $M(A, B, \gamma, \alpha, T_e)^{-1}$

D'après l'équation (2.6), on peut noter que  $M = \exp(AT_e)\tilde{M}$ , avec  $\tilde{M} = \int_0^{T_e} \exp(-A\tau)B\gamma \exp(\alpha\tau)d\tau$ .

Il est évident que les conditions d'existence de  $M^{-1}$  sont les mêmes que celles de  $\tilde{M}^{-1}$ .

Pour poser les conditions d'existence, nous notons :

$$\exp(-A\tau) = \sum_{j=0}^{n-1} p_j(-\tau)A^j, \quad (2.9a)$$

$$\exp(\alpha\tau) = \sum_{j=0}^{n-1} q_j(\tau)\alpha^j, \quad (2.9b)$$

où les coefficients réels  $p_j(-\tau)$  et  $q_j(\tau)$ , ( $j=0, \dots, n-1$ ), notés  $p_j$  et  $q_j$ , constituent les coordonnées de deux vecteurs  $P(-\tau)$  et  $Q(\tau)$ , notés plus simplement  $P$  et  $Q$ , tels que :

$$P^T = [p_0, p_1, \dots, p_{n-1}], \quad (2.10a)$$

$$Q^T = [q_0, q_1, \dots, q_{n-1}]. \quad (2.10b)$$

Dans ces conditions, nous pouvons écrire, d'après (2.9a) et (2.9b) :

$$\tilde{M} = \int_0^{T_e} \left[ \sum_{j=0}^{n-1} p_j A^j \right] B \gamma \left[ \sum_{j=0}^{n-1} q_j \alpha^j \right] d\tau \quad (2.11)$$

Ou encore :

$$\tilde{M} = \int_0^{T_e} \left[ \sum_{j=0}^{n-1} A^j B (p_j I_r) \right] \left[ \sum_{j=0}^{n-1} (q_j I_r) \gamma \alpha^j \right] d\tau,$$

$I_r$  étant la matrice identité d'ordre  $r$ . Soit, d'après (2.10) :

$$\tilde{M} = \int_0^{T_e} K(P \otimes I_r)(Q^T \otimes I_r)\Omega d\tau, \quad (2.12)$$

dans laquelle  $\otimes$  désigne le produit de Kronecker.  $K \in R^{n \times nr}$  correspond à la matrice de commandabilité de la paire  $(A, B)$  et  $\Omega \in R^{nr \times n}$  représente la matrice d'observabilité de la paire  $(\alpha, \gamma)$ .

L'expression (2.12) peut encore s'écrire :

$$\tilde{M} = \int_0^{T_e} K(PQ^T \otimes I_r)\Omega d\tau = K \left[ \int_0^{T_e} (PQ^T \otimes I_r) d\tau \right] \Omega, \quad (2.13)$$

En posant  $\Pi = \int_0^{T_e} P(-\tau)Q^T(\tau)$  et  $\hat{\Pi} = \Pi \otimes I_r$ , on obtient :

$$\tilde{M} = K\hat{\Pi}\Omega \quad (2.14)$$

**Théorème 2.1.** (Condition d'existence d'une commande CCM non optimisée)

Soient un processus linéaire MIMO continu  $\Sigma_c(\{t_0\}, A, B, I_n, C)$  et un CCM non optimisé défini par  $\Sigma_c(\{k.T\}, \alpha, 0, \beta_d, \gamma)$ . Si l'on note :

- $K = [B \mid AB \mid \dots \mid A^{n-1}B]$ , matrice de commandabilité de la paire  $(A, B)$ ,
- $\Omega = [\gamma^T \mid \alpha^T \cdot \gamma^T \mid \dots \mid (\alpha^T)^{n-1} \cdot \gamma^T]^T$ , matrice d'observabilité de la paire  $(\gamma, \alpha)$ ,
- $\Pi = \int_0^{T_e} P(-\tau)Q^T(\tau)$ , où  $P(t)$  et  $Q(t)$  sont respectivement les vecteurs des coordonnées de  $\exp(At)$  et  $\exp(\alpha t)$  sur les bases des  $A^i$  et des  $\alpha^i$  ( $i = 0, \dots, n-1$ ),
- $\hat{\Pi} = \Pi \otimes I_r$ .

alors, pour que la poursuite échantillonnée, avec un retard d'une période d'échantillonnage, soit réalisable, il faut et il suffit que :

- $\text{Ker}(\Omega) = \{0\}$
- $\text{Ker}(\hat{\Pi}) \cap \text{Im}(\Omega) = \{0\}$
- $\text{Ker}(K) \cap \text{Im}(\hat{\Pi}\Omega) = \{0\}$

La matrice  $\beta_d$  existe, est régulière et est définie par :

$$(\beta_d)^{-1} = M(A, B, \gamma, \alpha, T_e) = \exp(AT_e) \int_0^{T_e} \exp(-A\tau) B \gamma \exp(\alpha\tau)$$

**Démonstration :** Sachant que  $\tilde{M} = K\hat{\Pi}\Omega$  est une matrice carrée (de dimension  $n \times n$ ), son inversion est possible si et seulement si  $Ker(K\hat{\Pi}\Omega) = \{0\}$ .

Pour évaluer cette condition nécessaire et suffisante, nous écrivons :

$$\forall \eta \in Ker(K\hat{\Pi}\Omega) \Leftrightarrow \\ [\eta \in Ker(\Omega)] \text{ OU } [\Omega\eta \in Ker(\hat{\Pi})] \text{ OU } [\hat{\Pi}\Omega\eta \in Ker(K)]$$

Ainsi,  $Ker(K\hat{\Pi}\Omega) = \{0\}$  est équivalent à :

$$\forall \eta \neq 0 \Leftrightarrow \left\{ \begin{array}{l} [\eta \notin Ker(\Omega)] \Rightarrow Ker(\Omega) = \{0\} \\ ET [\Omega\eta \notin Ker(\hat{\Pi})] \Rightarrow Ker(\hat{\Pi}) \cap Im(\Omega) = \{0\} \\ ET [\hat{\Pi}\Omega\eta \notin Ker(K)] \Rightarrow Ker(K) \cap Im(\hat{\Pi}\Omega) = \{0\} \end{array} \right\}$$

La matrice  $M$  étant de même rang que  $\tilde{M}$ , ces conditions valent pour  $M$ .

#### 2.3.1.4. Validation expérimentale

Nous considérons ici, un processus linéaire continu instable défini par  $\Sigma_s(\{t_0\}, A, B_c, I_2, C)$  avec :

$$A = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad C = [1 \quad 0],$$

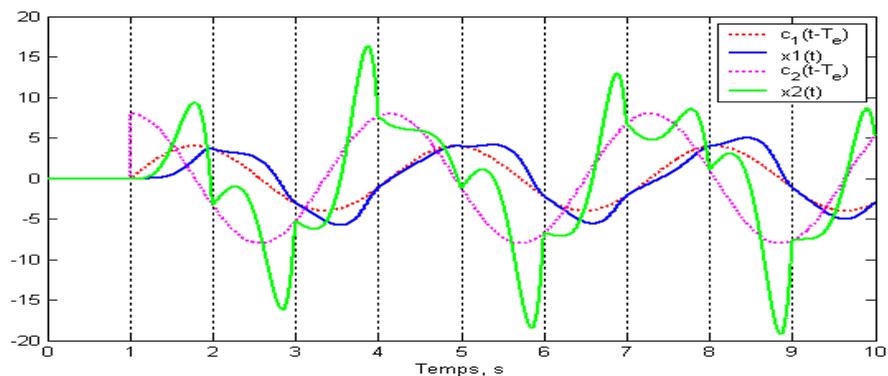
commandé par deux CCM non optimisés déterminés par  $\Sigma_c(\{iT_e\}, \alpha, 0, M^{-1}, \gamma)$ , où :

$$\text{C1 (CCM instable) : } \alpha = \begin{bmatrix} -2 & 8 \\ -7 & 10 \end{bmatrix}, \quad \gamma = [10 \quad 10] \text{ et } T_e = 1s$$

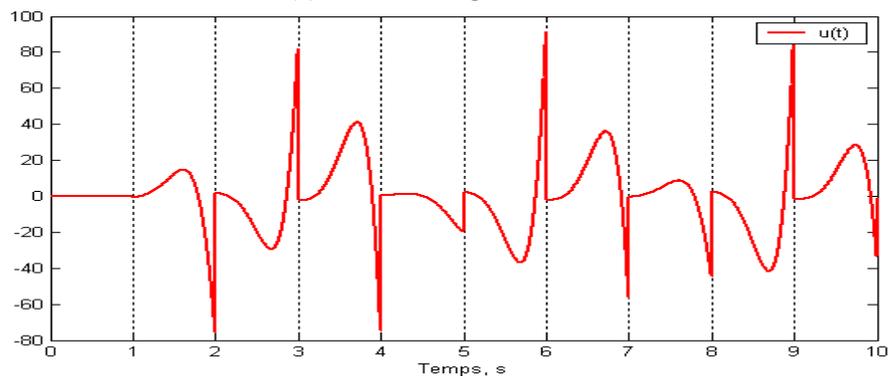
$$\text{C2 (CCM stable) : } \alpha = \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}, \quad \gamma = [10 \quad 10] \text{ et } T_e = 1s,$$

pour suivre une consigne d'état définie par :  $c(t) = \begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} = \begin{bmatrix} 4 \sin(2t) \\ 8 \cos(2t) \end{bmatrix}$ .

Les résultats sont présentés figures 2.3 et 2.4. Sachant que la poursuite est réalisée avec un retard d'une période de commutation, nous avons retardé les signaux de consigne d'une période, afin de faciliter la comparaison entre les courbes.

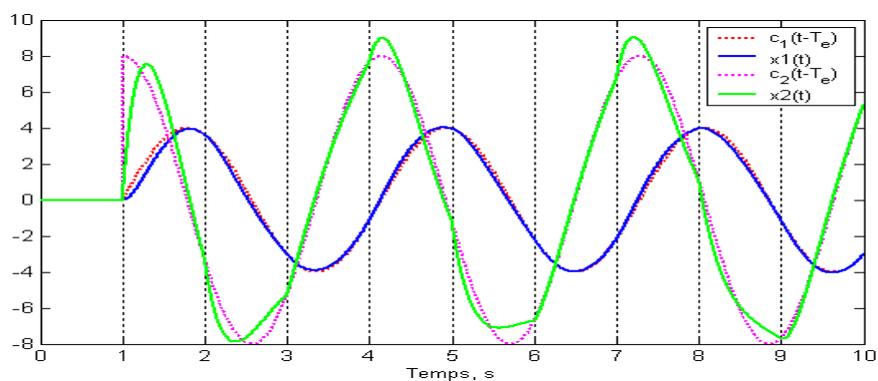


(a). Etat et consigne retardée

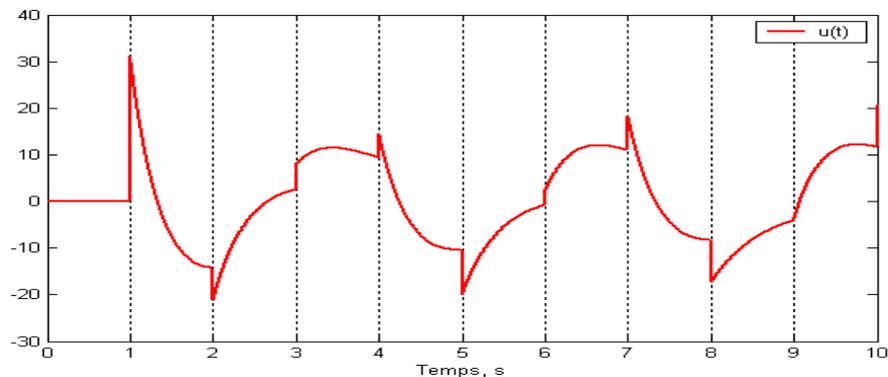


(b). Commande

**Fig. 2.3.** Poursuite échantillonnée non optimisée par C1 avec  $T_e = 1s$



(a). Etat et consigne retardée



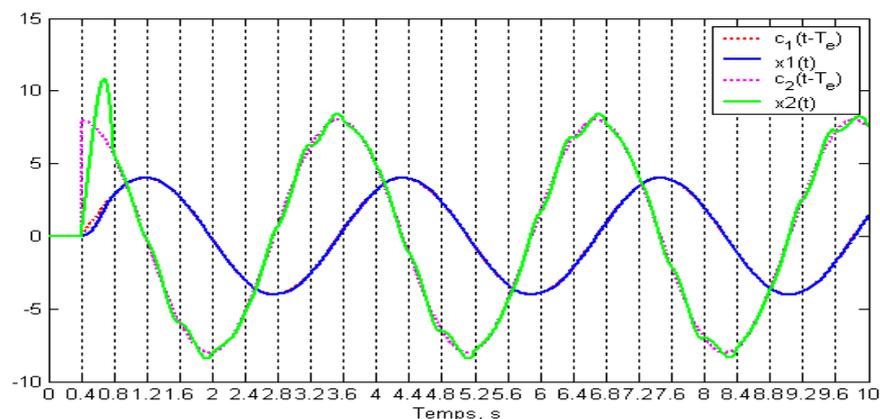
(b). Commande

**Fig. 2.4.** Poursuite échantillonnée non optimisée par C2 avec  $T_e = 1s$

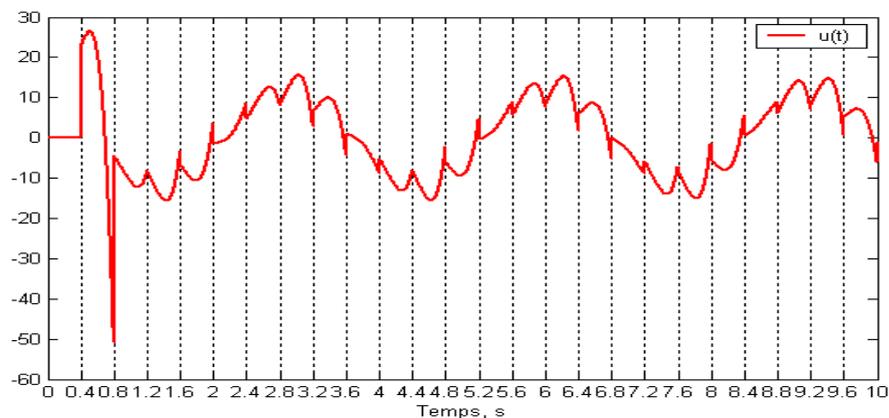
Les figures 2.3a et 2.4a font apparaître que les deux composantes d'état suivent parfaitement la consigne aux instants de commutation, ce qui réalise la condition de poursuite. L'évolution de l'état entre deux instants de commutations ne dépend que du choix de la matrice  $\alpha$ .

La comparaison entre les résultats des figures 2.3b et 2.4b fait apparaître que la stabilité de la matrice  $\alpha$  permet normalement d'éviter des valeurs excessives de la commande.

Il faut noter aussi que la période de commutation influe sur les performances du contrôleur. La valeur adoptée dans les deux exemples précédents a été choisie volontairement longue pour des raisons de démonstrations. Les performances de poursuite sont naturellement meilleure s'il on applique une commutation plus rapide. La comparaison peut se faire sur les figures 2.3 et 2.5, dans lesquelles la seule différence se situe sur la durée de la période de commutation (dans la figure 2.5, on utilise  $T_e = 0.4s$ ). On constate que, dans figure 2.5, avec le même contrôleur, les oscillations diminuent entre les instants de commutations et la commande devient moins forte aux instants de commutations.



(a). Etat et consigne retardée



(b). Commande

**Fig. 2.5.** Poursuite échantillonnée non optimisée par CI avec  $T_e = 0.4s$

Pour tester la robustesse de notre méthode, on applique C1 avec  $T_e = 0.4s$  et en supposant l'état bruité : bruit blanc de moyenne 0 et de variance  $0.01$ . Les résultats correspondant sont illustrés figure 2.6. On constate que la poursuite est réalisée sans trop déformer les performances par rapport à la figure 2.5.

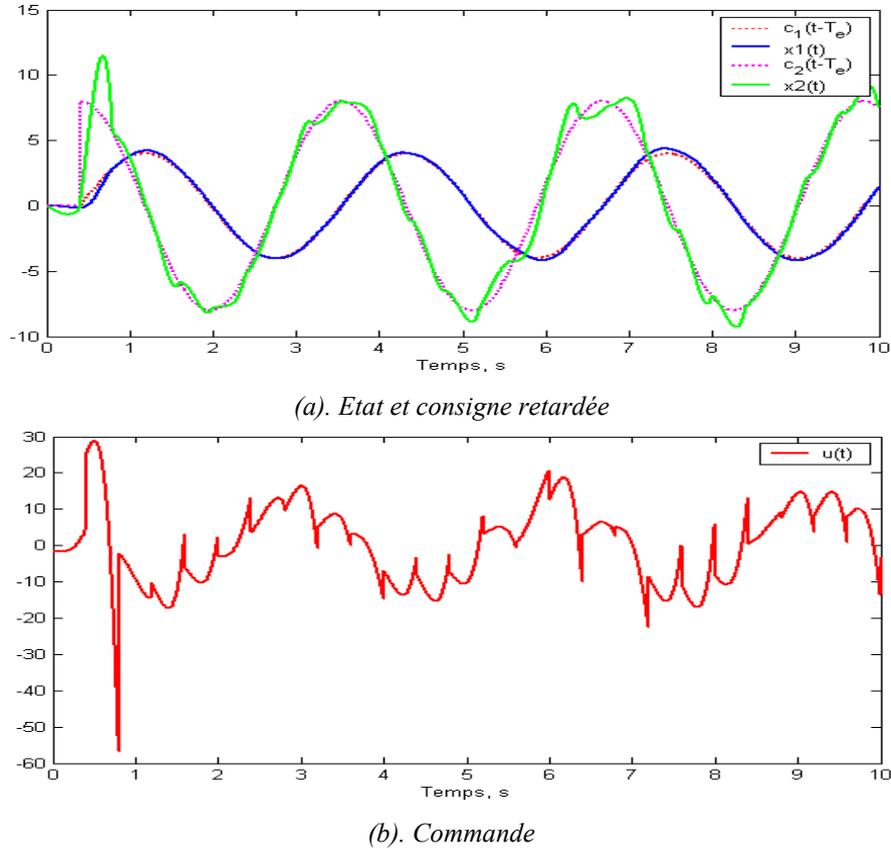


Fig. 2.6. Poursuite échantillonnée non optimisée par C1 à retour perturbé avec  $T_e = 0.4s$

### 2.3.2. CCM optimisé

Dans ce cas, nous considérons une optimisation du CCM présenté précédemment en utilisant, en plus de la boucle aux commutations (discrète) et de la condition de poursuite, une boucle inter-commutations (continue) permettant de réagir entre les instants de commutation. Les deux bouclages (discret et continu) opèrent en parallèle.

Le CCM optimisé est défini par  $\Sigma_c(\{iT_e\}, \alpha, \beta_c, \beta_d, \gamma)$ . L'idée est de minimiser les oscillations entre les instants de commutation en définissant convenablement les paramètres et les entrées du SCLM. Le mode de fonctionnement souhaité est défini par :

- $x_{i+1} = c_i, \forall i = 0, 1, 2, \dots,$
- minimiser la distance entre  $c(t - T_e)$  et  $x(t)$  sur le morceau  $q_i = ]t_i, t_{i+1}]$ ,

La poursuite échantillonnée est garantie par la boucle discrète aux commutations comme précédemment et l'optimisation se fait par la boucle continue inter-commutations.

### 2.3.2.1. Mise en équation

La fonctionnelle à minimiser sur chaque morceau est de la forme :

$$J = \frac{1}{2} \int_{iT_e}^{(k+1)T_e} \left[ (c(\tau - T_e) - x(\tau))^T E (c(\tau - T_e) - x(\tau)) + w^T(\tau) G w(\tau) \right] d\tau \quad (2.15)$$

dans laquelle  $E \in \mathfrak{R}^{n \times n}$  et  $G \in \mathfrak{R}^{r \times r}$  sont deux matrices symétriques définies positives.

La minimisation de  $J$  conduit à minimiser la distance entre  $c(t - T_e)$  et  $x(t)$ , tout en modérant la commande  $u(t) = w(t)$ .

En reprenant l'équation (2.2a) du processus avec pour entrée la sortie du CCM, la solution à ce problème de commande optimale est obtenue en appliquant le principe de Pontryagin à l'hamiltonien  $H$  défini par :

$$H = -\frac{1}{2} \left[ (c(t - T_e) - x(t))^T E (c(t - T_e) - x(t)) + u^T(t) G u(t) \right] + \lambda^T(t) [Ax(t) + Bu(t)] \quad (2.16)$$

où  $\lambda(t) \in R^n$  correspond au vecteur adjoint (multiplicateur de Lagrange).

Le principe de Pontryagin donne la solution suivante :

$$\frac{dH}{d\lambda} = x'(t) = Ax(t) + Bu(t), \quad (2.17a)$$

$$-\frac{dH}{dx} = \lambda'(t) = -A^T \lambda(t) - E [c(t - T_e) - x(t)], \quad (2.17b)$$

$$\frac{dH}{du} = 0 = -Gu(t) + B^T \lambda(t), \text{ soit : } u(t) = G^{-1} B^T \lambda(t), \quad (2.17c)$$

### 2.3.2.2. Identification des paramètres du CCM optimisé

La solution de Pontryagin donnant les équations (2.17) peut être interprétée de la manière suivante :

- l'équation (2.17a) est l'équation d'état du système à commander.
- Les équations (2.17b) et (2.17c) suggèrent de définir le CCM optimisé  $\Sigma_c(S, \alpha, \beta_c, \beta_d, \gamma)$  comme suit :

$$S = \{iT_e, i = 0, 1, 2, \dots\}, \quad (2.18a)$$

$$\alpha = -A^T, \quad (2.18b)$$

$$\beta_c = -E, \quad (2.18c)$$

$$\gamma = G^{-1}B^T, \quad (2.18d)$$

$$\varphi(t) = c(t - T_e) - x(t). \quad (2.18e)$$

L'intégration du système d'équations (2.17) sur le morceau  $q_i$  permet de compléter la définition du CCM et d'en déduire  $\beta_d$ . De plus, elle permet de préciser la fonctionnelle  $\Psi(c(t), x(t))$  caractérisant la boucle discrète.

En utilisant les systèmes d'équations (2.17) et (2.18), on peut écrire :

$$\begin{bmatrix} x'(t) \\ \lambda'(t) \end{bmatrix} = H \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} + Kc(t - T_e) \quad (2.19a)$$

$$\text{Avec : } H = \begin{bmatrix} A & BG^{-1}B^T \\ E & -A^T \end{bmatrix} \in \mathfrak{R}^{2n \times 2n} \text{ et } K = \begin{bmatrix} 0 \\ -E \end{bmatrix} \in \mathfrak{R}^{2n \times n} \quad (2.19b)$$

En intégrant (2.19a) sur le morceau  $q_i$ , on obtient :

$$\begin{bmatrix} x_{i+1} \\ \lambda_{i+1} \end{bmatrix} = \exp(HT_e) \begin{bmatrix} x_i \\ \lambda_i^+ \end{bmatrix} + I_{i+1} \quad (2.20a)$$

$$\text{Avec : } I_{i+1} = \int_{iT_e}^{(i+1)T_e} \{ \exp(H[(i+1)T_e - \tau])Kc(\tau - T_e) \} d\tau \quad | \quad I_{i+1} \in R^{2n \times 1} \quad (2.20b)$$

Si l'on note :

$$\exp(HT_e) = \Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} \text{ et } I_{i+1} = \begin{bmatrix} I_x \\ I_\lambda \end{bmatrix}, \quad (2.21)$$

$I_x \in R^{n \times 1}$  et  $I_\lambda \in R^{n \times 1}$  étant les moitiés supérieure et inférieure de  $I_{i+1}$ , on obtient à partir de la première relation de (2.20a) :

$$x_{i+1} = \Theta_{11}x_i + \Theta_{12}\lambda_i^+ + I_x \quad (2.22)$$

Rappelons que la valeur  $\lambda_i^+$  de l'état imposé du contrôleur lors de la commutation est donnée par  $\lambda_i^+ = \beta_d \psi_i$  selon (2.2b). Il faut donc identifier la matrice  $\beta_d$  et l'entrée  $\psi_k$ . Dans ce sens, en remplaçant la condition de poursuite (2.1) dans (2.22) :

$$c_i = \Theta_{11}x_i + \Theta_{12}\beta_d\psi_i + I_x \quad (2.23)$$

et en supposant que  $\Theta_{12}$  soit inversible, on obtient, à partir de la relation  $\lambda_i^+ = \beta_d\psi_i$  :

$$\beta_d\psi_i = \Theta_{12}^{-1}(c_i - \Theta_{11}x_i - I_x) \quad (2.24)$$

L'équation (2.24) suggère donc d'adopter :

$$\beta_d = \Theta_{12}^{-1} \quad (2.25a)$$

$$\psi(t) = c(t) - \Theta_{11}x(t) - I_x \quad (2.25b)$$

La structure du CCM optimisé est ainsi entièrement définie par les relations (2.18a-d) et (2.25a). Le bouclage continu inter-commutations est défini par la relation (2.18e) et le bouclage discret aux commutations est défini par (2.25b).

### 2.3.2.3. Architecture du CCM optimisé

#### 2.3.2.3.1. Calculs préliminaires

L'équation (2.20b) fait intervenir le terme  $I_x$ . Nous définissons donc, au préalable, un mode de calcul de ce terme permettant une réalisation simple de la fonctionnelle de bouclage  $\Psi(c(t), x(t))$ .

En posant  $\tau = \theta + T_e$  dans l'équation (2.20b), il vient :

$$I_{i+1} = \int_{(i-1)T_e}^{iT_e} \{ \exp(H[iT_e - \theta])Kc(\theta) \} d\theta \quad (2.26)$$

On peut encore écrire :

$$I_{i+1} = \lim_{t \rightarrow iT_e, t < iT_e} [\tilde{I}(t)] = \tilde{I}_i^- \quad (2.27)$$

avec :

$$\tilde{I}(t) = \int_{(i-1)T_e}^t \{ \exp(H[t - \theta])Kc(\theta) \} d\theta \quad (2.28)$$

L'équation (2.20b) fait apparaître une intégrale de convolution dont la moitié supérieure correspond, à l'instant,  $iT_e^-$  à  $I_x$ . Ce dernier peut donc être interprété comme la sortie, à  $iT_e^-$ , d'un SLCM auxiliaire défini par  $\Sigma_c(\{iT_e^-\}, H, K, 0, [I_n \ 0])$ , dont seule l'entrée continue est alimentée par  $c(t)$ .

## 2.3.2.3.2. Structure du contrôleur optimisé

Nous donnons en 2.3.2.4 les conditions d'existence de la poursuite échantillonnée optimisée garantie par le CCM optimisé défini par  $\Sigma_c(\{iT_e\}, -A^T, -E, \Theta_{12}^{-1}, G^{-1}B^T)$ .

En considérant les fonctionnelles définissant les entrées de ce CCM telles que définies précédemment, nous proposons un schéma de construction optimisé figure 2.7 :

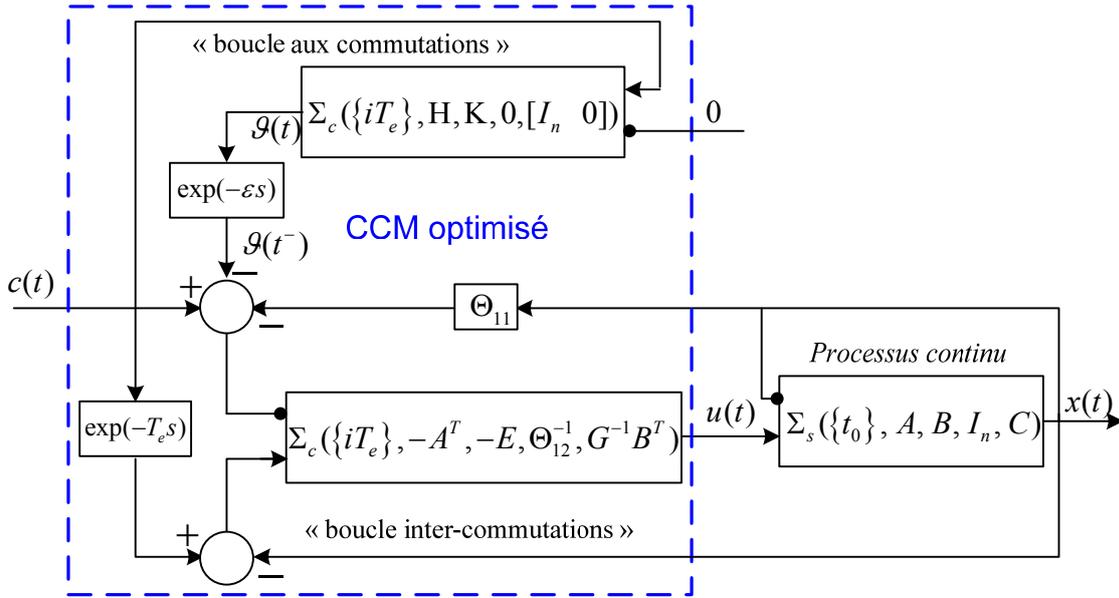


Fig. 2.7. Architecture de CCM optimisé

Dans ce schéma :

- $\exp(-\varepsilon s)$  représente un retard infiniment petit permettant de prendre en compte la sortie du SLCM auxiliaire  $\Sigma_c(\{iT_e\}, H, K, 0, [I_n \ 0])$  juste avant l'instant de commutation (instant marqué par une discontinuité),
- $\exp(-T_e s)$  représente un retard d'une période d'échantillonnage,
- l'entrée à discrétiser de  $\Sigma_c(\{iT_e\}, -A^T, -E, \Theta_{12}^{-1}, G^{-1}B^T)$  est :  $\psi(t) = c(t) - \Theta_{11} \cdot x(t) - \mathcal{G}(t^-)$  où  $\mathcal{G}(t)$  représente la sortie du SLCM auxiliaire  $\Sigma_c(\{iT_e\}, H, K, 0, [I_n \ 0])$ .

 2.3.2.4. Condition d'existence de  $\Theta_{12}^{-1}$ 

 2.3.2.4.1. Bloc-triangularisation de  $H$ 

Selon (2.21),  $\Theta_{12}$  est un bloc de  $\exp(HT_e)$ . De plus, les matrices  $G^{-1}$  et  $E$  apparaissant dans  $H$  (2.19b) sont symétriques et définies positives.

On peut donc les factoriser selon la méthode de Cholesky [Rot 95] et écrire :

$$G^{-1} = UU^T \text{ et } E = V^T V, \quad (2.29)$$

expression dans laquelle  $U \in R^{r \times r}$  et  $V^T \in R^{n \times n}$  sont des matrices triangulaires inférieures régulières, dont les éléments diagonaux sont tous strictement positifs.

Dans ces conditions, en posant  $BU = \tilde{B}$ , on peut écrire H sous la forme suivante :

$$H = \begin{bmatrix} A & \tilde{B}\tilde{B}^T \\ V^T V & -A^T \end{bmatrix} \quad (2.30)$$

dans laquelle H est une matrice hamiltonienne. Elle est donc bloc-triangularisable selon la transformation régulière :

$$\tilde{H} = \begin{bmatrix} I_n & 0 \\ X & I_n \end{bmatrix} \begin{bmatrix} A & \tilde{B}\tilde{B}^T \\ V^T V & -A^T \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -X & I_n \end{bmatrix}, \quad (2.31)$$

expression dans laquelle  $X \in R^{n \times n}$  est une solution de l'équation de Riccati symétrique :

$$-X\tilde{B}\tilde{B}^T X + XA + A^T X + V^T V = 0. \quad (2.32)$$

On obtient alors :

$$\tilde{H} = \begin{bmatrix} L & \tilde{B}\tilde{B}^T \\ 0 & -L^T \end{bmatrix}, \text{ avec } L = a - \tilde{B}\tilde{B}^T X \quad (2.33)$$

#### 2.3.2.4.2. Interprétation

La transformation (2.31) appliquée à l'équation (2.19) conduit à la nouvelle équation :

$$\begin{bmatrix} x'(t) \\ \tilde{\lambda}'(t) \end{bmatrix} = \begin{bmatrix} L & \tilde{B}\tilde{B}^T \\ 0 & -L^T \end{bmatrix} \begin{bmatrix} x(t) \\ \tilde{\lambda}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ E \end{bmatrix} c(t - T_e) \quad (2.34a)$$

$$\text{avec } \tilde{\lambda}(t) = \lambda(t) + Xx(t). \quad (2.34b)$$

Cette équation conduit à un système de la même structure que le système (2.2) dans lequel on ferait :

$$A = L, \quad (2.35a)$$

$$\alpha = -L^T, \quad (2.35b)$$

$$B = \tilde{B}, \quad (2.35c)$$

$$\gamma = \tilde{B}^T. \quad (2.35d)$$

On a donc, en intégrant (2.34a), sur le morceau  $q_i$  :

$$x_{i+1} = \exp(LT_e)x_i + \tilde{\Theta}_{12}\tilde{\lambda}_i^+ - \hat{I}_{i+1} \quad (2.36a)$$

$$\text{avec : } \tilde{\Theta}_{12} = \int_0^{T_e} e^{\exp(L[T_e - \tau])} \tilde{B}\tilde{B}^T \exp(-L^T \tau) d\tau \quad (2.36b)$$

$$\text{et } \hat{I}_{i+1} = \int_{iT_e}^{(i+1)T_e} \exp(L[(i+1)T_e - \tau]) E c(\tau - T_e) d\tau \quad (2.36c)$$

Dès lors, on peut énoncer les propositions suivantes :

- $\Theta_{12}$  inversible  $\Leftrightarrow \lambda_i^+$  calculable, d'après (2.22),
- $\lambda_i^+$  calculable  $\Leftrightarrow \tilde{\lambda}_i^+$  calculable, d'après (2.34b),
- $\tilde{\lambda}_i^+$  calculable  $\Leftrightarrow \tilde{\Theta}_{12}$  inversible d'après (2.36a).

Donc,  $\Theta_{12}$  inversible  $\Leftrightarrow \tilde{\Theta}_{12}$  inversible.

#### 2.3.2.4.3. Calcul de $\tilde{\Theta}_{12}$

D'après (2.36b), on peut écrire :

$$\tilde{\Theta}_{12} = e^{\exp(LT_e)} \int_0^{T_e} e^{\exp(-L\tau)} \tilde{B}\tilde{B}^T \exp(-L^T \tau) d\tau$$

Par un raisonnement analogue à celui utilisé pour calculer  $M$ , dans la section 2.3.1.3, on obtient :

$$\tilde{\Theta}_{12} = e^{\exp(LT_e)} K \hat{\Pi} \Omega \quad (2.37a)$$

$$\text{avec } K = [\tilde{B}, L\tilde{B}, \dots, L^{n-1}\tilde{B}], \quad (2.37b)$$

$$\Omega = [\tilde{B}, L\tilde{B}, \dots, L^{n-1}\tilde{B}]^T = K^T, \quad (2.37c)$$

$$\Pi = \int_0^{T_c} P(-\tau)P^T(-\tau)d\tau, \quad (2.37d)$$

où  $P(t)$  est le vecteur des coordonnées de  $\exp(Lt)$  sur les  $L^i$  ( $i = 0, \dots, n-1$ ),  $\hat{\Pi} = \Pi \otimes I_r$ .

Les conditions d'existence de  $\tilde{\Theta}_{12}^{-1}$  s'en déduisent immédiatement, d'après le théorème 2.1 :

- $Ker(\Omega) = \{0\}$ ,
- $Ker(\hat{\Pi}) \cap Im(\Omega) = \{0\}$ ,
- $Ker(K) \cap Im(\hat{\Pi}\Omega) = \{0\}$ ,

expressions dans lesquelles on a :  $\Omega = K^T$ .

**Théorème 2.2.** (Condition d'existence d'une commande par le CCM optimisé)

Soient un processus linéaire MIMO continu  $\Sigma_c(\{t_0\}, A, B, I_n, C)$  et un CCM optimisé basé sur  $\Sigma_c(\{iT_e\}, -A^T, -E, \beta_d, G^{-1}B^T)$ , avec  $G^{-1} = U.U^T$  et  $E = V^T.V$ , matrices définies positives. Si l'on note :

- $\tilde{B} = BU$ ,
- $L = A - \tilde{B}\tilde{B}^T X$ ,  $X$  étant solution de l'équation de Riccati symétrique :  
 $-X\tilde{B}\tilde{B}^T X + XA + A^T X + V^T V = 0$ ,
- $K = [\tilde{B}, L\tilde{B}, \dots, L^{n-1}\tilde{B}]$ ,
- $\Pi = \int_0^{T_c} P(-\tau)P^T(-\tau)d\tau$ , où  $P(t)$  est le vecteur des coordonnées de  $\exp(Lt)$  sur les  $L^i$  ( $i = 0, \dots, n-1$ ),
- $\hat{\Pi} = \Pi \otimes I_r$ ,
- $H = \begin{bmatrix} a & \tilde{b}\tilde{b}^T \\ V^T.V & -a^T \end{bmatrix}$ .

alors, pour que la poursuite échantillonnée sans oscillations, avec un retard d'une période d'échantillonnage, soit réalisable, il faut et il suffit que :

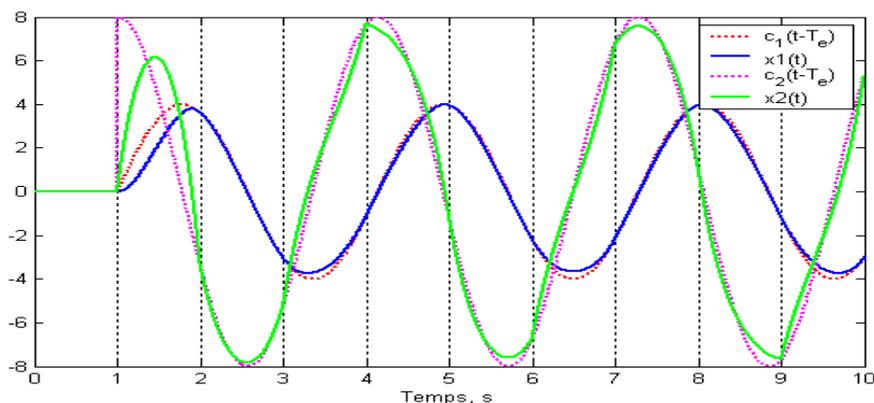
- $Ker(K^T) = \{0\}$ ,
- $Ker(\hat{\Pi}) \cap Im(K^T) = \{0\}$ ,
- $Ker(K) \cap Im(\hat{\Pi}K^T) = \{0\}$ ,

La matrice  $\beta_d$  est alors régulière et est défini par :

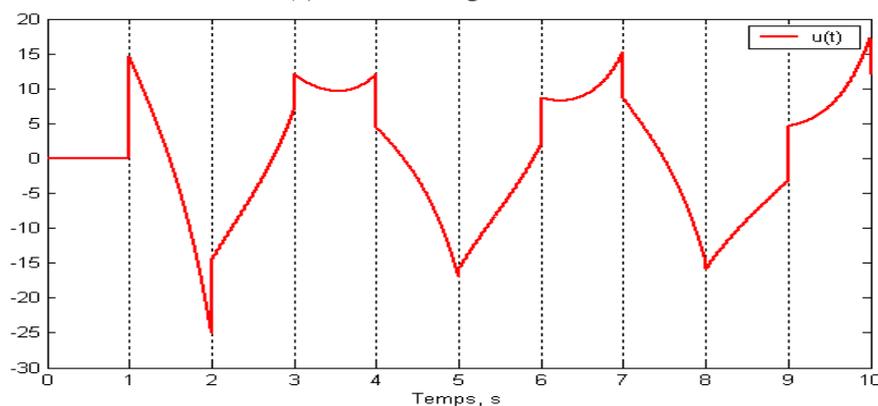
$$(\beta_d)^{-1} = \Theta_{12} = \text{bloc supérieur droit } (n \times n) \text{ de } e^{HT_e}.$$

### 2.3.2.5. Validations expérimentales

Pour faciliter la comparaison avec le CCM non optimisé, nous considérons le même processus que dans le §2.3.1.4, mais commandé par un CCM optimisé défini par  $\Sigma_c(\{iT_e\}, -A^T, -E, \Theta_{12}^{-1}, G^{-1}B^T)$ , avec  $E = 0.01I_2$ ,  $G = 100$  et  $T_e = 1s$  pour suivre la même consigne d'état.



(a). Etat et consigne retardée



(b). Commande

**Fig. 2.8.** Poursuite échantillonnée optimisée avec  $T_e = 1s$

Les résultats illustrés figure 2.8 montrent que les fortes oscillations sont atténuées. De plus, la commande optimisée est à la fois moins importante en amplitude et plus douce.

Par contre, pour les périodes de commutation plus petites (par exemple  $T_e = 0.1s$ ), la différence entre les deux modes de contrôle (CCM non optimisé et CCM optimisé) n'est évidente ni sur la réponse ni sur la commande. Le processus suit parfaitement la consigne avec une période de retard. Dans ces conditions, la commande optimisée n'apparaît plus comme nécessaire si la période de commutation est très faible.

En poussant ce raisonnement, nous proposons, dans la section 2.5, un contrôleur spécial appelé *Contrôleur Récursif* obtenu à partir d'un CCM non optimisé standard, pour lequel la période de commutation tend vers zéro.

## 2.4. Contrôleur bi-échantillonné

Dans le cas où l'état du système associé évolue de manière *échantillonnée* entre les instants de commutation, nous définissons les Contrôleurs Bi-Echantillonnés (CBE). Dans ce cas, la commande générée a une forme *échantillonnée*. Chaque morceau est défini par :  $q_i = \{t_i^k = iT_e + kt_e, k = 0, 1, 2, \dots, p-1\}$ , avec  $T_e = pt_e$  ( $p$  entier positif,  $T_e$  période de commutation,  $t_e$  période d'échantillonnage à l'intérieur du morceau).

Dans ce contexte, le CBE alimente le processus continu  $\Sigma_s(\{t_0\}, A, B, I_n, C)$  par une commande échantillonnée bloquée nécessitant un retour d'état discrétisé.

Le processus à commander, est décrit par :

$$x_i^{k+1} = fx_i^k + hu_i^k, \quad k = 0, 1, 2, \dots, p-1 \quad (2.38a)$$

$$x_i^0 = x_{i-1}^p \quad (2.38b)$$

$$y_i^k = Cx_i^k \quad (2.38c)$$

avec :  $f = \exp(At_e)$  et  $h = f \int_0^{t_e} \exp(-A\tau) B d\tau$ , (2.38d)

Nous rappelons la condition de poursuite échantillonnée définie par (2.1) :

$$x((i+1)T_e) = c(iT_e), \quad \forall i = 0, 1, 2, \dots$$

Selon la notation à deux échelles temps du formalisme des SLBE, cette condition peut s'écrire sous l'autre forme suivant :

$$x_{i+1}^0 = c_i^0, \quad \forall i = 0, 1, 2, \dots \quad (2.39a)$$

ou encore :

$$x_i^p = c_i^0 \quad (2.39b)$$

## 2.4.1. CBE non optimisé

### 2.4.1.1. Mise en équation

Dans ce cas, nous choisissons pour le CBE non optimisé, un SLBE défini par :  $\Sigma_d(S, \alpha, 0, \beta_d, \gamma)$  avec :

- $\dim(\lambda(t)) = \dim(x(t)) = n$ ,
- $\dim(w(t)) = \dim(u(t)) = r$ ,
- $\dim(\alpha) = n \times n$ ,
- $\dim(\gamma) = r \times n$  avec  $\gamma$  de rang plein,
- une entrée à discrétiser  $\psi(t)$  provenant d'une fonctionnelle  $\Psi$  (section 2.2),
- aucune entrée continue ( $\varphi(t) = 0 \quad \forall t$ ).

De la même manière que pour (2.38), sur le morceaux  $q_i$ , le contrôleur est décrit comme :

$$\lambda_i^0 = \beta_d \psi_i^0 \quad (2.40a)$$

$$\lambda_i^{k+1} = \alpha \lambda_i^k, \quad k = 0, 1, 2, \dots, p-1 \quad (2.40b)$$

$$u_i^k = w_i^k = \gamma \lambda_i^k \quad (2.40c)$$

En combinant les équations (2.38) et (2.40) sur le morceau  $q_i$ , on obtient :

$$x_i^p = f^p x_i^0 + \sum_{l=1}^p f^{p-l} h \gamma \alpha^{l-1} \lambda_i^0 \quad (2.41)$$

qui peut s'écrire sous forme matricielle :

$$x_i^p = f^p x_i^0 + M \lambda_i^0 \quad (2.42)$$

$$\text{avec : } M = \begin{bmatrix} f^{p-1} h, & f^{p-2} h, & \dots & f^0 h \end{bmatrix} \begin{bmatrix} \gamma \alpha^0 \\ \gamma \alpha^1 \\ \cdot \\ \cdot \\ \gamma \alpha^{p-1} \end{bmatrix} \quad (2.43)$$

Dans ces conditions, la condition de poursuite (2.39) impose que  $M$  soit inversible :

$$\lambda_i^0 = \beta_d v_i^0 = M^{-1} [c_i^0 - f^p x_i^0] \quad (2.44)$$

Ainsi, si  $M^{-1}$  existe, alors on peut définir le CBE non optimisé avec

$$\beta_d = M^{-1}, \quad (2.45a)$$

$$\psi(t) = c(t) - f^p x(t). \quad (2.45b)$$

#### 2.4.1.2. Condition d'existence de $M^{-1}$

Selon (2.43), on peut écrire  $M = K\Omega$  avec :

- $K = [f^{p-1}h, f^{p-2}h, \dots, f^0h]$ ,
- $\Omega^T = [\gamma\alpha^0 \quad \gamma\alpha^1 \quad \cdot \quad \cdot \quad \gamma\alpha^{p-1}]$ .

$M \in R^{n \times n}$  est inversible si et seulement si :

$$\begin{aligned} \text{Ker}(K\Omega) &= \{0\} \\ \equiv \text{Ker}(\Omega) &= \{0\} \text{ et } \text{Ker}(K) \cap \text{Im}(\Omega) = \{0\} \end{aligned}$$

**Remarque 2.2.** Si  $p = n$ , c'est à dire si le nombre de périodes d'échantillonnage  $t_e$  entre deux commutations est égal à l'ordre  $n$  du système, alors  $K$  et  $\Omega^T$  sont respectivement les matrices de commandabilité du système à commander et d'observabilité du contrôleur.

En général, la matrice  $M^{-1}$  n'existe pas si  $p < n$ .

#### 2.4.1.3. Architecture du CBE non optimisé

D'après la mise en équations, la construction d'un CBE non optimisé est facile. La figure 2.9 donne le schéma de réalisation à partir d'un SLBE défini par :

$$\Sigma_d(\{iT_e\}, \alpha, 0, M^{-1}, \gamma).$$

Le CBE non optimisé garantit, comme le CCM non optimisé, la poursuite échantillonnée d'une consigne d'état par le processus avec un retard d'une période de commutation  $T_e$ .

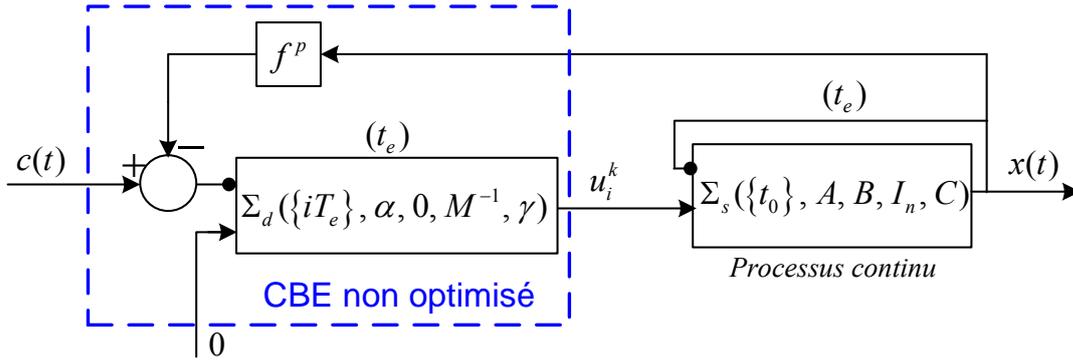


Fig. 2.9. Poursuite échantillonnée par un CBE non optimisé

#### 2.4.1.4. Validation expérimentale

Nous considérons le même processus qu'au §2.3.1.4, avec :

$$A = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad C = [1 \quad 0],$$

La consigne d'état imposée est définie comme :

$$c(t) = \begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} = \begin{bmatrix} 4 \sin(2t) \\ 8 \cos(2t) \end{bmatrix}.$$

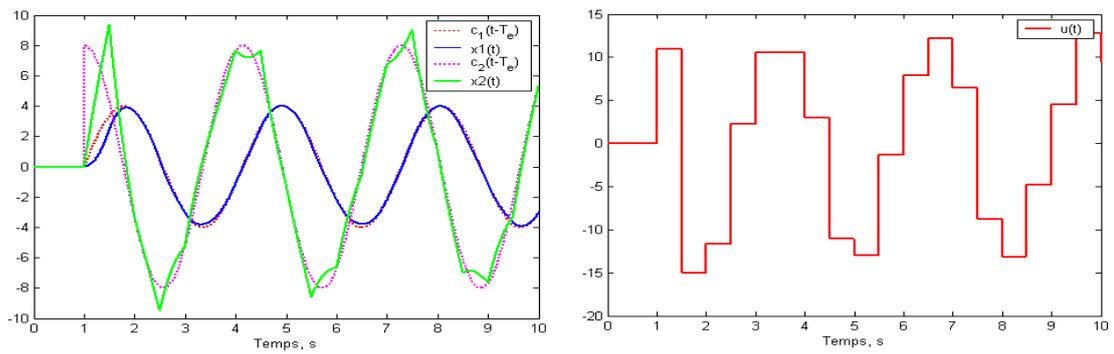
Nous réalisons ici la poursuite échantillonnée par un CBE défini par :

$$\Sigma_d(\{iT_e\}, \alpha, 0, M^{-1}, \gamma),$$

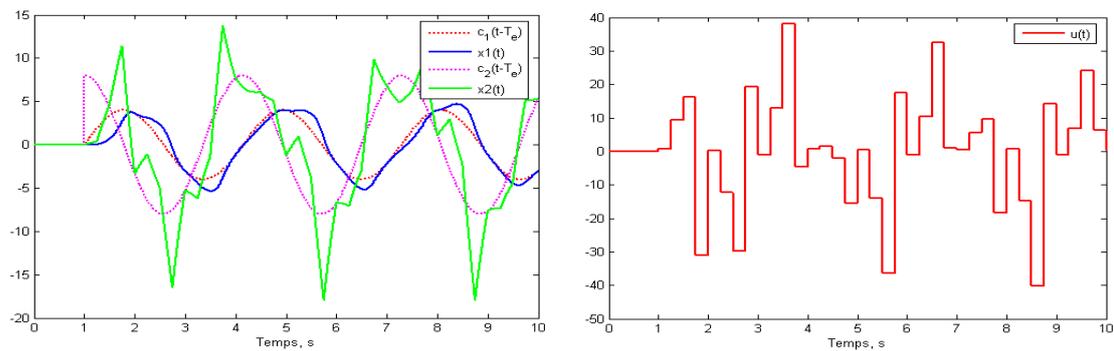
dans lequel :

$$T_e = 1s \text{ avec } T_e = p t_e, \quad \alpha = \exp\left(\begin{bmatrix} -2 & 8 \\ -7 & 10 \end{bmatrix} t_e\right), \quad \gamma = [10 \quad 10] \text{ et } M \text{ défini par (2.43).}$$

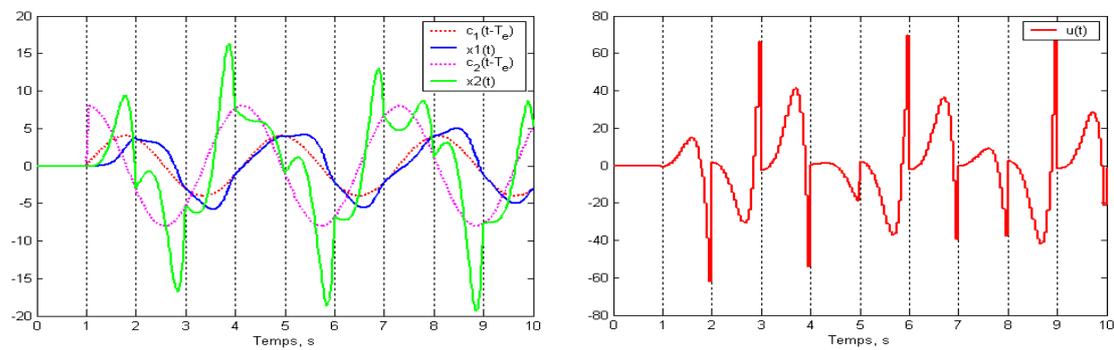
Les résultats, avec différentes valeurs de  $p$  (2, 4, 20) illustrés figure 2.10, sont comparés à ceux du CCM non optimisé équivalent. La poursuite est garantie à chaque instant de commutation mais des oscillations importantes existent à l'intérieur de chaque morceau. On remarque que la commande CBE devient proche de celle du CCM quand  $p$  augmente.



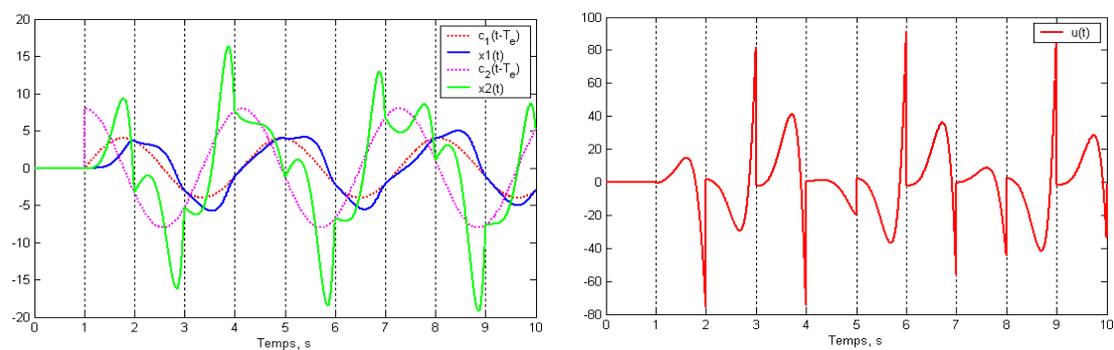
(a). CBE non optimisé ( $p=2$ )



(b). CBE non optimisé ( $p=4$ )



(c). CBE non optimisé ( $p=20$ )



(d). CCM non optimisé

**Fig. 2.10.** Poursuite échantillonnée par CBE non optimisé avec  $T_e=1s$

### 2.4.2. CBE optimisé

Dans ce paragraphe, nous traitons une optimisation du CBE présenté précédemment en utilisant, en plus de la boucle discrète et de la condition de poursuite, un bouclage entre les instants de commutation. Le CBE optimisé est défini par  $\Sigma_d(\{iT_e\}, \alpha, \beta_c, \beta_d, \gamma)$ . L'idée est de minimiser les oscillations dans chaque morceau en paramétrant convenablement le contrôleur.

Sur le morceau  $q_i$ , le mode de fonctionnement souhaité pour le CBE optimisé est défini par :

- $x_i^p = c_i^0, \forall i = 0, 1, 2, \dots,$
- minimiser la distance entre  $c(t - T_e)$  et  $x(t)$ .

#### 2.4.2.1. Mise en équation

Le critère de coût exprimant la contrainte supplémentaire sur le morceau est le suivant :

$$J = \frac{1}{2} * \sum_{k=0}^{p-1} \left[ (c_{i-1}^k - x_i^k)^T E (c_{i-1}^k - x_i^k) + u_i^{kT} G u_i^k \right] \quad (2.46)$$

où  $E \in R^{n \times n}$  et  $G \in R^{r \times r}$  sont deux matrices symétriques définies positives. La minimisation du critère  $J$  vise à réduire l'erreur entre l'état et la consigne d'état retardée de  $T_e$ , tout en modérant la commande. Selon la théorie de la commande optimale, l'hamiltonien correspondant s'écrit [Por 90] :

$$H = -\frac{1}{2} * \sum_{k=0}^{p-1} \left[ (c_{i-1}^k - x_i^k)^T E (c_{i-1}^k - x_i^k) + u_i^{kT} G u_i^k \right] + \sum_{k=0}^{p-1} (\lambda_i^{k+1})^T [f x_i^k + h u_i^k] \quad (2.47)$$

$\lambda_i^k$  étant le vecteur multiplicateur de Lagrange de dimension  $n$ .

Le principe de Pontryagin conduit à :

$$\lambda_i^k = \partial H / \partial x_i^k = E (c_{i-1}^k - x_i^k) + f^T \lambda_i^{k+1}$$

soit :

$$\lambda_i^{k+1} = (f^T)^{-1} \lambda_i^k - (f^T)^{-1} E (c_{i-1}^k - x_i^k) \quad (2.48a)$$

et 
$$u_i^k = G^{-1} h^T \lambda_i^{k+1} \quad (2.48b)$$

Les équations (2.48a) et (2.48b) peuvent être interprétées respectivement comme l'équation d'état et l'équation de sortie d'un système bi-échantillonné sur le morceau  $q_i$ . L'entrée de ce système est  $(c_{i-1}^k - x_i^k)$ , ce qui exprime un bouclage à retour unitaire.

Il reste à déterminer la valeur de  $\lambda_i^0$  permettant de satisfaire la condition de poursuite  $x_i^p = c_i^0$ .

#### 2.4.2.2. Calcul de la condition initiale du CBE optimisé $\lambda_i^0$

Pour calculer la valeur initiale de l'état du CBE à chaque commutation, on considère le système augmenté suivant, obtenu en combinant (2.38a) et (2.48) :

$$\begin{bmatrix} x_i^{k+1} \\ \lambda_i^{k+1} \end{bmatrix} = H \begin{bmatrix} x_i^k \\ \lambda_i^k \end{bmatrix} + K c_{i-1}^k \quad (2.49a)$$

avec :

$$H = \begin{bmatrix} f + hG^{-1}h^T (f^T)^{-1} E & hG^{-1}h^T (f^T)^{-1} \\ (f^T)^{-1} E & (f^T)^{-1} \end{bmatrix}, \quad K = \begin{bmatrix} -hG^{-1}h^T (f^T)^{-1} E \\ -(f^T)^{-1} E \end{bmatrix} \quad (2.49b)$$

La résolution se traduit par une solution de la forme :

$$\begin{bmatrix} x_i^p \\ \lambda_i^p \end{bmatrix} = H^p \begin{bmatrix} x_i^0 \\ \lambda_i^0 \end{bmatrix} + \begin{bmatrix} H^{p-1}K & \dots & H^{p-2}K & \dots & H^0K \end{bmatrix} \cdot \begin{bmatrix} c_{i-1}^0 \\ c_{i-1}^1 \\ \vdots \\ c_{i-1}^{p-1} \end{bmatrix} \quad (2.50)$$

Ce qui donne :

$$x_i^p = \Theta_{11} x_i^0 + \Theta_{12} \lambda_i^0 + I_h \quad (2.51a)$$

$$\text{avec } \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} = H^p \text{ et } \begin{bmatrix} I_h \\ I_b \end{bmatrix} = \begin{bmatrix} H^{p-1}K & \dots & H^{p-2}K & \dots & H^0K \end{bmatrix} \cdot \begin{bmatrix} c_{i-1}^0 \\ c_{i-1}^1 \\ \vdots \\ c_{i-1}^{p-1} \end{bmatrix}, \quad (2.51b)$$

$I_h \in R^{n \times 1}$  et  $I_b \in R^{n \times 1}$  étant les moitiés supérieure et inférieure de  $\begin{bmatrix} I_h & I_b \end{bmatrix}^T$ .

Dans ces conditions, la condition initiale de l'état du CBE à chaque commutation s'écrit :

$$\lambda_i^0 = \Theta_{12}^{-1} \left[ c_i^0 - \Theta_{11} x_i^0 - I^h \right] \quad (2.52)$$

En récapitulatif, si  $\Theta_{12}^{-1}$  existe, le CBE optimisé  $\Sigma_d(\{iT_e\}, \alpha, \beta_c, \beta_d, \gamma)$  est complètement défini par :

$$\lambda_i^0 = \beta_d \psi_i^0 \quad (2.53a)$$

$$\lambda_i^{k+1} = \alpha \lambda_i^k + \beta_c (c_{i-1}^k - x_i^k), \quad k = 0, 1, 2, \dots, p-1 \quad (2.53b)$$

$$u_i^k = w_i^k = \gamma \lambda_i^k \quad (2.53c)$$

avec :

- $\alpha = (f^T)^{-1}$  et  $\beta_e = (f^T)^{-1} E$  selon (2.48a),
- $\gamma = G^{-1} h^T$  selon (2.48b),
- $\beta_d = \Theta_{12}^{-1}$ , selon (2.52),
- $\psi(t) = c(t) - \Theta_{11} x(t) - I_h$  tel que  $\psi_i^0 = c_i^0 - \Theta_{11} x_i^0 - I_h$  aux instants de commutation.
- $\varphi(t) = c(t - T) - x(t)$ .

#### 2.4.2.3. Condition d'existence de $\Theta_{12}^{-1}$

Dans le cas où  $p$  tend vers l'infini et le produit  $pt_e$  reste fini égal à  $T_e$ , il est possible d'exprimer des conditions géométriques d'existence de  $\Theta_{12}^{-1}$  comme dans la section 2.3.2.4. Dans le cas où  $p$  est fini, nous ne disposons actuellement que de la condition d'existence numérique :  $\det(\Theta_{12}^{-1}) \neq 0$ .

#### 2.4.2.4. Architecture du CBE optimisé

D'après les résultats précédents, il est possible de réaliser la construction du CBE optimisé selon le schéma présenté figure 2.11.

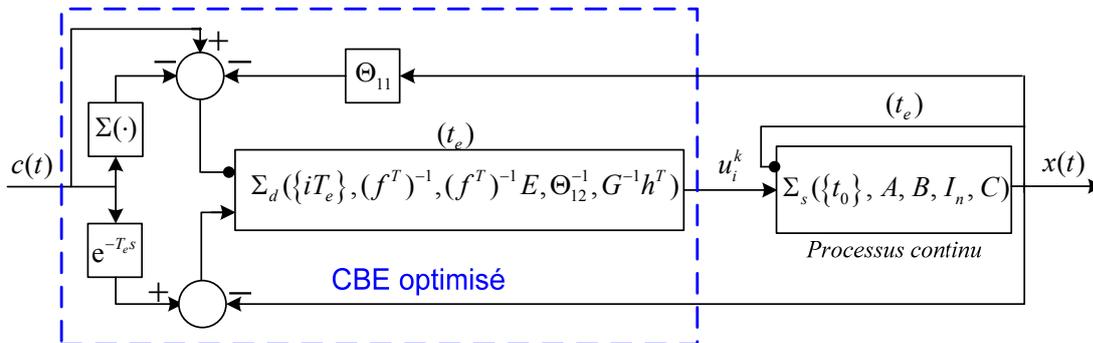


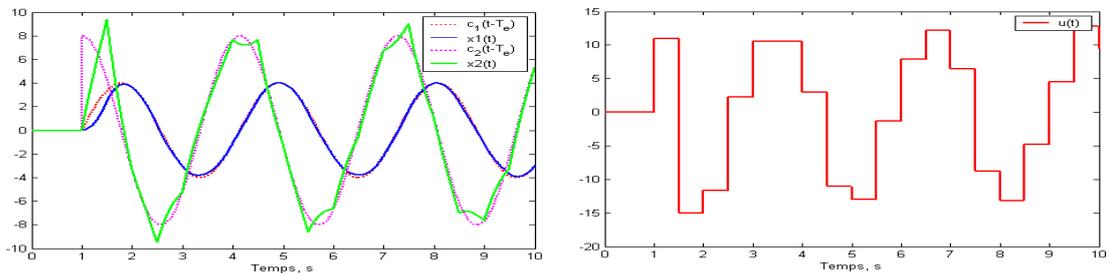
Fig. 2.11. Poursuite échantillonnée par un CBE optimisé

Sur cette architecture,  $\Sigma(\cdot)$  représente le calcul de  $I_h$  selon (2.51b). En pratique,  $I_h$  peut être interprété comme étant la sortie d'un SBE auxiliaire défini par  $\Sigma_d(\{iT_e\}, H, K, 0, [I_n \ 0])$  avec  $\psi(t) = 0$  et  $\varphi(t) = c(t - T_e)$ .

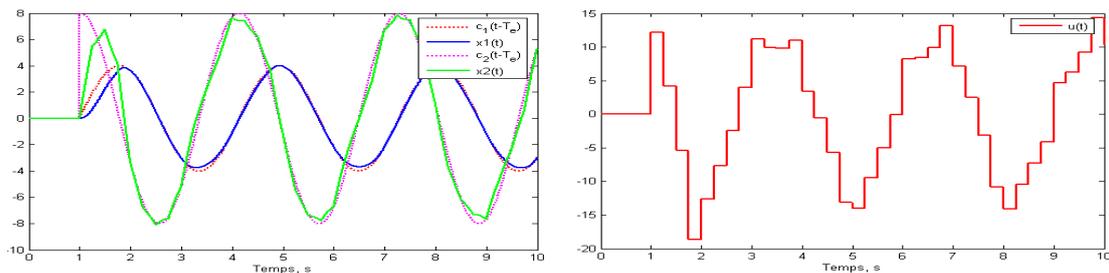
### 2.4.2.5. Validation expérimentale

Nous reprenons exactement le même cas que précédemment en utilisant ici une commande par CBE optimisé défini par l'architecture précédente avec  $E = 0.01 \times I_n$  et  $G = 100$ . En pratique, nous constatons que la performance d'un CBE optimisé est meilleure avec une faible valeur de  $E$  et une grande valeur de  $G$ .

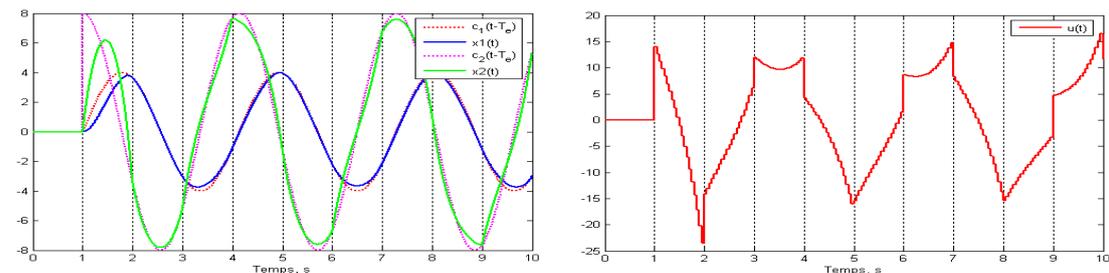
Les résultats donnés figure 2.12 offrent une comparaison avec le CBE non optimisé.



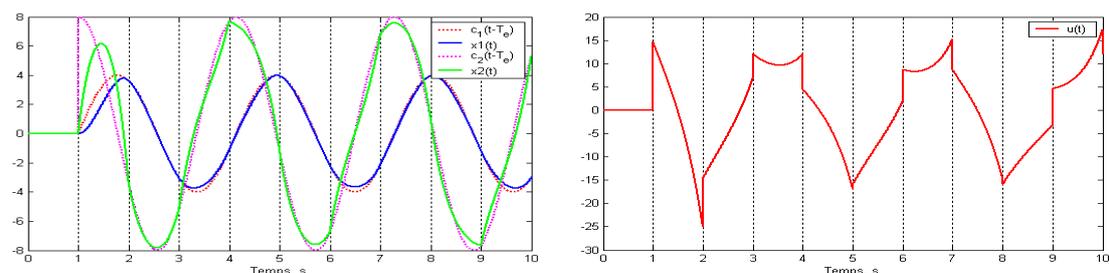
(a). CBE optimisé ( $p=2$ )



(b). CBE optimisé ( $p=4$ )



(c). CBE optimisé ( $p=20$ )



(d). CCM optimisé

**Fig. 2.12.** Poursuite échantillonnée par CBE optimisé avec  $T_e = 1s$

D'une part, nous constatons que l'évolution entre les instants de commutation est améliorée et d'autre part que la commande est fortement atténuée dans le cas optimisé.

L'utilisation de deux échelles de temps échantillonnée conduit à un formalisme spécifique permettant l'introduction du rapport  $p = T_e / t_e$ . Ceci ouvre la perspectives intéressante de gérer  $p$  de manière dynamique et ainsi de piloter les instants de commutation. Ainsi le pilotage de  $p$  apparaît comme un degré de liberté supplémentaire pour la commande du système. Comme précédemment, nous proposons une comparaison avec un CCM optimisé. Il est évident que le CBE optimisé tend vers le comportement du CCM optimisé pour une valeur élevée de  $p$ .

Par ailleurs, nous constatons qu'il existe une solution quasi-optimale en choisissant la valeur de  $p$  égale à l'ordre  $n$  du processus à commander : les résultats sont les mêmes sur les figures 2.10a et 2.11a pour un CBE non optimisé et un CBE optimisé. En effet, dans ce cas, il n'existe qu'une seule solution pour que la commande satisfasse la condition de poursuite. De ce fait, la solution est forcément la même dans les deux cas (optimisé ou non optimisé).

Le fonctionnement des CBE est quasiment le même que celui des CCM. La seule différence se trouve dans l'évolution échantillonnée de la commande entre les instants de commutation.

De plus, tout comme les CCM, les CBE sont robustes face aux bruits et aux variations des paramètres du processus et leur performance est meilleure dans le cas d'une commutation plus rapide.

### 2.4.3. Conclusion

Après lecture et comparaison des figures 2.3 et 2.8 dans le cas continu et des figures 2.10 et 2.12 dans le cas bi-échantillonné, il apparaît que, quand la période de commutation  $T_e$  est grande, l'utilisation des contrôleurs optimisés améliore sensiblement les performances.

Par contre, lorsque la période de commutation  $T_e$  devient très petite (tend vers zéro), le comportement des contrôleurs non optimisés devient très proche de celui des contrôleurs optimisés. Autrement dit, quand  $T_e$  est très petit, l'utilisation des contrôleurs optimisés n'apparaît plus comme nécessaire.

A partir de ces remarques, nous proposons d'examiner plus en détail le cas où  $T_e \rightarrow 0$ , pour définir un nouveau type de contrôleur appelé *Contrôleur Récursif*. Ce contrôleur, d'un très grand intérêt à la fois pour sa mise en œuvre aisée et ses performances, est présenté ci-après.

## 2.5. Poursuite de sortie : Contrôleur Récursif (CR)

Le formalisme de base des CFM permet d'imposer au processus une consigne d'état. De ce fait, il est nécessaire de définir de manière cohérente les valeurs souhaitées des variables d'état. Il faut soit faire un bon choix de l'évolution de chaque composante de la consigne d'état, soit adopter un *modèle de référence* qui génère une consigne cohérente.

Toutefois, il est possible de considérer l'application d'une consigne de sortie  $c_s(t) \in Y^m$  en adoptant une condition de la forme :

$$y((i+1)T_e) = c_s(iT_e), \quad \forall i = 0, 1, 2, \dots \quad (2.54)$$

où  $T_e$  est la période d'échantillonnage constante correspondant aux instants de commutation du SFM.

Pour simplifier, l'équation (2.54) est notée  $y_{i+1} = c_{s,i}$ . Elle conduit à réaliser la poursuite échantillonnée de  $c_s(t)$  par  $y(t)$ , avec un retard d'une période de commutation.

Nous proposons, dans ce paragraphe d'adapter le CCM non optimisé défini précédemment pour effectuer une poursuite de sortie, sans faire appel à un observateur. Pour cela nous faisons tendre la période de commutation vers zéro.

### 2.5.1. CCM non optimisé : adaptation à une consigne de sortie

Nous considérons un processus continu  $\Sigma_s(\{t_0\}, A, B, I_n, C)$  commandé par un CCM non optimisé défini par :

$$\Sigma_c(\{iT_e\}, \alpha, 0, M^{-1}, \gamma)$$

où  $M(A, B, \gamma, \alpha, T_e) = \exp(AT_e) \int_0^{T_e} \exp(-A\tau) B \gamma \exp(\alpha\tau) d\tau$ ,  $M \in R^{n \times m}$ .

Si l'on considère la sortie  $y(t) = Cx(t)$ , alors l'équation (2.5) conduit à :

$$y_{i+1} = C \exp(AT_e) x_i + CM \lambda_i^+ \quad (2.55)$$

Avec :  $\lambda_i^+ = \beta_d \psi_i$  d'après (2.2b).

Ainsi, en considérant la condition de poursuite donnée par (2.54), nous sommes amenés à définir  $\beta_d = (CM)^{-1}$  en faisant en sorte que  $CM$  soit une matrice carrée non singulière de dimension  $m \times m$ . Pour tenir compte de ce nouveau dimensionnement, nous redéfinissons les dimensions du contrôleur, comme suit :

- $\dim(\lambda(t)) = m$ ,
- $\dim(\alpha) = m \times m$ ,
- $\dim(\gamma) = r \times m$ .

Par ailleurs, l'équation (2.8b) devient  $\psi(t) = c_s(t) - C \exp(AT_e)x(t)$ .

## 2.5.2. Contrôleur Récursif (CR)

### 2.5.2.1. Mise en équation

En imposant au CCM une période de commutation  $T_e \rightarrow 0^+$ , il est possible de remplacer dans l'équation (2.55), les termes en exponentiel par un développement limité d'ordre 1 :

$$y_{i+1} = C(I_n + AT_e)x_i + C(I_n + AT_e) \int_0^{T_e} (I_n - A\tau) B \gamma (I_n + AT_e) d\tau \lambda_k \quad (2.56)$$

Ce qui donne :

$$y_{i+1} = y_i + CAT_e x_i + [CB\gamma T_e + \varepsilon(T_e^2)] \lambda_i, \quad (2.57)$$

$\varepsilon(T_e^2)$  étant négligeable pour  $T_e \rightarrow 0^+$ .

Compte tenu de la condition de poursuite  $y_{i+1} = c_{s,i}$ , il vient :

$$[CB\gamma T_e + \varepsilon(T_e^2)] \lambda_i = c_{s,i} - y_i - CAT_e x_i \quad (2.58)$$

Dans l'équation (2.58), la grandeur  $[CB\gamma T_e + \varepsilon(T_e^2)]$  représente la matrice (CM) supposé inversible. Cette équation peut être réécrite :

$$\lambda_i - \lambda_i + [CB\gamma T_e + \varepsilon(T_e^2)] \lambda_i = c_{s,i} - y_i - CAT_e x_i \quad (2.59)$$

Ce qui conduit à la forme récursive suivante :

$$\lambda_i = [I_m - CB\gamma T_e - \varepsilon(T_e^2)] \lambda_i + c_{s,i} - y_i - CAT_e x_i \quad (2.60)$$

De plus, lorsque  $T_e \rightarrow 0^+$ , on obtient :

$$\lambda_i = I_m^- \lambda_i + c_{s,i} - y_i \quad (2.61)$$

### 2.5.2.2. Interprétation algorithmique

D'un point de vue algorithmique, l'équation (2.61) traduit une évaluation récursive de  $\lambda_i$ , qui, d'un point de vue algorithmique, correspond à une opération de réaffectation à  $\lambda_i$  de la valeur  $I_m^- \lambda_i + c_{s,i} - y_i$ , jusqu'à convergence., soit :

$$\lambda_i \leftarrow I_m^- \lambda_i + c_{s,i} - y_i \quad (2.62)$$

**Remarque 2.3.** La condition de convergence de l'équation précédente est que  $I_m^-$  converge (valeurs propres inférieures à 1 en module).

**Remarque 2.4.** Pratiquement chaque pas du calculateur peut être de l'ordre du dixième de nanoseconde (c'est le cas sur le simulateur Matlab/Simulink). Sur une architecture temps réel, ce pas de calcul est de l'ordre du dixième de milliseconde (c'est le cas pour la technologie dSpace/Matlab/Simulink). Ces temps sont donc parfaitement compatibles avec la dynamique de la plupart des systèmes physiques.

### 2.5.2.3. Architecture du CR

Il est important de noter que, lorsque  $T_e \rightarrow 0^+$ , les signaux de  $c_{s,i}$ ,  $y_i$ ,  $\lambda_i$  peuvent être considérés comme des signaux continus  $c_s(t)$ ,  $y(t)$ ,  $\lambda(t)$ . Dans ces conditions, la relation (2.62) se traduit par le schéma fonctionnel bouclé de la figure 2.13.

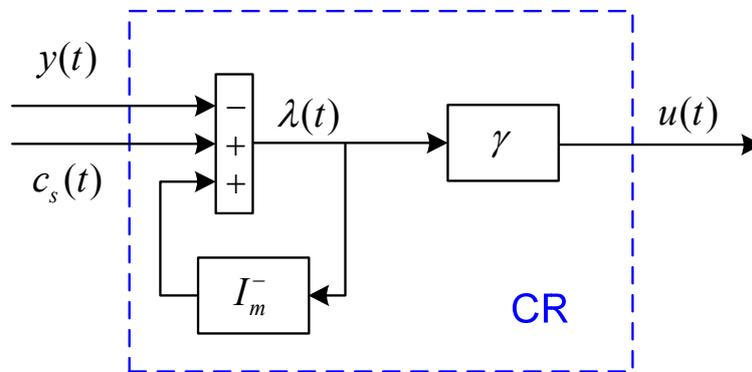


Fig. 2.13. CR à retour de sortie

L'équation (2.61), que traduit ce schéma, fait apparaître deux cas possibles :

- Dans le cas où  $y(t) = c_s(t)$ ,  $\lambda(t)$  existe, de valeur finie,
- Dans le cas où  $y(t) \neq c_s(t)$ , la seule solution possible est  $\lambda(t) = \infty$ .

Dans le premier cas, le CR garantit la poursuite échantillonnée de période  $T_e \rightarrow 0^+$ , avec un retard d'une période. Cela équivaut à une poursuite continue sans retard.

Le second cas se produit, soit au démarrage lorsque  $y(0) \neq c_s(0)$ , soit lorsque  $c_s(t)$  présente une discontinuité,  $y(t)$  étant obligatoirement continue. La solution  $\lambda(t) = \infty$  n'ayant pas de réalité physique, nous proposons, ci après, d'intégrer dans l'architecture de commande, un *amortisseur*, dont le rôle est d'absorber les discontinuités de  $c_s(t)$ .

### 2.5.3. CR avec amortisseur

#### 2.5.3.1. Amortisseur

L'idée est de modifier l'architecture de commande en incluant l'*amortisseur* présenté figure 2.14, où  $r(t)$  représente l'information de retour. Celle-ci peut être soit l'état  $x(t)$  ou la sortie  $y(t)$  du processus, selon le cas envisagé.

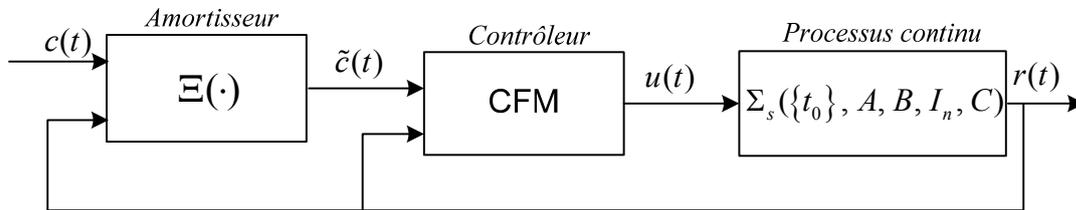


Fig. 2.14. CFM avec amortisseur

L'*amortisseur* a pour fonction de *lisser* la consigne, uniquement en cas de discontinuité, pour la transformer en une consigne  $\tilde{c}(t)$  continue. Lorsque le fonctionnement est *normal* (absence de discontinuité) l'amortisseur est inopérant ( $\tilde{c}(t) = c(t)$ ) et reste transparent. Cela se traduit mathématiquement par :

$$\tilde{c}(t) = r(t) + \zeta(c(t), r(t)) \cdot (c(t) - r(t)),$$

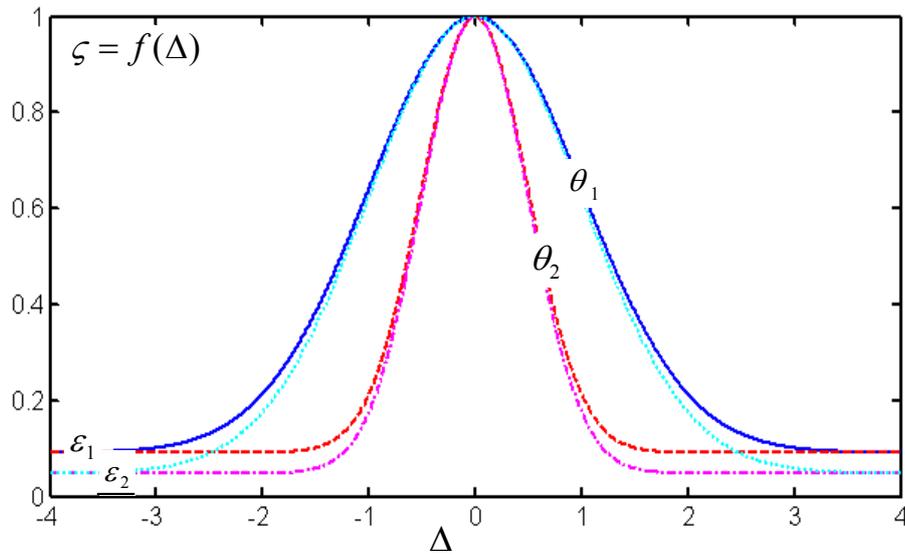
Le coefficient  $\zeta(\cdot)$  de l'amortisseur est une fonction scalaire telle que  $0 < \zeta(\cdot) \leq 1$ .

Dans le fonctionnement *normal* ( $r(t)$  est *accroché* à sa consigne retardée de  $T_e$ ),  $\zeta(\cdot) = 1$  de sorte que  $\tilde{c}(t) = c(t)$ . Autrement, si  $r(t)$  s'écarte de sa consigne,  $\zeta(\cdot)$  décroît vers zéro selon une fonction en cloche donnée par :

$$\zeta(\cdot) = \frac{1}{1 + \varepsilon} \left\{ \varepsilon + e^{-\frac{\theta}{2}(\Delta^T \Delta)} \right\}, \text{ avec } \Delta = c(t - T_e) - r(t), \quad (2.61)$$

$\varepsilon$  et  $\theta$  étant des réels positifs. Dans la suite, nous désignons l'amortisseur par  $\Xi(\varepsilon, \theta)$ .

La figure 2.15 donne l'allure de  $\zeta(\cdot)$  en fonction de l'écart  $\Delta$  qui sépare la sortie de sa consigne retardée de  $T_e$ .



**Fig. 2.15.** Coefficient de l'amortisseur ( $\theta_1 < \theta_2, \varepsilon_1 > \varepsilon_2$ )

L'amortisseur considéré  $\Xi(\varepsilon, \theta)$  est configurable à partir des valeurs de  $\varepsilon$  et  $\theta$ , afin de le rendre plus ou moins réactif.

La figure 2.15 donne les allures de quatre courbes d'amortisseur, avec  $\theta_1 < \theta_2, \varepsilon_1 > \varepsilon_2$  :

- courbe continue bleu :  $\Xi(\varepsilon_1, \theta_1)$ ,
- courbe en tirets en rouge :  $\Xi(\varepsilon_1, \theta_2)$ ,
- courbe en pointillés cyan :  $\Xi(\varepsilon_2, \theta_1)$ ,
- courbe en points-tirets magenta :  $\Xi(\varepsilon_2, \theta_2)$ .

### 2.5.3.2. CR avec amortisseur intégré

Nous avons intégré l'amortisseur  $\Xi(\varepsilon, \theta)$  dans l'architecture du Contrôleur Récursif (CR).

#### 2.5.3.2.1. Stratégie de commande

L'équation résultant de l'intégration de l'amortisseur dans l'architecture du contrôleur s'établit comme suit :

$$\lambda_i = \zeta I_m^- \lambda_i + c_{s,i} - y_i, \text{ avec : } 0 < \zeta \leq 1 \quad (2.62)$$

$\zeta(\cdot)$  est la fonction d'amortissement définie précédemment.

## 2.5.3.2.2. Architecture de commande

L'architecture de commande se déduit immédiatement de l'équation 2.62 :

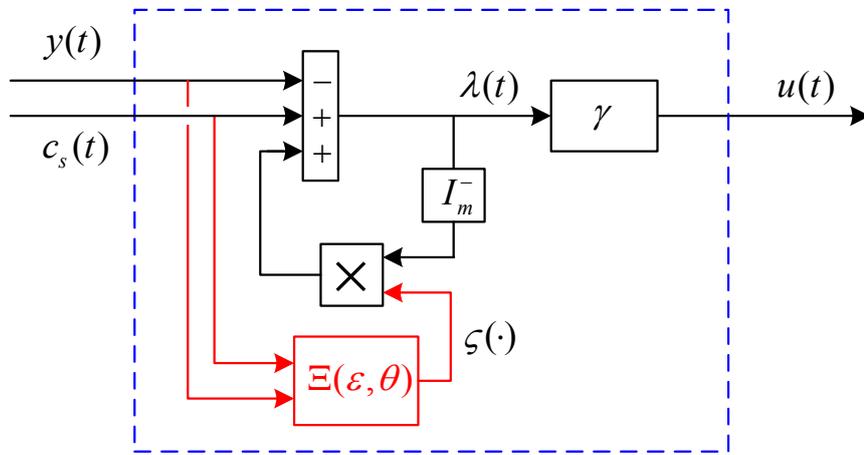


Fig. 2.16. CR avec amortisseur intégré

## 2.5.4. Performances du CR avec amortisseur intégré

### 2.5.4.1. Poursuite d'un processus linéaire ou changeant

#### 2.5.4.1.1. Poursuite d'un processus linéaire avec ou sans amortisseur

Nous commençons ici par considérer le chariot x-y modélisé en annexe I. La structure étant symétrique, nous ne présentons que la commande selon l'axe x. Ce système du type amplificateur-moteur-chariot est défini par :

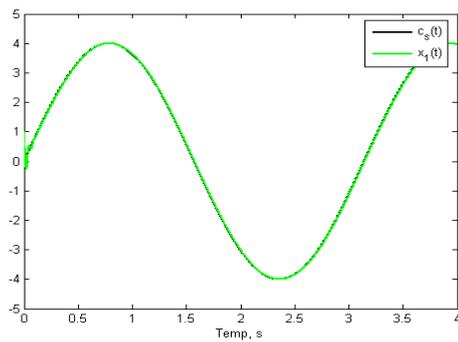
$$\Sigma_s(\{t_0\}, A, B_c, I_2, C),$$

$$\text{avec : } A = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau_x \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ K_x/\tau_x \end{bmatrix}, \quad C = [1 \quad 0], \quad x_0 = [1 \quad 0]^T.$$

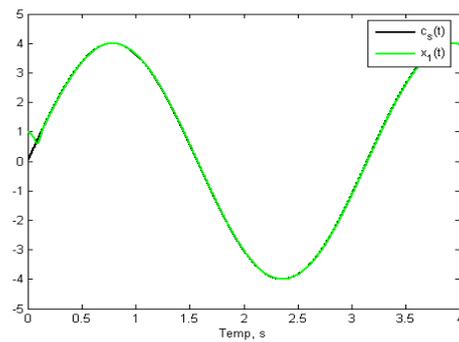
Les valeurs des paramètres  $1/\tau_x = 72$  et  $K_x/\tau_x = 190$  sont évaluées par la méthode d'identification développée dans [Cha 05, Cha 06].

La consigne de sortie est définie par  $c_s(t) = 4 \sin(2t)$ .

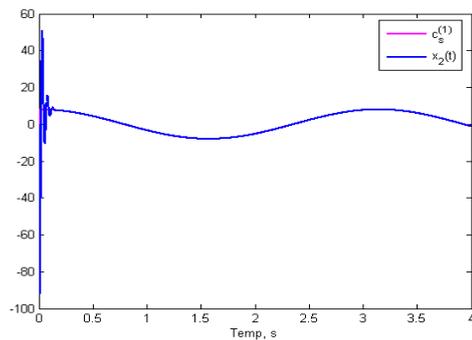
Nous présentons cet exemple en utilisant un CR sans ou avec amortisseur. Nous imposons  $I_m^- = 0.99$ ,  $\gamma = 1$  et  $\Xi(\varepsilon, \theta) = \Xi(0.1, 2.4)$ . Les résultats de simulation sont présentés respectivement figure 2.17 (sans amortisseur) et 2.18 (avec amortisseur).



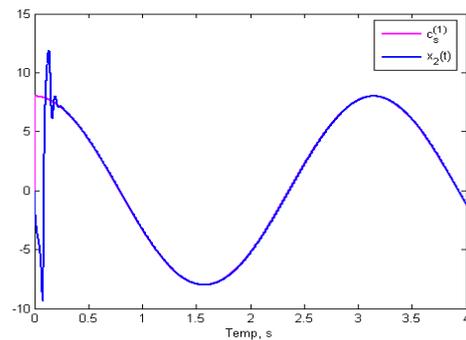
(a). Comparaison entre  $c_s(t)$  et  $x_1(t)$  (m)



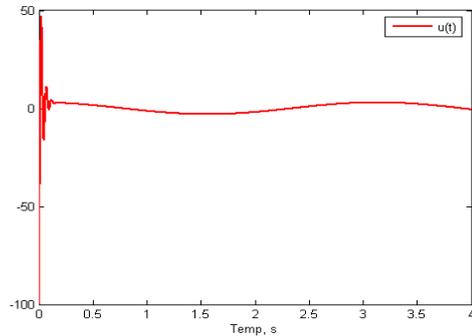
(a). Comparaison entre  $c_s(t)$  et  $x_1(t)$  (m)



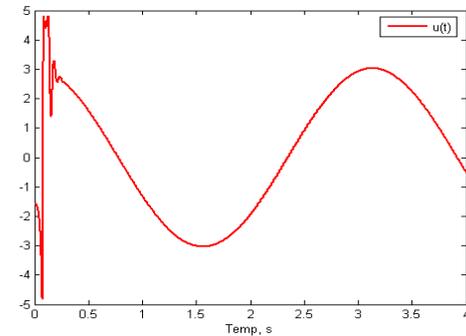
(b). Comparaison entre  $\dot{c}_s(t)$  et  $x_2(t)$  (m/s)



(b). Comparaison entre  $\dot{c}_s(t)$  et  $x_2(t)$  (m)

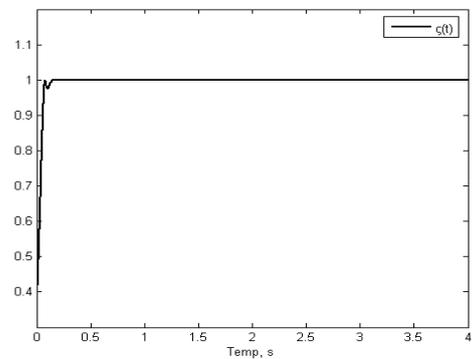


(c). Commande  $u(t)$  (V)



(c). Commande  $u(t)$  (V)

**Fig. 2.17. CR sans amortisseur**



(d). Amortisseur  $\zeta(t)$

**Fig. 2.18. CR avec amortisseur**

Pour chaque type d'asservissement, nous donnons : (a) l'évolution de la sortie du système (position), selon la consigne imposée, (b) l'évolution de la deuxième composante d'état (vitesse), (c) la commande correspondante. Dans le cas du CR avec amortisseur, nous présentons également l'évolution de  $\zeta(t)$  (2.18d).

Nous constatons que le CR assure non seulement la poursuite de la consigne de sortie, mais aussi la poursuite de la consigne dérivée. Nous constatons également que l'utilisation de l'amortisseur réduit l'amplitude de la commande aux instants de fortes variations (démarrage de la poursuite). Enfin,  $\zeta(t)$  est voisin de 1, lorsque l'état est *accroché* à sa consigne.

#### 2.5.4.1.2. Validation de la robustesse : poursuite d'un processus à paramètres changeants

Les CR ne font apparaître aucun paramètre du système. Ce type de commande est donc indépendant du paramétrage du système. Pour tester cette propriété remarquable, nous faisons varier les paramètres du système, en définissant :

$$A = \begin{bmatrix} 2 \sin(4t) & 1 \\ 0 & (-1/\tau_x) + 10 \sin(2t) \end{bmatrix}, \quad B_c = \begin{bmatrix} 2 \sin(t) \\ (K_x/\tau_x) + 20 \sin(4t) \end{bmatrix}.$$

avec  $1/\tau_x = 72$  et  $K_x/\tau_x = 190$ . Les résultats de simulation avec le même CR et le même amortisseur que précédemment sont présentés figure 2.19.

Nous constatons que, même si les paramètres du processus changent notablement, le CR avec amortisseur assure la poursuite de la consigne de sortie. La commande correspondante est propre et d'amplitude plus réduite que dans le cas du contrôleur sans amortisseur utilisé figure 2.17 pour un processus à paramètres fixes. Pour assurer la poursuite, tout en tenant compte de la variation des paramètres, la commande varie plus fréquemment que celle de la figure 2.18.

En conclusion, lorsque des discontinuités apparaissent sur la consigne,  $\zeta$  décroît vers zéro et l'amortisseur  $\Xi(\varepsilon, \theta)$  limite la commande. L'asservissement devient alors moins réactif car équivalent à un asservissement proportionnel (P) à gain réduit. Ensuite, lorsque le système est accroché à sa consigne,  $\zeta$  tend vers un et la commande devient indépendante de l'amortisseur  $\Xi(\varepsilon, \theta)$ .

Le CR garantit donc un asservissement robuste, même s'il existe des fortes discontinuités sur la consigne et si les paramètres du système varient.

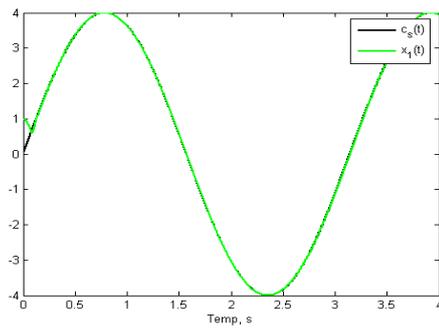
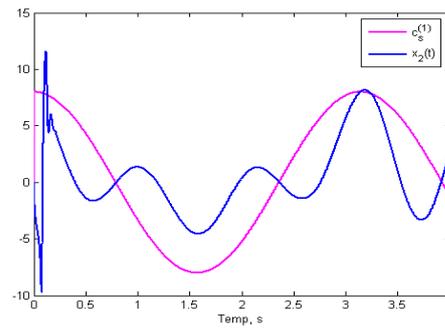
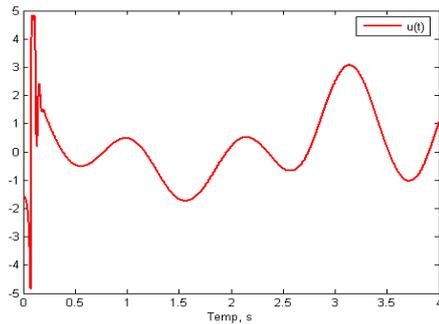
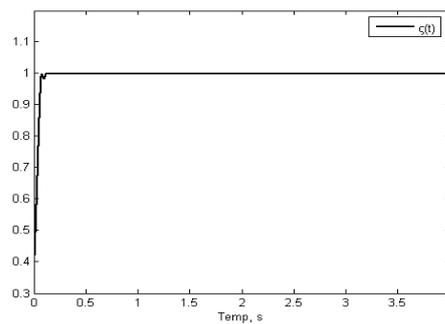

 (a). Comparaison entre  $c_s(t)$  et  $x_1(t)$  (m)

 (b). Comparaison entre  $\dot{c}_s(t)$  et  $x_2(t)$  (m/s)

 (c). Commande  $u(t)$  (V)

 (d). Amortisseur  $\zeta(t)$ 

Fig. 2.19. Poursuite à l'aide d'un CR, pour un système à paramètres changeants

### 2.5.4.2. Linéarisation d'un processus non linéaire

Dans cette partie, nous continuons de prendre le modèle de chariot x-y comme objet d'étude.

Pour un tel système mécanique, il existe différents types de frottement [Sel 02] : frottements de Coulomb, frottements visqueux et frottements de Stribeck [Dej 94, Wang 04, Che 04, Lin 06]. Contrairement à précédemment, nous considérons donc ici un modèle plus complexe faisant apparaître les coefficients de frottement de Coulomb  $f_c$ , de frottement visqueux  $f_v$  et de frottement de Stribeck  $v_s$ .

L'équation résultant est de la forme :

$$\ddot{x} = - \left\{ \frac{1}{\tau} \dot{x} + f_c \operatorname{sgn}(\dot{x}) + (f_c - f_v) \exp \left( - \left( \frac{\dot{x}}{v_s} \right)^2 \right) \right\} + \frac{K}{\tau} u \quad (2.63)$$

avec des paramètres imposés :  $1/\tau = 72$ ,  $f_c = 72$ ,  $f_s = 50$ ,  $v_s = 0.001$  et  $K/\tau = 190$ ,  $x$  étant la position du chariot selon l'axe  $x$ .

2.5.4.2.1. Architecture de linéarisation

L'idée de base est d'utiliser un Contrôleur Récursif pour faire suivre au système non linéaire l'état d'un système linéaire imposé à paramètres librement choisis.

Le système linéaire imposé a pour équation :

$$\ddot{x}_l = -\frac{1}{\tau_x} \dot{x}_l + \frac{K_x}{\tau_x} u, \quad (2.64)$$

avec les paramètres librement choisis :  $1/\tau_x = 120$  et  $K_x/\tau_x = 350$ .

L'architecture de linéarisation utilisant un CR à amortisseur intégré est illustrée figure 2.20.

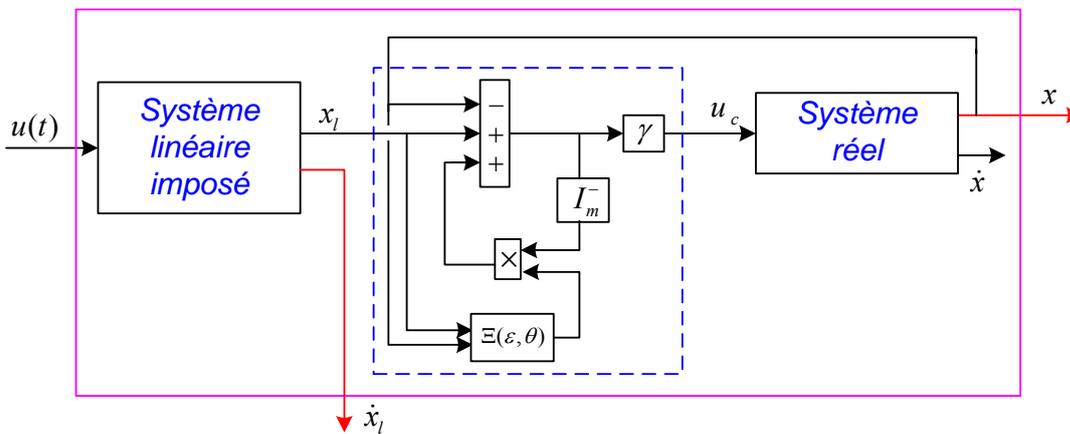


Fig. 2.20. Architecture de linéarisation

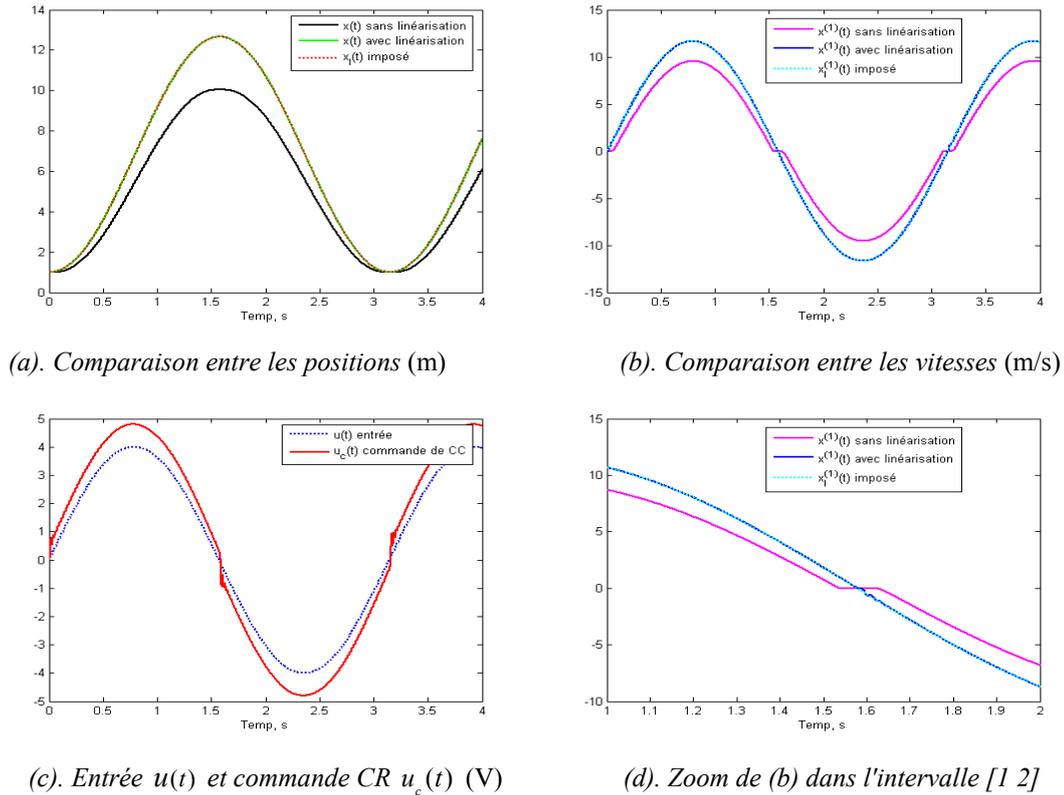
2.5.4.2.2. Résultats de simulation

D'après les résultats précédents, on sait que  $x=x_l$ . Il en résulte que la relation reliant  $x$  et  $u(t)$  est la même que celle reliant  $x_l$  et  $u(t)$ . Le système non linéaire est donc linéarisé selon le modèle du système linéaire imposé.

Cette méthode a été simulée sur Matlab/Simulink, avec le système non linéaire défini par l'équation 2.63 (condition initiale :  $(x_0, \dot{x}_0) = (1, 0)$ ) et le système linéaire imposé défini par l'équation 2.64. Le contrôleur CR est paramétré par  $I_m^- = 0.998$  et  $\gamma = 1$ . L'amortisseur est  $\Xi(\epsilon, \theta) = \Xi(0.1, 2.4)$ . Enfin, l'entrée est  $u(t) = 4 \sin(2t)$ .

Pour faciliter la lecture des performances de la méthode, nous avons également simulé le système non linéaire en boucle ouverte avec la même d'entrée  $u(t) = 4 \sin(2t)$ . Les résultats correspondants sont illustrés figure 2.21.

Nous constatons que, malgré les trois types de frottements, la position  $x$  et la vitesse  $\dot{x}$  du système non linéaire suivent parfaitement la position  $x_l$  et la vitesse  $\dot{x}_l$  du système


**Fig. 2.21.** Résultats de simulation de la linéarisation

linéaire imposé. Ainsi il apparaît que le contrôleur CR impose au système non linéaire un fonctionnement semblable à celui du système imposé.

Cette méthode de linéarisation va nous permettre de modéliser le chariot x-y selon chaque axe comme un système linéaire avec des paramètres parfaitement identifiés, selon des équations de la forme :

$$\ddot{x} = (-\dot{x} + k_x u_x) / \tau_x \quad (2.65a)$$

$$\ddot{y} = (-\dot{y} + k_y u_y) / \tau_y \quad (2.65b)$$

où  $k_x$ ,  $\tau_x$ ,  $u_x$  et  $x$  représentent respectivement le gain statique, la constante de temps, la tension d'entrée et la position selon l'axe  $x$ . De même pour  $k_y$ ,  $\tau_y$ ,  $u_y$  et  $y$  selon l'axe  $y$ . Cet aspect sera repris chapitre 4, pour la stabilisation du pendule inversé 2D et 3D.

### 2.5.4.3. Application au système réel : chariot x-y motorisé

Les performances du CR ont été testées sur la maquette décrite en annexe I. Dans ce paragraphe, nous nous intéressons aux retours de positions continus du chariot ( $x$ ,  $y$ ) utilisés pour réaliser une poursuite de sortie.

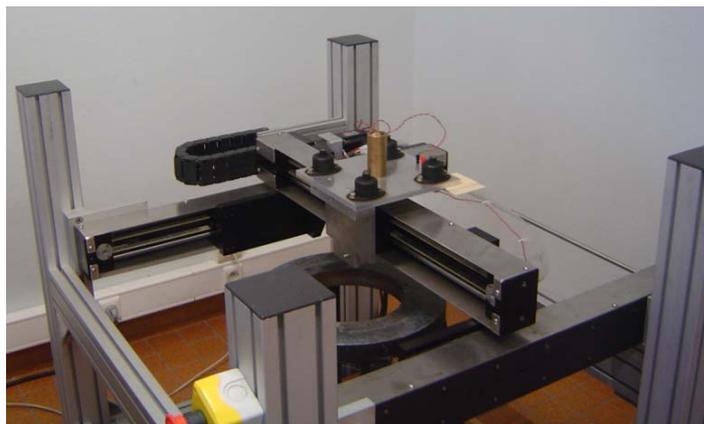


Fig. 2.22. Chariot x-y avec retour sur la sortie continue (codeurs)

Comme illustré figure 2.22, les axes  $x$  et  $y$  sont actionnées par deux servomoteurs. Un système de courroies crantées permet de réaliser les translations selon les axes  $x$  et  $y$ .

#### 2.5.4.3.1. Poursuite sans modèle

Afin de valider la commande proposée figure 2.16, nous définissons d'abord une consigne de sortie pour chaque axe sous la forme suivante :

$$\begin{bmatrix} c_{s,x} \\ c_{s,y} \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} m_x \sin \omega_x t \\ m_y \sin(\omega_y t + \varphi) \end{bmatrix},$$

dans laquelle :

$\alpha$  (rad) est un angle d'inclinaison,

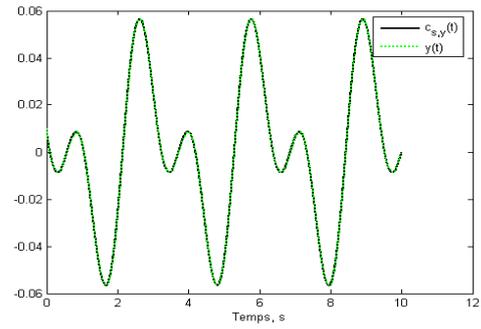
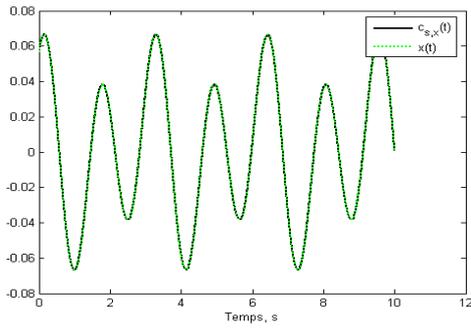
$m_x$  et  $m_y$  (m) sont les amplitudes selon chaque axe,

$\omega_x$  et  $\omega_y$  (rad/sec) sont les pulsations selon chaque axe, et

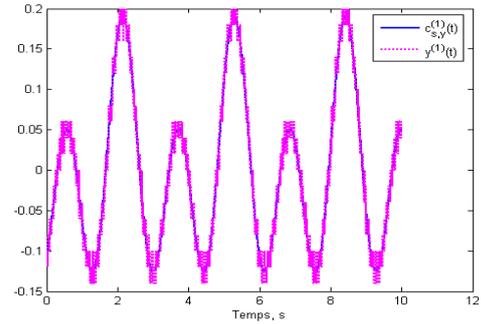
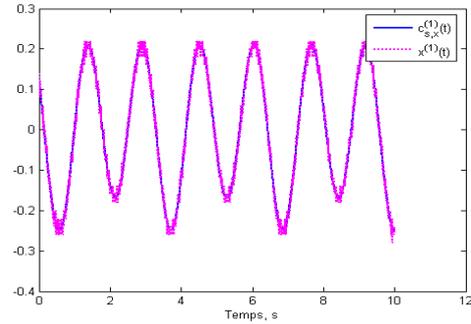
enfin,  $\varphi$  (rad) représente un déphasage.

Les contrôleurs CR utilisés sur chaque axe sont identiques avec  $I_m^- = 0.99$  et  $\gamma = 1$ . Nous choisissons aussi les mêmes valeurs pour les amortisseurs intégrés selon chaque axe :  $\Xi(0, 2.4)$ .

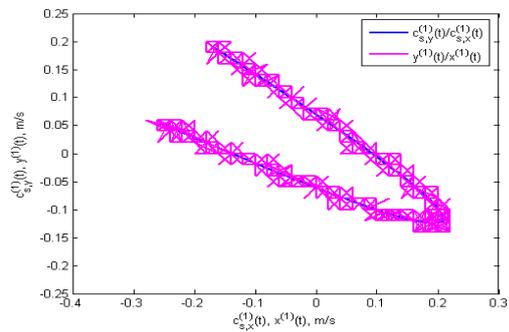
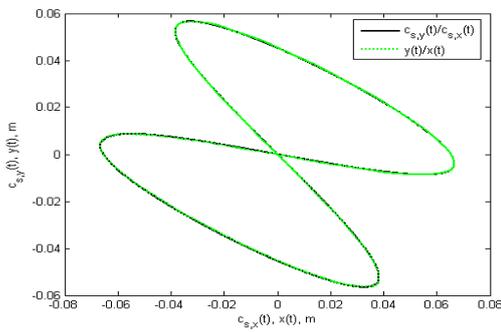
Les résultats expérimentaux avec  $(\alpha, m_x, m_y, \omega_x, \omega_y, \varphi) = (\pi/3, 0.04, 0.06, 2, 4, 0)$  sont présentés figure 2.23.



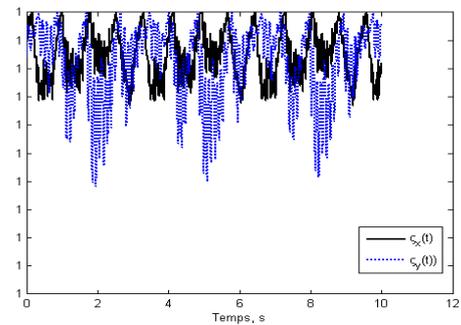
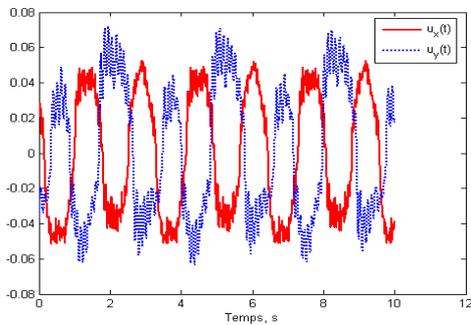
(a). Déplacements  $c_{s,x}(t)$  et  $x(t)$  selon l'axe  $x$  (m) (b). Déplacements  $c_{s,y}(t)$  et  $y(t)$  selon l'axe  $y$  (m)



(c). Vitesses  $\dot{c}_{s,x}(t)$  et  $\dot{x}(t)$  selon l'axe  $x$  (m/s) (d). Vitesses  $\dot{c}_{s,y}(t)$  et  $\dot{y}(t)$  selon l'axe  $y$  (m/s)



(e).  $(c_{s,x}(t), c_{s,y}(t))$  et  $(x(t), y(t))$  (m) (f).  $(\dot{c}_{s,x}(t), \dot{c}_{s,y}(t))$  et  $(\dot{x}(t), \dot{y}(t))$  (m/s)

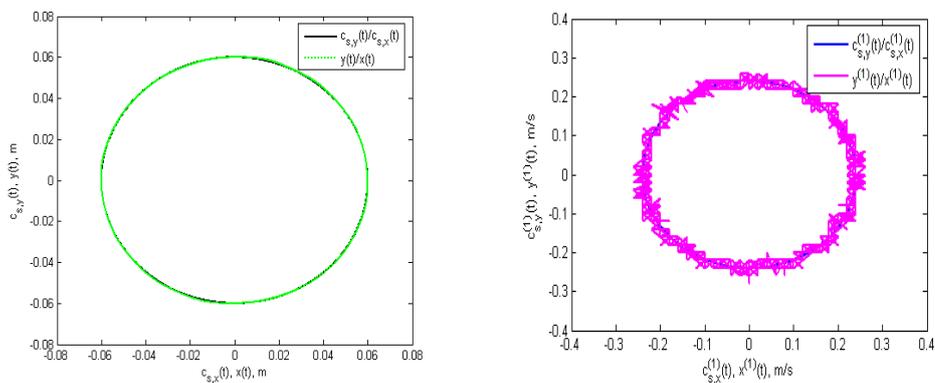


(g).  $(u_x(t), u_y(t))$  (V) (h). Amortisseur  $\zeta(t)$  selon chaque axe

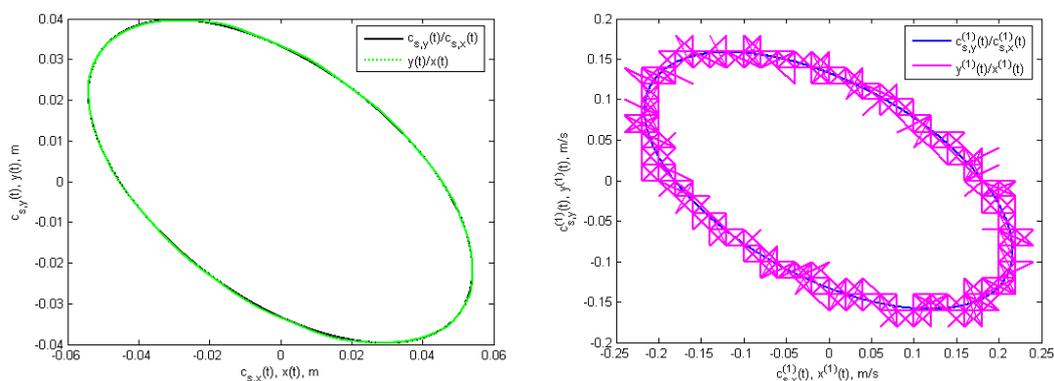
**Fig. 2.23.** Poursuite échantillonnée avec retour sur la sortie par CR avec amortisseur intégré

La figure 2.23e montre une excellente superposition de la consigne et de la sortie dans le plan de phase. S'agissant des vitesses (figure 2.23f), les performances restent acceptables.

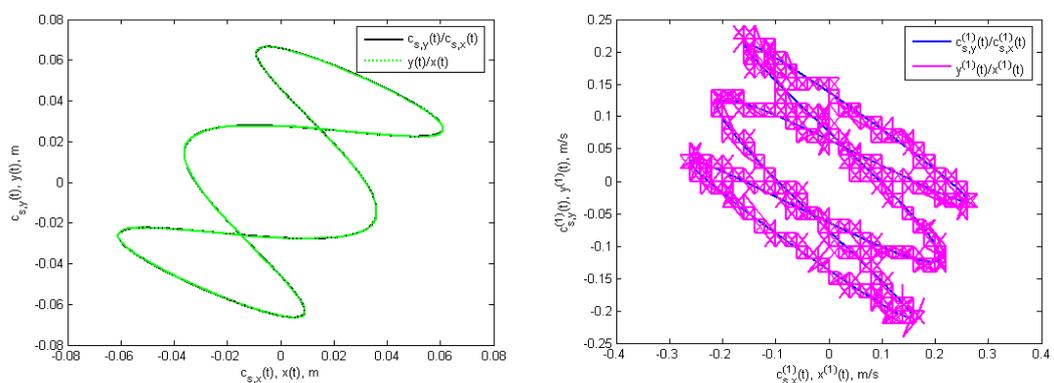
En gardant les mêmes CR à amortisseur intégré, nous avons également fait varier les paramètres de consigne, pour obtenir un cercle, une ellipse et une courbe de Lissajous. Les résultats expérimentaux apparaissent sur les figures 2.24, 2.25 et 2.26.



**Fig. 2.24.** Poursuite d'un cercle  $(\alpha, m_x, m_y, \omega_x, \omega_y, \varphi) = (\pi/3, 0.06, 0.06, 4, 4, \pi/2)$



**Fig. 2.25.** Poursuite d'une ellipse  $(\alpha, m_x, m_y, \omega_x, \omega_y, \varphi) = (\pi/3, 0.03, 0.06, 4, 4, \pi/2)$



**Fig. 2.26.** Poursuite d'une courbe de Lissajous  $(\alpha, m_x, m_y, \omega_x, \omega_y, \varphi) = (\pi/3, 0.06, 0.04, 2, 6, \pi/2)$

Après lecture des résultats expérimentaux, nous pouvons remarquer que le contrôleur CR assure la poursuite non seulement des positions, mais également des vitesses. Quand les positions tendent vers les consignes, l'amortisseur  $\zeta(t)$  tend vers un. Il devient donc transparent.

#### 2.5.4.3.2. Validation de la méthode de linéarisation

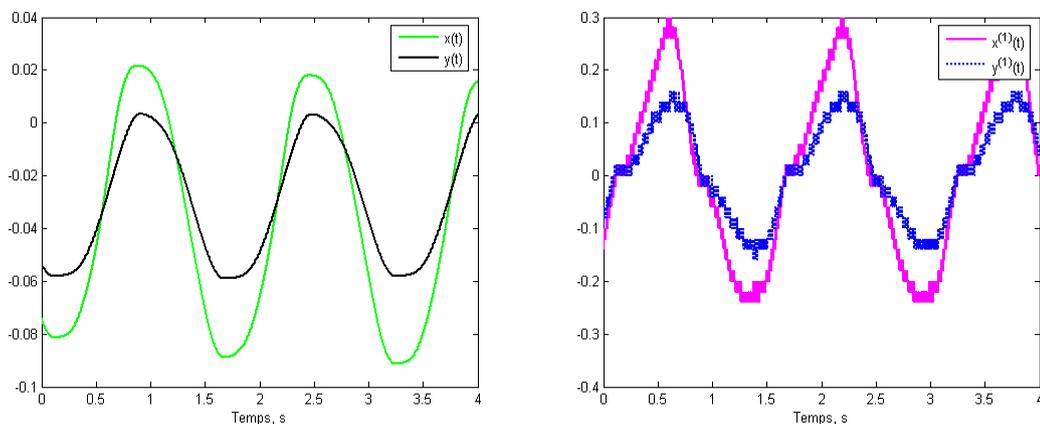
Nous reprenons ici la figure 2.20 qui donne l'architecture de linéarisation. Dans ce cadre, le système réel est le chariot x-y illustré de la figure 2.22, et les systèmes linéaires imposés selon chaque axe (x ou y) sont définis respectivement par les équations 2.65a et 2.65b.

Afin de valider notre méthode nous imposons  $k_x = k_y = 120$  et  $\tau_x = \tau_y = 35/12$  comme dans §2.5.4.2.1.

Les contrôleurs CR selon chaque axe sont identiques, avec  $I_m^- = 0.99$  et  $\gamma = 1$ , ainsi que les amortisseurs  $\Xi(\varepsilon, \theta) = \Xi(0, 2.4)$ .

Finalement, nous choisissons l'entrée  $u(t) = 0.05 \sin(4t)$  (V).

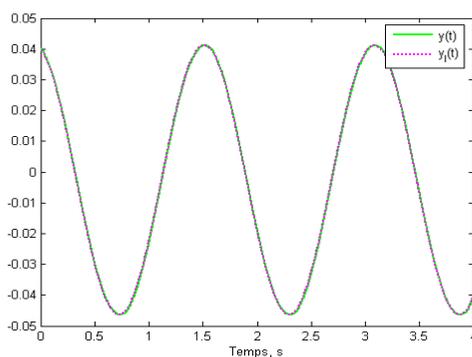
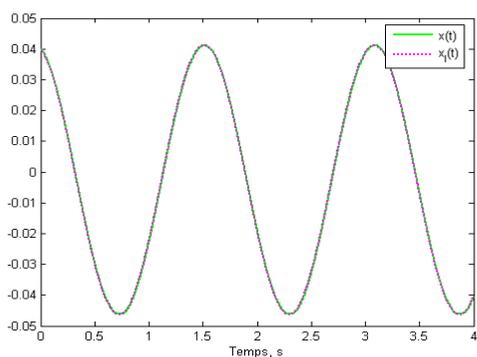
Pour faciliter la lecture des performances de la méthode linéarisation, nous présentons d'abord la figure 2.27, dans laquelle nous appliquons directement  $u(t)$  à l'entrée du système.



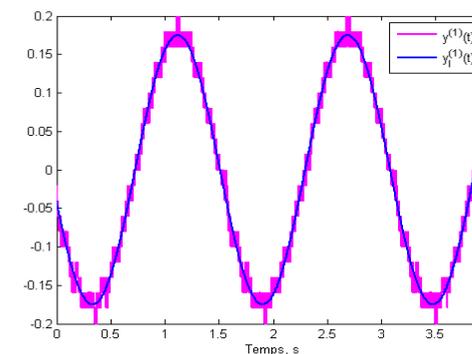
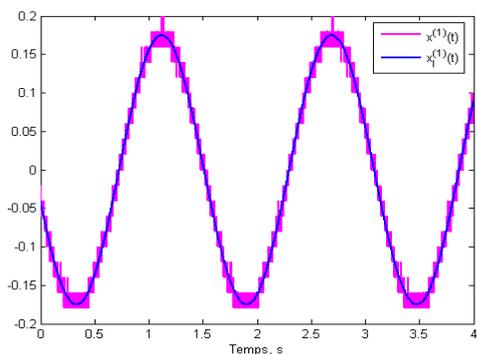
(a). Positions  $x$  et  $y$  sans linéarisation (m) (b). Vitesses  $\dot{x}$  et  $\dot{y}$  sans linéarisation (m/s)

**Fig. 2.27.** Application directe de  $u(t) = 0.05 \sin(4t)$  (V) à l'entré du chariot x-y

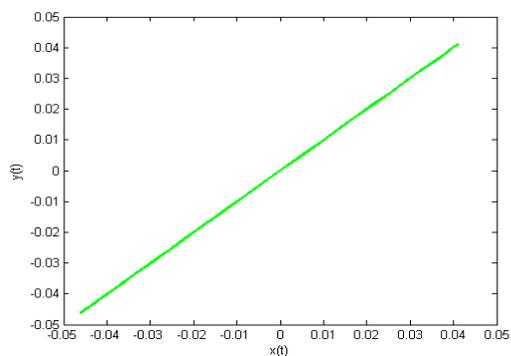
Nous pouvons constater que les paramètres selon chaque axe sont différents et qu'il existe des frottements de Coulomb, des frottements visqueux et des frottements de Stribeck [Sel 02]. Les résultats expérimentaux après application de la méthode linéarisation sont illustrés figure 2.28.



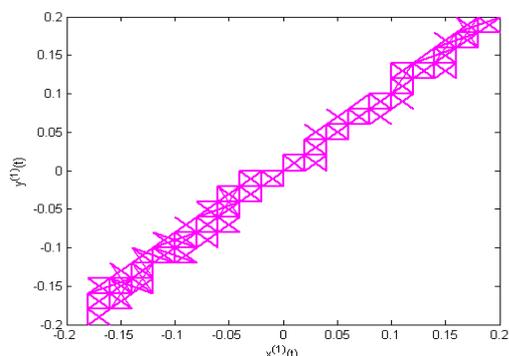
(a). Comparaison entre les positions selon l'axe x (m) (b). Comparaison entre les positions selon l'axe y (m)



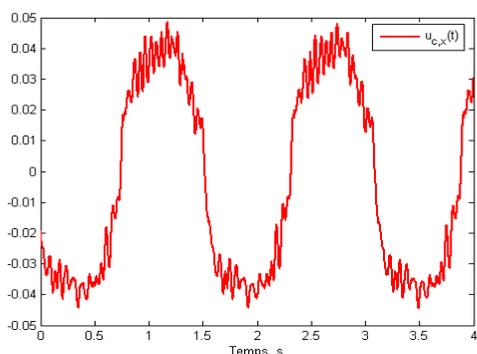
(c). Comparaison entre les vitesses selon l'axe x (m/s) (d). Comparaison entre les vitesses selon l'axe y (m/s)



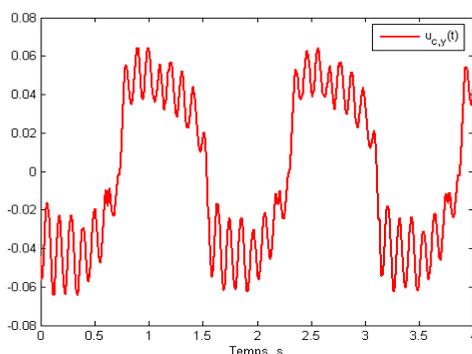
(e).  $(x(t), y(t))$  (m)



(f).  $(\dot{x}(t), \dot{y}(t))$  (m/s)



(g.) Commande  $u_{c,x}(t)$  (V)



(h). Commande  $u_{c,y}(t)$  (V)

**Fig. 2.28.** Validation expérimentale de la méthode de linéarisation

Les résultats montrent que le système chariot x-y non linéaire fonctionne, selon chaque axe, exactement de la même façon que les systèmes imposés. Les non linéarités ont complètement disparu. De plus, en choisissant les mêmes paramètres libres pour chaque axe, le fonctionnement devient totalement symétrique.

La méthode proposée permet donc d'éliminer très simplement les frottements, en comparaison d'autres méthodes [Wan 04, Che 04, Lin 06].

## 2.6. Conclusion

Dans un premier temps, nous avons proposé, dans ce chapitre, une architecture de commande utilisant deux boucles de retour basée utilisant les SFM présentés au chapitre précédent. A partir de cette architecture, nous avons proposé deux types de Contrôleurs à Fonctionnement par Morceaux (CFM) : les Contrôleurs Continus à fonctionnement par Morceaux (CCM) et les Contrôleurs Bi-Echantillonnés (CBE). Ces deux types de contrôleurs garantissent la poursuite échantillonnée d'une trajectoire d'état à partir de l'état du processus à commander ( $x_{i+1} = c_i$ ).

Dans le cas des CFM non optimisés la poursuite n'est garantie qu'aux instants de commutation. Le retour n'est pris en compte qu'à ces instants et le processus évolue en roue libre ailleurs. Lorsque la période de commutation est élevée, on peut mettre en œuvre les CFM optimisés qui minimisent l'écart entre l'état et sa consigne entre les instants de commutation.

Dans un deuxième temps, à partir d'un CCM non optimisé à commutation rapide ( $T_e \rightarrow 0$ ), nous avons développé un contrôleur spécifique appelé Contrôleur Récursif (CR), dans lequel nous avons intégré un amortisseur, qui permet d'éviter une commande excessive, lorsque la consigne présente des discontinuités. L'intégration de cet amortisseur fait apparaître des propriétés intéressantes. En particulier, le contrôleur CR assure non seulement la poursuite d'une trajectoire de sortie, mais il révèle aussi une excellente aptitude à réaliser la linéarisation de systèmes non linéaires, tout en choisissant librement les valeurs de paramètres de linéarisation.

Toutes les méthodes proposées sont illustrées par des exemples de simulations, qui montrent que le contrôleur CR à amortisseur intégré est robuste et performant, même en présence de perturbations et de variations des paramètres du processus à commander.

Afin de mieux valider nos méthodes, nous avons appliqué le contrôleur CR au système réel « chariot x-y » en réalisant la poursuite et la linéarisation des paramètres du système.

Les résultats obtenus confirment la robustesse et les performances du contrôleur CR.



## Chapitre 3

# Observateur continu par morceaux : application aux systèmes visuels

*L'un des objectifs de cette thèse est de développer et de proposer les méthodes d'observation à partir de la sortie retardée et échantillonnée. Il semble donc essentiel de proposer, dans un premier temps, une méthode de synthèse d'observateurs à partir de la sortie retardée ou/et échantillonnée. Les Observateurs Continus Par Morceaux (OCM) introduits dans ce chapitre s'imposent généralement sur les SFM et l'Observateur de Luenberger Discret d'Ordre Réduit (OLDOR). Ensuite, nous avons proposé une architecture de poursuite intégrant les CFM à l'aide d'un OCM. Enfin, l'OCM est validé par une application à la poursuite d'un chariot x-y motorisé ayant pour la boucle de retour une caméra CCD placée au dessus de la table.*

*L'OCM proposé dans ce chapitre est également utilisé pour les systèmes non linéaires mécaniques sous actionnés à déphasage non minimal dans le chapitre suivant.*

### 3.1. Introduction

Dans les premières parties de ce mémoire, nous avons développé les Contrôleur à Fonctionnement par Morceaux en se basant les SFM dans le but de réaliser une poursuite échantillonnée coïncidant avec une consigne de l'état ou de la sortie aux instants de commutation. Toutes ces applications requièrent un retour d'état ou de sortie (une partie de l'état) sans aucun retard. Néanmoins, dans certaines applications, le retard n'est pas évitable et l'état n'est pas toujours disponible.

On constate que la plupart des commandes actuellement implantées sont réalisées à l'aide des calculateurs numériques. Par conséquent, même si le processus à contrôler ne contient pas de retard intrinsèque, bien souvent des retards apparaissent dans la boucle de commande par l'intermédiaire des temps de réaction des actionneurs (1) ou des capteurs (2), des temps de transmissions des informations (3) ou des temps de calculs (4). Ces retards peuvent quelquefois être négligés, mais lorsque leur taille devient significative au regard des performances temporelles du système dynamique (en boucle ouverte et fermée), il n'est plus possible de les ignorer. On retrouve le cas idéal du banc d'essai de plateforme 2D : un chariot x-y à l'asservissement visuel illustré en détail dans l'annexe I,

- Il s'agit d'un pendule inversé fixé dont le point bas se déplace sur le plan horizontal (plan x-y). Sur l'extrémité du point haut de la barre du pendule se situe une LED infrarouge reconnue par la caméra. Dans une première phase, les informations échantillonnées issues du capteur caméra<sup>1</sup> doivent être interprétées et modifiées en vue de délivrer des données utilisables par le contrôleur. Cette opération d'extraction des coordonnées  $(x_C, y_C)$  nécessite un temps de calcul non négligeable qui introduit des retards dans la boucle de commande :

$$\mathcal{G}^*(t) = (x_C^*(t - T_e), y_C^*(t - T_e))$$

où \* représente un échantillonnage de période  $t_e$  (durée d'une prise d'image) et  $T_e = pt_e$  (avec  $p$  un entier) correspond au temps nécessaire pour le traitement d'image et pour la transmission des signaux. Cela implique pour le système de vision, un certain nombre  $p$  d'images pour obtenir l'information requise.

---

<sup>1</sup> Parce que la cadence de la caméra qui détermine la fréquence de la prise d'image est limitée, les coordonnées  $(x_C, y_C)$  sont sous forme échantillonnée.

En effet, dans la littérature, la commande d'un système nécessite souvent la connaissance de l'état complet alors qu'en pratique la mesure de l'ensemble des variables est difficile. Ces limites proviennent de considérations technologiques, lorsqu'il est impossible de mesurer une donnée, ou économiques, lorsque la précision de la mesure d'un capteur est nécessaire, le prix devient souvent élevée, ou finalement pratiques, sensibilité au bruit, etc.

Par conséquent, les principaux cas que nous allons envisager dans ce chapitre sont ceux pour lesquels le seul retour disponible est :

- la sortie retardée et échantillonnée,

C'est la situation qui nous paraît le plus représentative et épanou.

Pour les cas suivants :

- Relié à l'état retardé et ou échantillonné,
- Relié à la sortie retardée ou échantillonnée,

nous considérons que c'est les cas particuliers du premier cas.

Il est évident que l'utilisation de la sortie retardée et échantillonnée peut poser des problèmes dans la construction de la commande. De plus, il est légitime de penser que plus on dispose d'information sur l'état d'un système, plus la construction d'une loi de commande sera aisée.

Les travaux présentés ici constituent un prolongement et développement de recherches précédentes [Kon 03, Cha 05a, Wan 05, Cha 06, Wan 06, Wan 06a Wan 07]. L'enjeu majeur de ce chapitre est donc de reconstruire l'état continu sans aucun retard dans le but d'améliorer les performances de la commande à partir de la sortie retardée et échantillonnée.

L'analyse des difficultés liées à la commande 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 qu'on appelle Observateur Continu par Morceaux (OCM), basé sur l'utilisation des Systèmes à Fonctionnement par Morceaux (SFM), présente l'avantage de compenser les effets de retard et d'échantillonnage pour fournir des estimations continues et sans retard [Wan 07a, Wan 08, Wan 08d, Wan 08e]..

Par ailleurs, afin de valider notre méthode, nous l'avons appliquée dans le système du chariot x-y à retour visuel réalisant une poursuite robuste. Cette méthode qui intègre le

contrôleur de CR présente des performances intéressantes même si le retour d'informations visuelles soit perdu durant quelques instants. De nos jours, les outils de commande par l'ordinateur associés aux déplacements selon deux axes dans le domaine de fabrications industriel sont largement utilisés [Kor 82, Lin 04]. Contrairement à méthodes présentées dans [Lin 04, Wang 04] basées sur la théorie de la logique floue ou les réseaux de neurones, toutes les méthodes présentées dans ce chapitre sont basées sur les SFM.

## 3.2. Formulation du problème

A partir du chariot x-y à retour visuel, nous généralisons notre modèle à commander comme un système linéaire MIMO sous la forme de  $\Sigma_s(\{t_0\}, A, B, I_n, C)$  défini par la représentation d'état d'ordre  $n$  comme suit :

$$x'(t) = Ax(t) + Bu(t) \quad (3.1a)$$

$$y(t) = Cx(t) \quad (3.1b)$$

avec :  $A \in R^{n \times n}$ ,  $B \in R^{n \times r}$ ,  $I_n \in R^{n \times n}$ ,  $C \in R^{m \times n}$  : matrices usuelles de la représentation d'état.  $u(t) \in U^r$ ,  $x(t) \in \Sigma^n$  et  $y(t) \in Y^m$  désignent respectivement l'entrée, l'état et la sortie du processus.

Il faut souligner que le système considéré  $\Sigma_s(\{t_0\}, A, B, I_n, C)$  est commandable et observable.

D'ailleurs, d'après notre problématique, nous considérons que la seule information dont on peut disposer est la sortie (une partie de l'état) retardée et échantillonnée définie comme :

$$\mathcal{G}^*(t) = y^*(t - T_e), \quad (3.2)$$

dans lesquels \* représentant un échantillonnage de période  $t_e$  et  $T_e = pt_e$  (avec  $p$  un entier) étant le retard sur l'information de retour.

De plus, si l'on adopte deux échelles de temps  $t_i^k$  ( $t_i^k = iT_e + kt_e$ ) définis dans § 1.3.4 pour décrire la sortie retardée et échantillonnée, on a :

$$\mathcal{G}_i^k = y_{i-1}^k \quad (3.3)$$

---

<sup>2</sup> Pour les cas suivants:  $x(t - T_e)$ ,  $x_{i-1}^k$ , et  $y(t - T_e)$ , nous les considérons comme les cas particuliers de (3.3).

Dans le cas plus particulier comme notre système de vision, c.-à-d. dans le cas  $p=1$ ,  $T_e = t_e$ , on note simplifierent  $t_i^k$  et  $\theta_i^k = y_{i-1}^k$  comme  $t_k$  ( $t_k = kt_e$ ) et  $\mathcal{G}_k = y_{k-1}$ .

### 3.3. Synthèse d'un observateur linéaire

#### 3.3.1. Introduction à l'observateur

Au cours des dernières décennies, une partie importante des activités de recherche en automatique, s'est focalisée sur le problème de l'observation des systèmes dynamiques linéaires. Ceci est motivé par le fait que l'estimation de l'état est une étape importante voir indispensable pour la synthèse de lois de commande, pour l'identification des paramètres et pour le diagnostic ou la supervision des systèmes industriels [Bes 07], illustré par la figure 3.1.

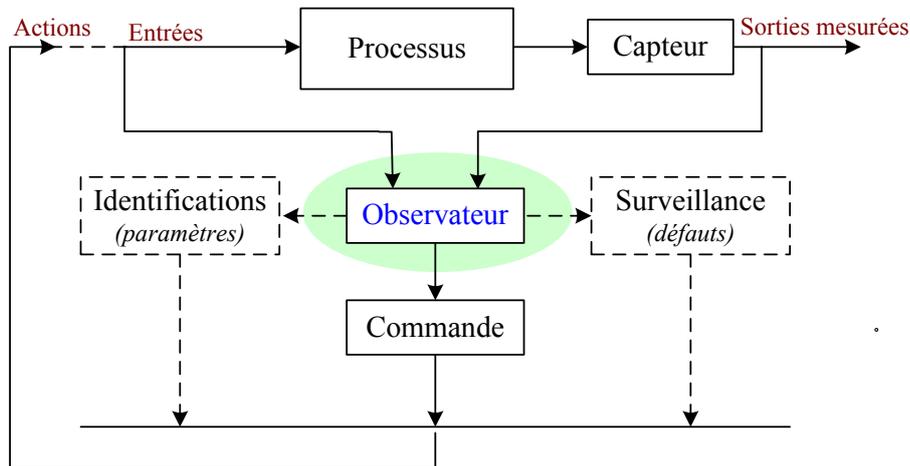


Fig. 3.1. Observateur: au centre des systèmes dynamiques

Un observateur d'état déterministe a été introduit dans les années soixante par Luenberger [Lue 66, Lue 71] pour les systèmes linéaires continus. Kalman [Kal 61] a également formulé un observateur en considérant un système linéaire déterministe ou stochastique. Dans le cas de l'observateur de Luenberger ou de Kalman il suffit de choisir  $L$  telle que la matrice  $(A-LC)$  soit une matrice Hurwitz, c'est-à-dire que ses valeurs propres soient toutes à parties réelles négatives dans le cas continu ou possèdent un module inférieur à 1 dans le cas discret. La différence entre les deux observateurs se trouve principalement dans la façon de calculer le gain de retour  $L$  :

- pour l'observateur de Luenberger, on utilise la technique de déplacement des pôles,

- pour le filtre de Kalman, on résout une équation de Ricatti en minimisant la matrice de covariance de l'erreur d'estimation et de perturbation.

Plus récemment, certains auteurs proposent des observateurs pour les systèmes linéaires invariants en retard dans le cas continu [Cho 96, Dar, 01], dans le cas discret [Tri 97, Dar 01, Dar 05], ou dans le cas stochastique [Kol 99, Zha 06]. Dans ce cadre, un aperçu global est dans [Sen 01, Ric 04].

D'ailleurs, en concernant les contraintes annoncées dans la section précédente, on s'intéresse ici plutôt à des observateurs reconstituant l'état continu à partir de la sortie retardée et échantillonnée. Néanmoins, dans le cadre d'applications réelles (commande téléopérée, systèmes en réseau par exemple), le retard et la période d'échantillonnage sont considérables, ils ont une influence forte sur la stabilité et les performances du système [Lel 01, Plo 04]. Une autre méthode [Fri 04, Seu 06, cha 06a] introduisant une écriture spéciale en modélisant les mesures retardés et échantillonnés sous la forme continue retardée est susceptible d'éliminer les effets d'échantillonnage. Ensuite les auteurs adoptent les techniques de continu pour construire un observateur de type Luenberger en se basant les techniques de Lyapunov-Krasovskii. Son gain de retour est calculé par les techniques d'optimisation d'Inégalité Matricielle Linéaire (IML). D'après [Bus 99], les observateurs avec un retour de gain constant sont plus intéressants d'un point de vue d'application sur les systèmes réels (le temps de calcul, la capacité de mémoire, etc). L'observateur présenté dans [Seu 06] et notre OCM correspondent à ce critère.

### 3.3.2. Observateur prédicteur de Seuret [Seu 06]

Dans cette section nous développons l'idée principale d'observateur de [Seu 06, cha 06a] afin de faciliter la comparaison de ses performances avec notre OCM.

Concernant notre problématique (3.3), la sortie retardée et échantillonnée  $y_{i-1}^k$  peut être notée comme  $y_{i-1}^k = y(t - \delta(t))$  où le retard  $\delta(t)$  lié à l'échantillonnage est défini comme  $\delta(t) = t - t_{i-1}^k = T_e + t - t_i^k$ , avec  $T_e$  retard et  $t_i^k = iT_e + kt_e$  instant d'échantillonnage le plus récent. Il faut souligner que la méthode originale admet un retard aperiodique, mais suppose tout de même que l'intervalle de  $0 \leq t_i^0 - t_{i-1}^0 \leq T_e$  est bornée et que la valeur maximale  $T_e$  est connue.

Ainsi, avec cette notation l'effet d'échantillonnage devient invisible. Les auteurs considèrent ensuite un observateur de Luenberger continu d'ordre complet sous la forme suivante :

$$\hat{x}'(t) = A\hat{x}(t) + Bu(t) - L(y(t - \delta(t)) - \hat{y}(t - \delta(t))) \quad (3.4a)$$

$$\hat{y}(t) = C\hat{x}(t). \quad (3.4b)$$

Sachant que la paire  $(A, C)$  est observable, il est possible de déterminer un gain linéaire  $L$  tel que l'état  $\hat{x}(t)$  de l'observateur (4.4) converge exponentiellement vers l'état du processus en l'absence du retard  $x(t)$ . Le théorème qui suit permet de calculer ce gain  $L$  de façon à ce que la convergence exponentielle malgré l'existence du retard variable sur  $\delta(t)$ . L'erreur d'estimation  $e(t) = x(t) - \hat{x}(t)$  est définie par :

$$e'(t) = Ae(t) + LCe(t - \delta(t)) \quad (3.5)$$

**Théorème 3.1.** Soit l'observateur défini par (4.4), on suppose que, pour les réels positifs  $\alpha$  et  $\varepsilon$ , il existe des matrices de dimensions  $n \times n$ ,  $P_1 > 0$ ,  $P$ ,  $S$ ,  $Y_1, Y_2$ ,  $Z_1, Z_2, Z_3$ ,  $R$ ,  $R_a$  et une matrice  $W$  de dimensions appropriées telles que les conditions IML soient satisfaites pour  $j=1,2$  :

$$\begin{bmatrix} \psi & \begin{bmatrix} \beta_1 WC - Y_1^T \\ \varepsilon \beta_1 WC - Y_2^T \end{bmatrix} & \mu \beta_1 \begin{bmatrix} WC \\ \varepsilon WC \end{bmatrix} \\ * & -S & 0 \\ * & * & -\mu R_a \end{bmatrix} < 0 \quad (3.6a)$$

$$\begin{bmatrix} R & Y_1 & Y_2 \\ * & Z_1 & Z_2 \\ * & * & Z_3 \end{bmatrix} < 0 \quad (3.6b)$$

où  $\beta_1 = e^{\alpha(\delta-\mu)}$ ,  $\beta_2 = e^{\alpha(\delta+\mu)}$  et la matrice symétrique  $\psi$  définie par :

$$\psi_{11} = P^T(A + \alpha I) + (A + \alpha I)^T P + S + \delta Z_1 + Y_1 + Y_1^T, \quad (3.7a)$$

$$\psi_{12} = P_1 - P + \varepsilon P^T(A + \alpha I)^T + \delta Z_2 + Y_2, \quad (3.7b)$$

$$\psi_{22} = -\varepsilon(P + P^T) + \delta Z_3 + 2\mu R_a. \quad (3.7c)$$

Avec le retard  $\delta(t)$ ,  $\delta$  et  $u$  étant imposés indépendamment par la période d'échantillonnage et le retard. La valeur de  $\alpha$  correspond à la vitesse de stabilisation. Ainsi, l'objectif est d'ajuster  $\varepsilon$  afin de maximiser  $\alpha$ .

Alors, la convergence exponentielle avec un taux  $\alpha$  de l'erreur (3.5) est assurée par le gain défini comme suit :

$$L = (P^T)^{-1}W . \quad (3.8)$$

En analysant cette méthode de synthèse, on voit clairement que la matrice  $W$  satisfaisant les IML avec des valeurs initiales imposées n'existe pas constamment. D'ailleurs, la valeur qu'on peut trouver pour maximiser sa vitesse de convergence ne satisfait pas toujours les usagers.

### 3.4. OCM, à partir de la sortie retardée et échantillonnée

Le rôle de cet observateur est de reconstituer l'état continu sans retard  $x(t) = [y(t) \quad w(t)]^T$ , à partir de la seule information disponible  $\mathcal{G}_i^k$ , où  $w(t) \in R^{n-m}$  représente la partie reste d'état.

#### 3.4.1. Mise en équation

##### 3.4.1.1. Première étape : utilisation du premier SLCM (I)

En utilisant le SLCM I sur morceau  $q_i = ]iT_e \quad (i+1)T_e]$ , avec  $\varphi(t)=u(t)$  et  $\psi(t)=0$ , nous obtenons :

$$M_{i-1}(t) = \int_{(i-1)T_e}^t \exp A(t-\tau)Bu(\tau)d\tau , \quad (3.9)$$

ce qui conduit avec  $t = iT_e$ , à :

$$M_{i-1}^i = \int_{(i-1)T_e}^{iT_e} \exp A(iT_e - \tau)Bu(\tau)d\tau = \begin{bmatrix} m1_{i-1}^i \\ m2_{i-1}^i \end{bmatrix} \quad (3.10)$$

Le rôle du BOZ est donc de récupérer et de mémoriser cette valeur à l'instant  $iT_e$ .

##### 3.4.1.2. Deuxième étape : application de l'Observateur Discret de Luenberger d'Ordre Réduit (ODLOR)

$x_{i-1}^k$  est estimé par l'ODLOR selon les équations :

$$z_i^k = Fz_{i-1}^k + Gy_{i-1}^k + (m2_{i-1}^k - Lm1_{i-1}^k) \quad (3.11a)$$

$$z_{i-1}^k = \hat{w}_{i-1}^k - Ly_{i-1}^k \quad (3.11b)$$

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

$$F = f_{22} - Lf_{12},$$

$$G = FL + (f_{21} - Lf_{11}),$$

dans lesquelles :  $f_{11} \in R^{m \times m}$ ,  $f_{12} \in R^{m \times (n-m)}$ ,  $f_{21} \in R^{(n-m) \times m}$ ,  $f_{22} \in R^{(n-m) \times (n-m)}$  et la valeur de matrice  $L \in R^{(n-m) \times m}$  est choisit de telle façon que les valeurs propres absolues de matrice  $F$  soient minimales afin que l'ODLOR ait une convergence maximale.

Ainsi, en estimant  $w_{i-1}^k$  par  $\hat{w}_{i-1}^k = z_{i-1}^k + Ly_{i-1}^k$ , on obtient  $\hat{x}_{i-1}^k$ , puis après un BOZ de période  $T_e$ , on a facilement  $\hat{x}_{i-1}^0$ . Avec cette valeur comme condition initiale et par intégration de (3.1a) sur morceaux  $q_i$ , on obtient finalement  $\hat{x}_i^0$  :

$$\hat{x}_i^0 = \exp(AT_e)\hat{x}_{i-1}^0 + M_{i-1}^i \quad (3.12)$$

### 3.4.1.3. Troisième étape : utilisation du deuxième SLCM (II)

En utilisant le SFM II, avec  $\varphi(t)=u(t)$  et  $\psi(t)=\hat{x}_i^0$ , on estime finalement  $x(t)$  par :

$$\hat{x}(t) = \exp(At)\hat{x}_i^0 + \int_{T_e}^t \exp A(t-\tau)Bu(\tau)d\tau \quad (3.13)$$

En se réinitialisant à chaque période d'échantillonnage  $T_e$ , le BOZ par rapport à l'équation (3.10) dans le but de prendre  $\hat{x}_{i-1}^0$ , il peut être éliminé.

L'OCM permet ainsi de compenser les effets de retard et de la discrétisation. Cela conduit à des estimations continues et sans retard,  $\hat{y}(t)$  et  $\hat{w}(t)$ .

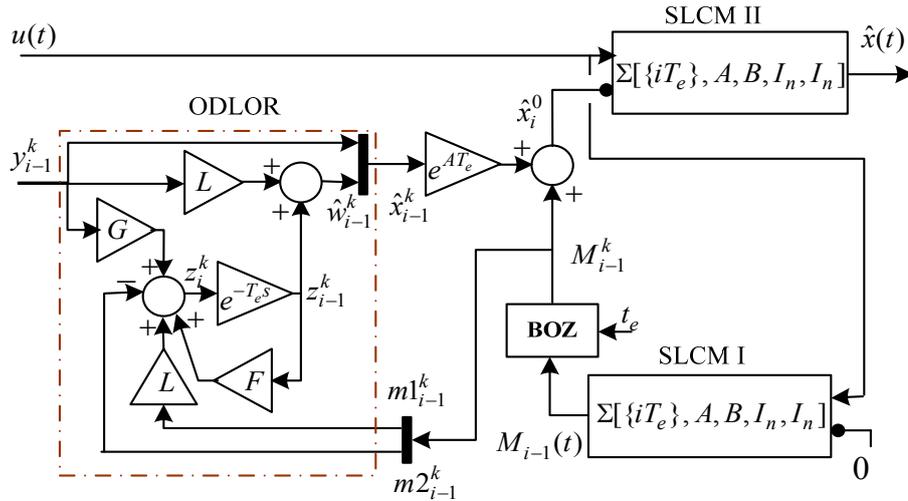
## 3.4.2. Architecture de réalisation

Dans cette partie, on réalise d'abord l'architecture de l'OCM à partir de la sortie retardée et échantillonnée  $y_{i-1}^k$ , ensuite on l'adapte dans les cas où on dispose de la sortie retardée  $y(t-T_e)$ , de l'état retardé et échantillonné  $x_{i-1}^k$  et finalement de l'état retardé  $x(t-T_e)$ .

### 3.4.2.1. OCM à partir de la sortie retardé et échantillonnée $y_{i-1}^k$

La figure 3.2 illustre la réalisation d'un observateur continue par morceaux. Il est clair que ce type d'observateur est composé de deux SLCM et d'un ODLOR.

Concernant le schéma de la Figure 3.2, il est important de souligner que l'on a besoin de placer un retard infiniment petit  $\exp(-\varepsilon s)$  en sortie du SLCM I et à l'entrée du BOZ, pour pouvoir prendre exactement la valeur correspondante de  $M_{i-1}^i$  définie par (3.10). Cette remarque a également la même importance dans les trois cas qui suivent.



**Fig. 3.2.** Observateur continu par morceaux à partir de la sortie retardée et échantillonnée  $y_{i-1}^k$

De plus, pour un schéma plus propre, on peut enlever le BOZ après le SLCM I, parce que le SLCM II se réinitialisant à chaque période de  $T_e$  a la même fonction que BOZ illustré sur la figure. Pour une raison de continuité sur la mise en équation, on le garde pour l'instant dans la figure.

### 3.4.2.2. OCM à partir de la sortie retardée $y(t-T_e)$

De la même façon dans le cas où on dispose la sortie retardée  $y(t-T_e)$ , il suffit de remplacer l'ODLOR dans la figure 3.2 par un Observateur Continu de Luenberger d'ordre Réduit (OCLOR) [Lue 71].  $x(t-T_e)$  est estimé par l'OCLOR selon les équations :

$$z'(t-T_e) = Fz(t-T_e) + Gy(t-T_e) + [m2(t-T_e) - Lm1(t-T_e)] \quad (3.14a)$$

$$\hat{w}(t-T_e) = z(t-T_e) + Ly(t-T_e) \quad (3.14b)$$

où  $F$  &  $G$  sont définis comme dans les équations (3.11) ( $F = f_{22} - Lf_{12}$  et  $G = FL + (f_{21} - Lf_{11})$ ), et  $m2(t-T_e)$  &  $m1(t-T_e)$  sont définis à partir de l'équation suivante :

$$M(t-T_e) = \int_{t-T_e}^t \exp A(t-\tau)Bu(\tau)d\tau = \begin{bmatrix} m1(t-T_e) \\ m2(t-T_e) \end{bmatrix} \quad (3.14c)$$

Par ailleurs  $L \in R^{(n-m) \times m}$  est choisie de telle façon que les valeurs propres de la matrice  $F$  soient négatives pour que l'OCLOR ait une convergence rapide.

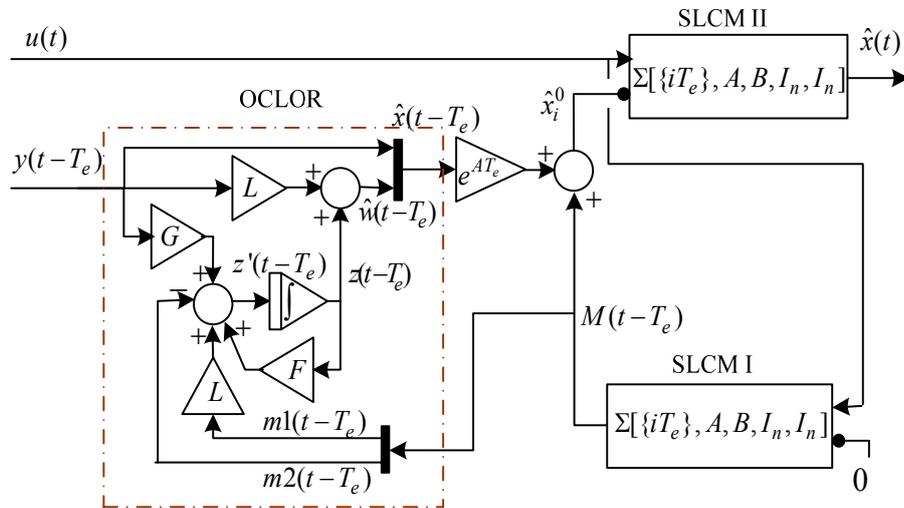


Fig. 3.3. Observateur continu par morceaux à partir de la sortie retardée  $y(t - T_e)$

Ainsi à l'aide des deux SLCM appliqués dans la figure réalisant une fonction de prédiction, on réalise l'architecture d'observation illustrée sur la figure 3.3.

### 3.4.2.3. OCM à partir de l'état retardé et échantillonné $x_{i-1}^k$

Cet observateur peut être utilisé dans le cas où on dispose de l'état retardé et échantillonné  $x_{i-1}^k$ , l'architecture correspondante est illustrée dans la figure 3.4. Il est

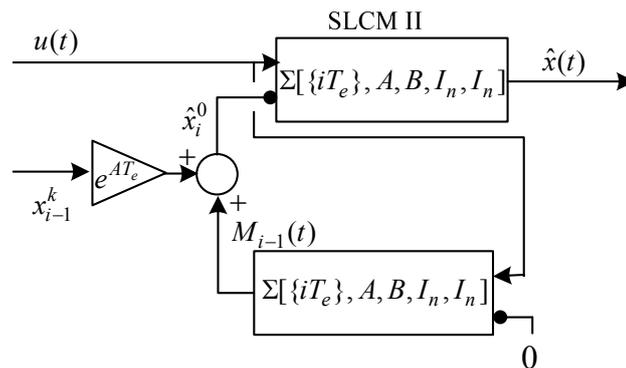


Fig. 3.4. Observateur continu par morceaux à partir de l'état retardé et échantillonné  $x_{i-1}^k$

évident que dans ce cas, on n'a pas besoin d'appliquer un ODLOR ou un OCLOR dans le cas continu comme illustré sur les figures 3.2 et 3.3.

Dans la structure donnée en figure 3.4, on alimente directement par  $x_{i-1}^k$  à la place de  $\hat{x}_{i-1}^k$  dans la figure 3.2. On peut donc conclure que les deux SLCM ont la capacité de prédire et reconstruire l'état continu à l'instant présent  $\hat{x}(t)$ .

#### 3.4.2.4. OCM à partir de l'état retardé $x(t-T_e)$

Dans le cas où on dispose de l'état retardé  $x(t-T_e)$ , le schéma est le même que celui de la figure 3.4 en remplaçant  $x_{i-1}^k$  par  $x(t-T_e)$  (cf. figure 3.5).

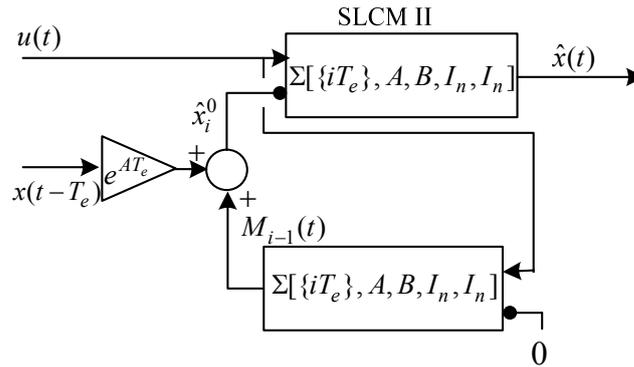


Fig. 3.5. Observateur continu par morceaux à partir de l'état retardé  $x(t-T_e)$

### 3.4.3. Validations expérimentales

#### 3.4.3.1. Sans bruit de mesure

Nous considérons ici le système à estimer  $\Sigma_c(\{t_0\}, A, B, I_2, C)$  défini par :

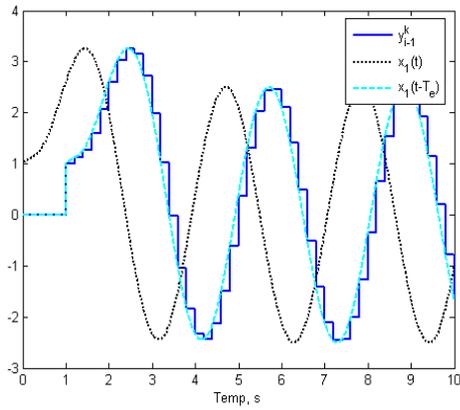
$$A = \begin{bmatrix} -2 & 1 \\ 0 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad C = [1 \quad 0] \quad \text{et} \quad x_0 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

L'entrée est définie par :  $u(t) = 10 \sin(2t)$ .

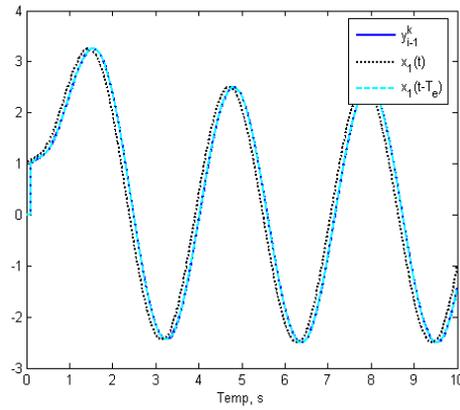
Par ailleurs, nous supposons que  $(T_e = 1 \text{ s}, p=5)$  et  $(T_e = 0.1 \text{ s}, p=20)$ .

Les résultats de simulations correspondants sont illustrés sur la figure 3.6 et la figure 3.7 respectivement.

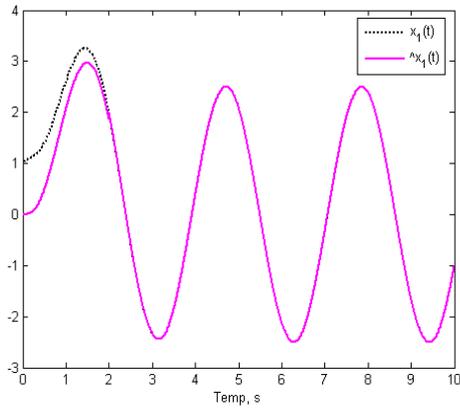
On peut constater que même dans le cas  $(T_e = 1 \text{ s}, p=5)$  avec un grand retard, l'OCM estime l'état avec une précision élevée après une courte transition.



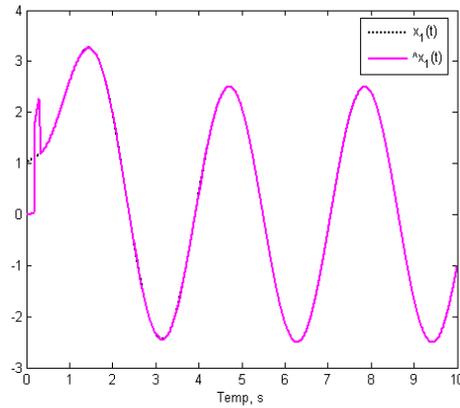
(a). Comparaison entre  $y_{i-1}^k$ ,  $x_1(t)$  et  $x_1(t-T_e)$



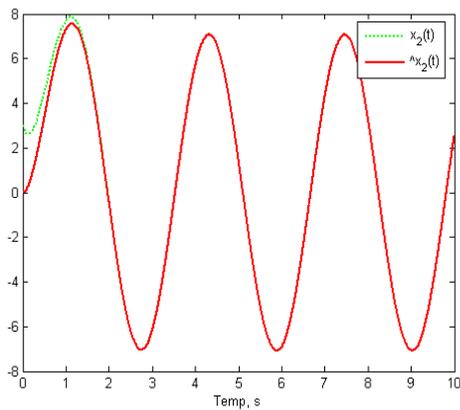
(a). Comparaison entre  $y_{i-1}^k$ ,  $x_1(t)$  et  $x_1(t-T_e)$



(b). Estimation de  $x_1(t)$

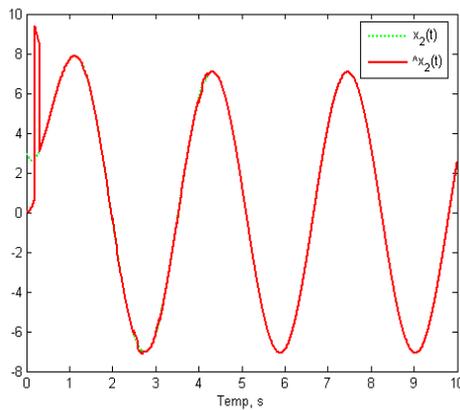


(b). Estimation de  $x_1(t)$



(c). Estimation de  $x_2(t)$

**Fig. 3.6.**  $T_e=1s$  et  $p=5$  ( $t_e=0.2s$ )



(c). Estimation de  $x_2(t)$

**Fig. 3.7.**  $T_e=0.1s$  et  $p=20$  ( $t_e=0.05s$ )

### 3.4.3.2. Avec bruit de mesure (bruit blanc)

Nous considérons ici le même processus et la condition initiale de §3.4.3.1. Nous ajoutons un bruit de mesure sous la forme de bruit blanc avec une puissance de bruit [0.01] représenté dans la figure 3.8.

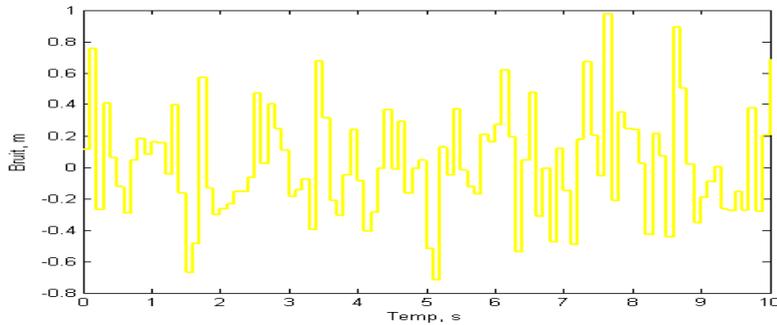
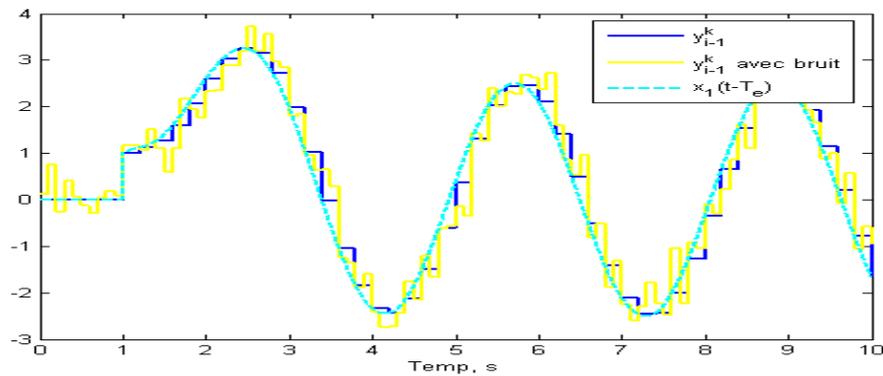
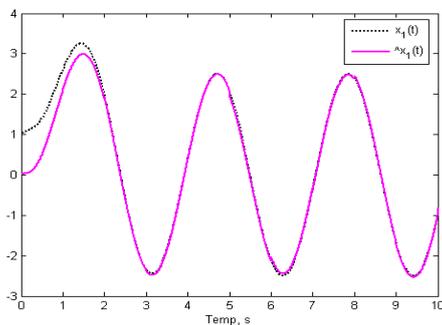


Fig. 3.8. Bruit blanc de mesure avec une puissance de [0.01]

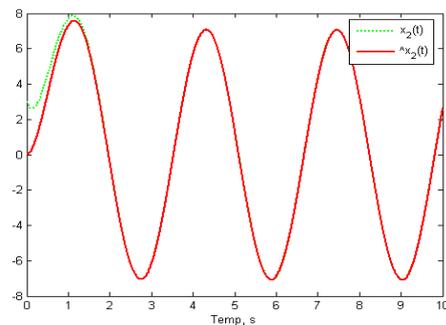
Ainsi les résultats de simulation correspondant sont montrés sur la figure 3.9. On peut conclure que l'OCM a une fonction de filtrage sur les mesures.



(a). Comparaison entre  $y_{i-1}^k$ ,  $y_{i-1}^k$  avec bruit et  $x_1(t - T_e)$



(b). Estimation de  $x_1(t)$

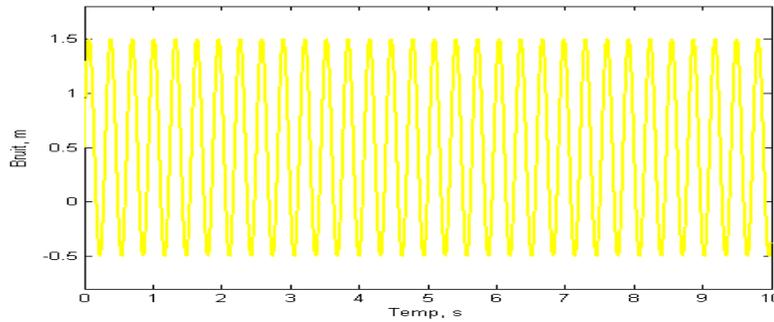


(c). Estimation de  $x_2(t)$

Fig. 3.9. Bruit blanc de mesure avec une puissance de [0.01] avec ( $T_e=1s$  et  $p=5$ )

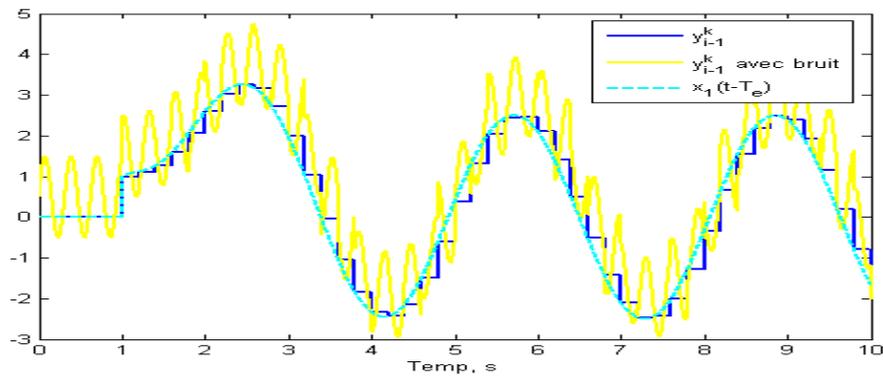
**3.4.3.3. Avec bruit de mesure ( $a_1 \sin(bt) + a_0$ )**

Avec le même processus et la condition initiale de §3.4.3.1, nous considérons ici un bruit de mesure sous la forme de  $a_1 \sin(bt) + a_0$ . Nous choisissons  $a_1 = 1$ ,  $b = 20$  et  $a_0 = 0.5$ . Ce bruit est représenté dans la figure 3.10.

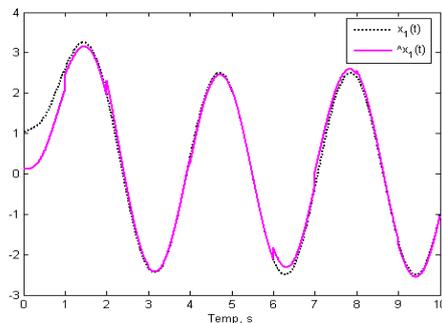


**Fig. 3.10.** Bruit de  $\sin(20t) + 0.5$

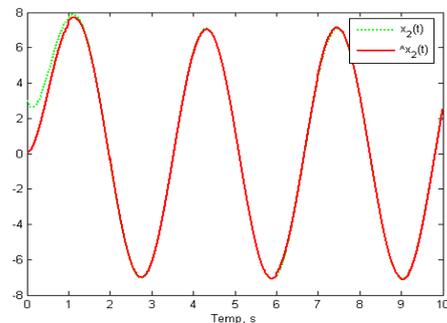
Les résultats de simulation correspondant sont montrés sur la figure 3.11. On peut conclure que l'OCM a une fonction de filtrage sur les mesures.



(a). Comparaison entre  $y_{i-1}^k$ ,  $y_{i-1}^k$  avec bruit et  $x_1(t - T_e)$



(b). Estimation de  $x_1(t)$



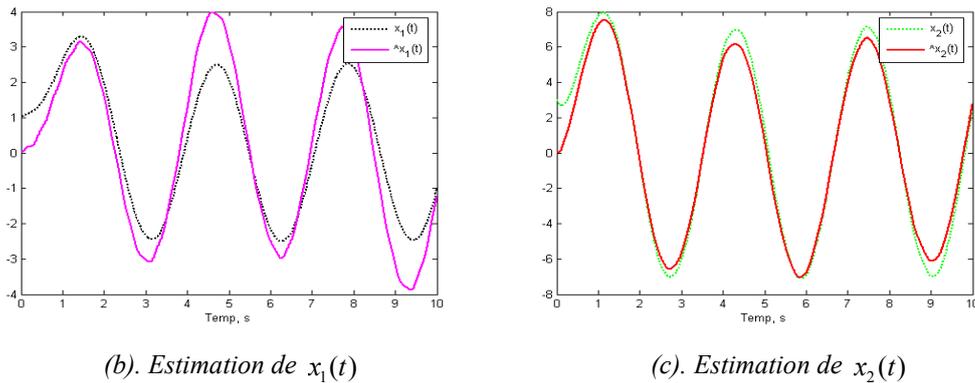
(c). Estimation de  $x_2(t)$

**Fig. 3.11.** Bruit de  $\sin(20t) + 0.5$  avec ( $T_e = 1s$  et  $p = 5$ )

### 3.4.3.4. Comparaison avec l'observateur de Seuret [Seu 06]

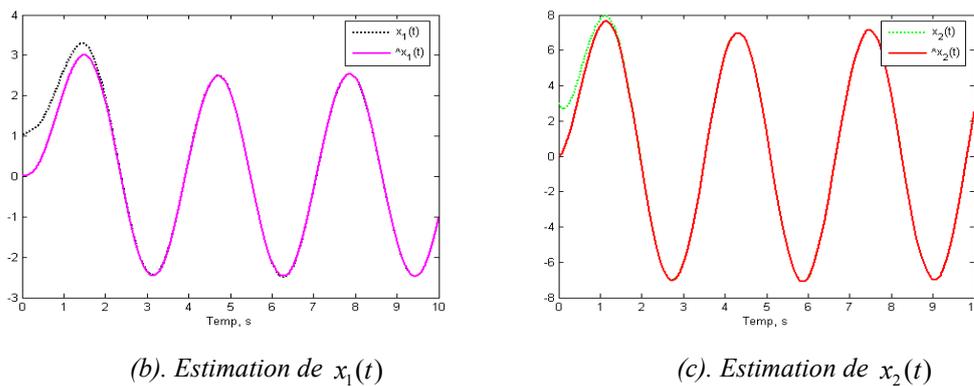
Dans ce cas, avec le même processus et la condition initiale de §3.4.3.1 incluant un bruit de mesure  $\sin(20t)+0.5$  illustré sur la figure 3.10, nous ajoutons une entrée inconnue représentée sur la figure 3.8 : un bruit blanc de puissance  $[0.01]$ .

Dans le premier cas, nous utilisons la méthode de [Seu 06] en appliquant le théorème 3.1 avec  $\alpha = 2$  (obtenu pour  $\varepsilon = 3.43$ ), le gain d'observateur est calculé :  $L = [-0.9372 \quad -0.4686]^T$ . Sous ces conditions les résultats de simulation sont présentés figure 3.12.



**Fig. 3.12.** Méthode de [seu 06] avec ( $T_e=2s$  et  $p=2$ )

La figure 3.13 illustre les résultats de simulation avec la même condition que pour la figure 3.12 en utilisant l'OCM.



**Fig. 3.13.** Méthode de l'OCM avec ( $T_e=2s$  et  $p=2$ )

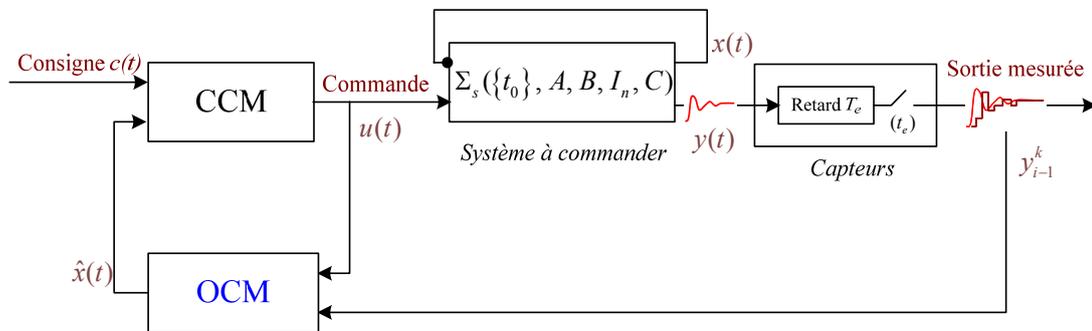
En comparant les deux Observateurs, il est évident que les performances de l'OCM sont meilleures par rapport à [Seu 06]. L'OCM fonctionne de la façon optimale et sa synthèse est plus simple et efficace. Malgré la possibilité d'augmenter la valeur de  $\alpha$  pour améliorer la performance de l'observateur de [Seu 06], le problème de l'inexistence permanente de la matrice  $W$  satisfaisant les conditions d'IML (3.6) demeure.

### 3.4.4. Conclusion

L'Observateur Continu par Morceaux développé dans ce chapitre composé d'un ODLOR et de deux SLCM semble robuste et utilisable dans différents environnements de travail (avec un bruit blanc, périodique ou constant). De plus, l'observateur proposé est simple à réaliser sans contraintes à respecter.

## 3.5. Adaptation de l'OCM à la poursuite à partir de la sortie retardée et échantillonnée

La figure 3.12 illustre l'architecture de poursuite à partir d'une mesure retardée et



**Fig. 3.12.** Adaptation OCM dans la poursuite à partir de la sortie retardée et échantillonnée

échantillonnée  $y_{i-1}^k$ . Le OCM considéré est bien évidemment positionné au centre du système global de commande.

En fonction des différents retours disponibles :  $y_{i-1}^k$ ,  $y(t-T_e)$ ,  $x_{i-1}^k$  ou  $x(t-T_e)$ , nous choisissons l'OCM approprié d'après les structures (de figure 3.2 – 3.5) pour reconstruire  $\hat{x}(t)$ .

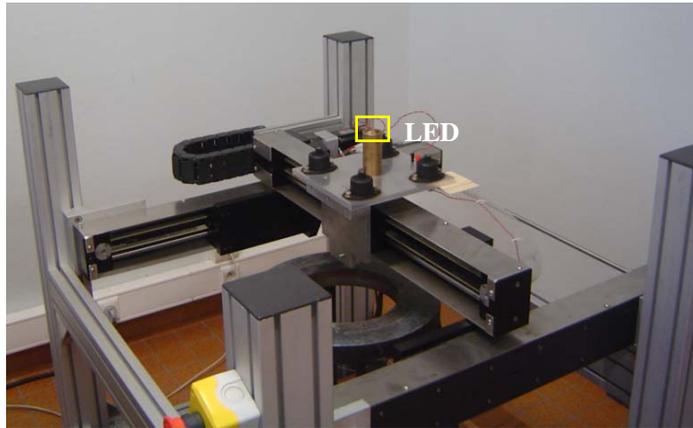
Une fois que l'état est estimé  $x(t)$ , on peut implanter facilement nos contrôleurs développés dans le chapitre 2.

Dans le cas Bi-échantillonné, les méthodes de poursuite bi-échantillonnée proposées en fonction des  $x_{i-1}^k$  ou  $y_{i-1}^k$  sont présentées dans [Wan 06, Wan 06a, Wan 07]<sup>3</sup>.

<sup>3</sup> Dans le but de la simplicité et la cohérence de la présentation, on ne les présente pas ici.

### 3.6. Application à la poursuite d'un chariot x-y motorisé observé par vision artificielle

Contrairement à la section §.2.5.4.3, nous introduisons ici une commande de poursuite en intégrant l'OCM et CR développé précédemment. Le chariot à retour visuel est montré sur la figure 3.13.



**Fig. 3.13.** Chariot x-y à retour visuel

Le système possède une structure symétrique sur les deux axes et le mouvement de la partie mobile du chariot x-y est produit par la combinaison des mouvements des deux axes. Pour les raisons de la simplicité, on présente ici seulement la commande selon l'axe x (cf. Annexe I).

#### 3.6.1. Formulation du problème

Le système est modélisé par un  $\Sigma_s(\{t_0\}, A, B, I_2, C)$  défini comme suit :

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau} \end{bmatrix}, B = [0 \quad k/\tau]^T \text{ et } C = [1 \quad 0].$$

$k$  et  $\tau$  sont respectivement le gain et la constante de temps de l'ensemble amplificateur-moteur-chariot.  $u(t)$  est l'entrée du servomoteur et  $x(t)$  vecteur d'état continu composé respectivement de la position et de la vitesse du chariot ( $x = [x_c \quad x'_c]^T$ ).

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

$$\mathcal{G}^*(t) = (x_c^*(t - T_e), y_c^*(t - T_e)) \tag{3.15}$$

où \* représente un échantillonnage de période  $t_e$  (durée d'une prise d'image) et  $T_e = pt_e$  (avec  $p$  entier) correspond au temps nécessaire pour le traitement d'image et pour la transmission des signaux.

Dans le cas particulier de notre système de vision, c-à-d dans le cas où  $p=1$ ,  $T_e = t_e$ , on note simplement  $t_i^k$  et  $\mathcal{G}_i^k = y_{i-1}^k$  comme  $t_k$  ( $t_k = kt_e$ ) et  $\mathcal{G}_k = y_{k-1}$ . A chaque instant d'échantillonnage  $kt_e$ , l'équation (3.15) fournit  $(x_{C,k-1}, y_{C,k-1})$ . Ainsi on note :  $y_{k-1} = x_{C,k-1}$ .

### 3.6.2. Adaptation de l'OCM dans le cas où $p=1$

Dans ce cas, nous réalisons un observateur continu par morceaux à partir de la sortie retardée et échantillonnée  $y_{k-1}$  avec  $p=1$ . En comparant avec la figure 3.2 où  $p \neq 1$ , nous trouvons que l'architecture de réalisation plus simple.

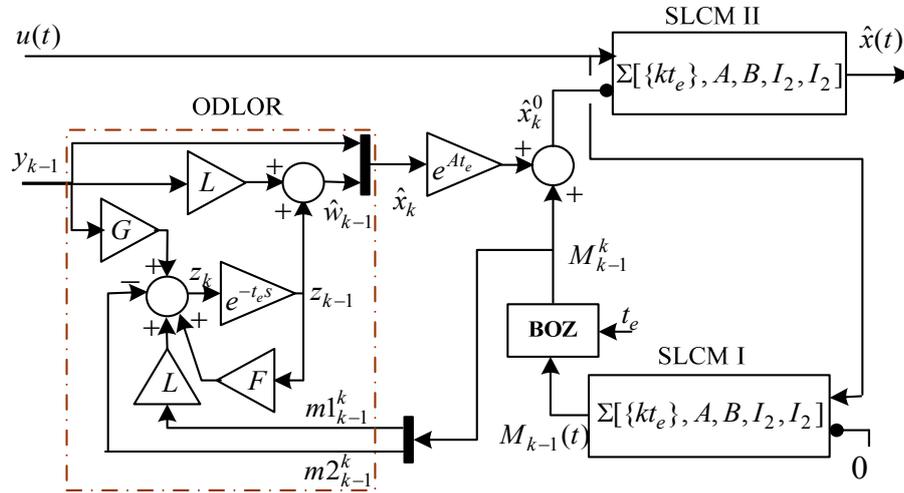


Fig. 3.14 OCM dans le cas  $p=1$  à partir de la sortie retardée et échantillonnée  $y_{k-1}$

L'ODLOR est défini par les équations suivantes :

$$z_k = Fz_{k-1} + Gy_{k-1} + (m2_{k-1}^k - Lm1_{k-1}^k) \quad (3.16a)$$

$$\hat{w}_{k-1} = z_{k-1} + Ly_{k-1} \quad (3.16b)$$

dans lesquelles  $m1_{k-1}^k$  et  $m2_{k-1}^k$  sont définies par :

$$M_{k-1}^k = \int_{(k-1)t_e}^{kt_e} \exp A(it_e - \tau) Bu(\tau) d\tau = \begin{bmatrix} m1_{k-1}^k \\ m2_{k-1}^k \end{bmatrix} \quad (3.16c)$$

D'ailleurs, les périodes de commutation dans les deux SLCM sont  $t_e$ . Ils correspondent à la période d'échantillonnage de capteur numérique. Il faut souligner que dans le système de vision réel, il est important de synchroniser les instants de commutation des deux SLCM avec la cadence des prises d'image de la caméra dans le but de minimiser l'erreur d'estimation.

### 3.6.3. Architecture de commande

Suivant l'architecture de la figure 3.12, nous réalisons ainsi une poursuite à partir de la sortie retardée et échantillonnée en remplaçant l'OCM par celui illustré dans la figure 3.14 et le CCM par le CR avec amortisseur  $\Xi(\varepsilon, \theta)$  présenté dans la figure 2.16 du Chapitre 2.

### 3.6.4. Validation expérimentale

Dans notre cas,  $t_e = 40$  ms et  $p = 1$  et nous définissons une consigne de sortie pour chaque axe (l'axe x et l'axe y) comme suit :

$$\begin{bmatrix} c_{s,x} \\ c_{s,y} \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} m_x \sin \omega_x t \\ m_y \sin(\omega_y t + \varphi) \end{bmatrix}$$

dans lesquels  $\alpha$  (rad) est angle oblique,  $m_x$  &  $m_y$  (m) sont amplitudes pour chaque axe, et  $\omega_x$  &  $\omega_y$  (rad/sec) sont les fréquences différentes imposées pour chaque axe. D'ailleurs,  $\varphi$  (rad) est le déphasage.

Pour valider notre méthode, nous imposons les valeurs dans le CR et l'amortisseur :  $I_m^- = 0.92$ ,  $\gamma = 1$  et  $\Xi(2.4, 0)$  selon chaque axe (x ou y).

Nous notons  $(x_C(t), y_C(t))$  les positions continues du chariot x-y mesurées par les codeurs,  $(x_{C,k-1}, y_{C,k-1})$  les positions mesurées par le système de vision et  $(u_x(t), u_y(t))$  les commandes d'entrées de chaque axe.

Les résultats d'expérimentations correspondantes avec  $(\alpha, m_x, m_y, \omega_x, \omega_y, \varphi) = (\pi/3, 0.04, 0.06, 2, 4, \pi/2)$  sont illustrés dans la figure 3.15. Cette figure fait apparaître : (a) les déplacements  $x_C(t)$ ,  $\hat{x}_C(t)$  et  $x_{C,k-1}$  selon axe x, (b) les déplacements  $y_C(t)$ ,  $\hat{y}_C(t)$  et  $y_{C,k-1}$  selon axe y, (c) & (d) zoom correspondant de (a) & (b) dans l'intervalle de  $[0.5 \ 2]$ , (e) les vitesses  $(\dot{x}_C(t), \hat{\dot{x}}_C(t))$  selon axe x, (f) les vitesses  $(\dot{y}_C(t), \hat{\dot{y}}_C(t))$  selon axe y, (g) la comparaison entre les consignes  $(c_{s,x}(t), c_{s,y}(t))$  et les positions  $(x_C(t), y_C(t))$  et finalement (h) la commande d'axes  $(u_x(t), u_y(t))$ . Après lecture des figures, on peut conclure que la méthode proposée en réalisant une commande de poursuite à partir de la sortie visuelle  $(x_{C,k-1}, y_{C,k-1})$  est effective et avec précision élevée.

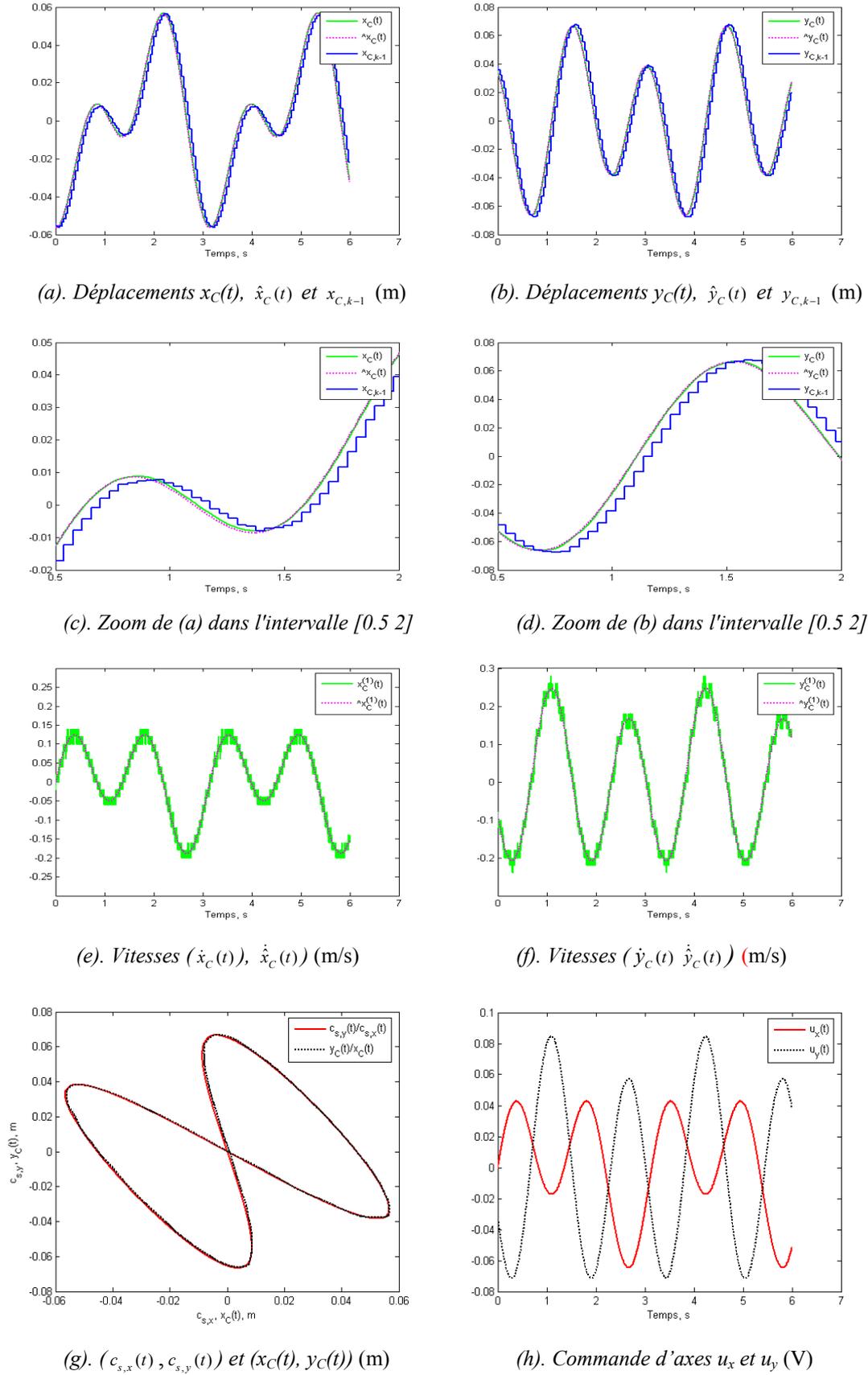
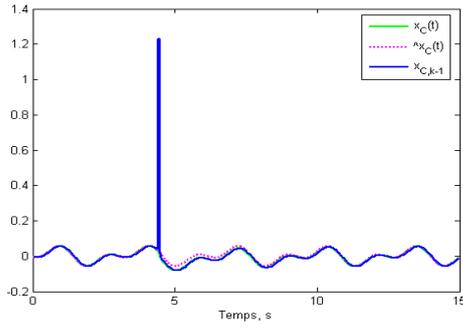
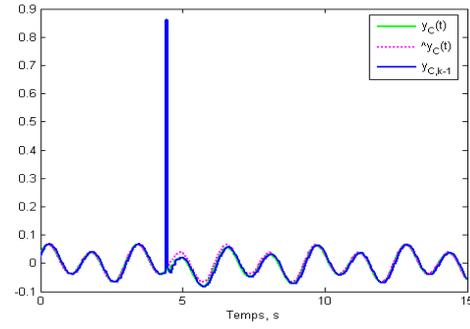


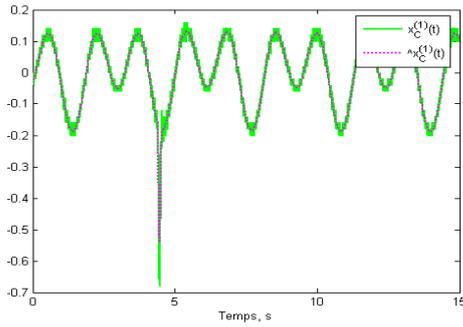
Fig. 3.15. Pursuite du processus réel d'un chariot x-y à retour visuel



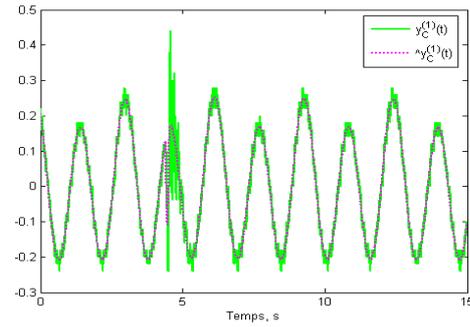
(a). Déplacements  $x_C(t)$ ,  $\hat{x}_C(t)$  et  $x_{C,k-1}$  (m)



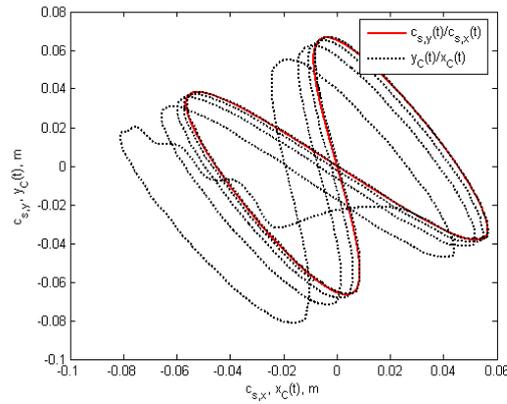
(b). Déplacements  $y_C(t)$ ,  $\hat{y}_C(t)$  et  $y_{C,k-1}$  (m)



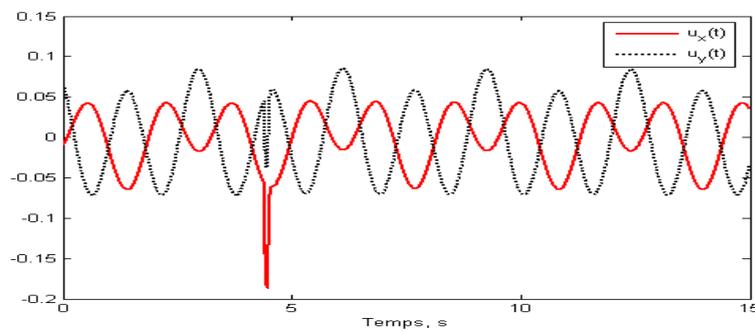
(c). Vitesses ( $\dot{x}_C(t)$ ,  $\hat{\dot{x}}_C(t)$ ) (m/s)



(d). Vitesses ( $\dot{y}_C(t)$ ,  $\hat{\dot{y}}_C(t)$ ) (m/s)



(e). ( $c_{s,x}(t)$ ,  $c_{s,y}(t)$ ) et ( $x_C(t)$ ,  $y_C(t)$ ) (m)

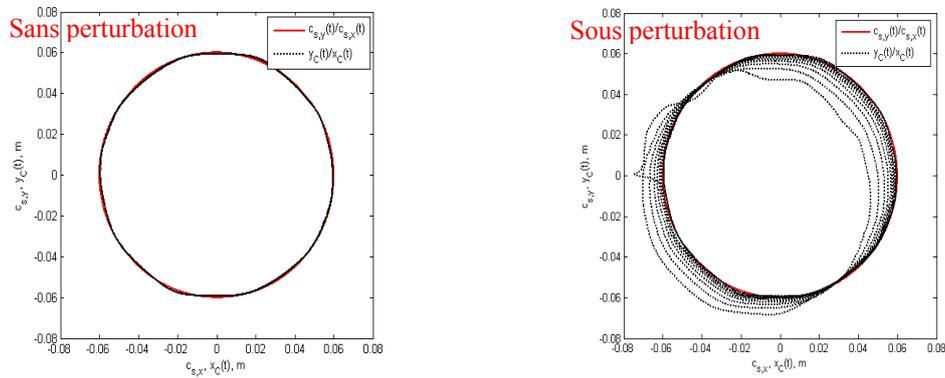


(f). Commande d'axes  $u_x$  et  $u_y$  (V)

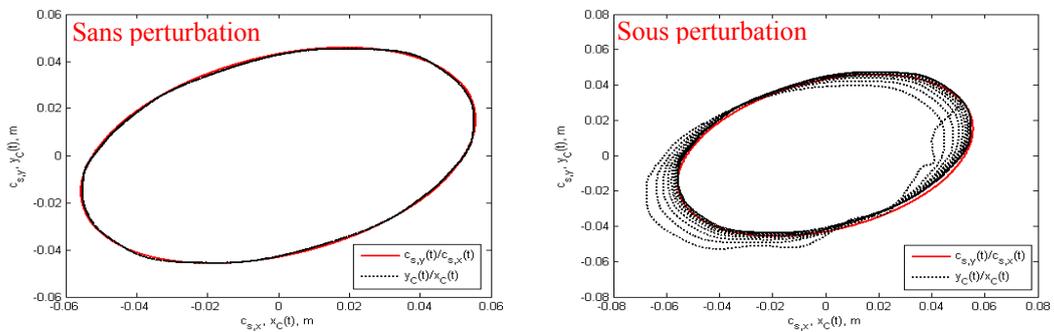
**Fig. 3.16.** poursuite du processus réel d'un chariot  $x$ - $y$  à retour visuel sous perturbation

Dans le but de faire une étude sur la robustesse de notre méthode, nous simulons une situation où l'on rend le système de vision aveugle (en masquant la LED par un livre, etc). Les résultats d'expérimentation correspondante sont illustrés dans la figure 3.16. On constate que même si les informations visuelles de retour saturent au moment où l'on perturbe, la commande reste effective et avec une amplitude réduite grâce à l'utilisation d'un amortisseur avec le contrôleur de CR.

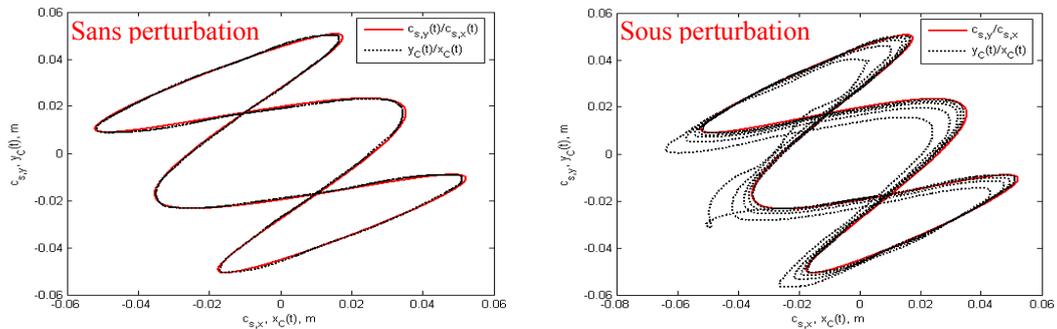
D'ailleurs nous changeons les différentes consignes, mais avec le même paramétrage de contrôleur CR et amortisseur, un cercle, une ellipse, une figure de Lissajous. Les résultats expérimentaux sont illustrés respectivement sur les figures 3.17, 3.18 et 3.19.



**Fig. 3.17.** Pursuite d'un cercle  $(\alpha, m_x, m_y, \omega_x, \omega_y, \varphi) = (\pi/3, 0.06, 0.06, 4, 4, \pi/2)$



**Fig. 3.18.** Pursuite d'une ellipse  $(\alpha, m_x, m_y, \omega_x, \omega_y, \varphi) = (\pi/6, 0.06, 0.04, 5, 5, \pi/2)$



**Fig. 3.19.** Pursuite d'une courbe de Lissajous  $(\alpha, m_x, m_y, \omega_x, \omega_y, \varphi) = (2\pi/3, 0.04, 0.04, 2.5, 7.5, \pi/2)$

Les vidéos illustratives de ces expériences avec les mêmes consignes (figures 3.17 – 3.19) sont disponibles sur le site :

<http://www-lagis.univ-lille1.fr/~wang/Research.html>

### **3.7. Conclusion**

Nous avons proposé dans ce chapitre un Observateur Continues par Morceaux qui est basé sur la théorie des SFM et l'Observateur de Luenberger Discret en Ordre Réduit (OLDOR). Ce type de l'observateur est bien adapté à des systèmes linéaires invariants avec une sortie retardée et échantillonnée  $y_{i-1}^k$ .

Nous avons également validé l'architecture de poursuite en se basant sur les observations issues d'un OCM appliqué au banc d'essai de chariot x-y à retour visuel. D'après les résultats expérimentaux, nous pouvons conclure que la méthode proposée est simple mais en même temps robuste et performante.

Enfin, il est important de souligner que l'approche OCM utilisée dans le cadre des systèmes linéaires peut être adaptée à des systèmes non linéaires ayant une dynamique élevée à déphasage non minimal.

## Chapitre 4

# Commande d'un pendule inversé 2D/3D par asservissement visuel

*Ce dernier chapitre débute par une introduction générale des difficultés de commande concernant notre maquette : pendule inversé 2D à l'asservissement visuel. Il s'agit d'un système mécanique non stable et sous actionné, avec une forte non linéarité et un retour de type hybride au sens temporel : continu et échantillonné (et retardé). Une analyse des commandes existantes des systèmes similaires a été réalisée. Cette étude nous permet de justifier l'approche proposée. Pour commander le pendule inversé par l'asservissement visuel, deux étapes sont mises en évidence :*

- *Une commande de stabilisation d'un pendule inversé 2D en utilisant deux boucles de retours : la première est une boucle appelée « boucle d'observation du pendule » qui traite les informations échantillonnées et retardées délivrées par le système de vision. Ce traitement vise à réaliser un modèle stable et linéaire du pendule sans aucune approximation et à en reconstruire l'état continu à l'aide d'un observateur spécifique appelé Observateur Continu par Morceaux (OCM). La seconde est une « boucle de stabilisation », visant à stabiliser l'ensemble chariot-pendule utilisant une fonction de Lyapunov. Les simulations et les essais sur plate forme réelle illustrent les performances de cette méthode.*
- *Ensuite, à chaque instant d'échantillonnage du système de vision, la commande agit dans le plan du pendule, ce qui permet de transformer le problème 3D en un problème 2D.*

## 4.1. Introduction

En raison de sa dynamique instable et fortement non linéaire, le système chariot-pendule inversé constitue un excellent support expérimental pour tester des algorithmes de commande. Les modèles les plus souvent rencontrés sont : le bras simple ou double [Olf 99], les pendules planaires simple, double, voire triple relié à un chariot par des liaisons rotoïdes [Bug 03, Che 97, Del 07, Loz 00, Ohu 95, Wei 95] et enfin, le pendule inversé relié à un robot x-y par un joint de cardan [Cho 03, Cho 03a, Jun 04, Yan, 00]. Dans tous ces cas, les contrôleurs utilisent des codeurs angulaires continus de précision élevée fixés directement sur l'axe de rotation du pendule.

Cependant, dans de nombreuses applications il peut s'avérer impossible de mettre le capteur en contact direct avec l'objet commandé. Les systèmes de vision artificielle permettent 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.

Dans cette partie de thèse, l'enjeu est d'utiliser une caméra CCD à bas coût comme capteur immatériel pour la commande d'un pendule inverse 3D (un pendule inverse sur un chariot x-y), dont les caractéristiques et le fonctionnement sont données en Annexe I.

Le système considéré à commander est un système mécanique sous actionné MIMO ayant une dynamique élevée à déphasage non minimal. L'utilisation d'un tel système visuel avec une cadence de la caméra lente (25 images/sec) ajoute une grande difficulté à la commande. Parce qu'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 [Hut 96, Chau 07], 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 » [San 04], soit simplement utilisé pour réaliser des visualisations et/ou des animations dans les systèmes de commande [San 04, Arm 05].

Plus récemment, des tentatives d'insertion de la vision dans la boucle de commande pour stabiliser le système chariot-pendule ont été présentées [Wen 00, Esp 06]. Toutefois, aucune réalisation convaincante de commandes simultanées de l'angle du pendule et de la position du chariot n'apparaît. Seul un contrôleur flou a été développé par [Mag 98], 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 [Fuk 06] 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  $\pm 26^\circ$ .

En analysant des méthodes existantes liées à la commande du pendule inversé par rétroaction visuelle fait apparaître que le problème reste toujours ouvert. C'est la raison pour laquelle, nous nous sommes efforcés de développer les méthodes de commande, afin de pallier les difficultés induites par la vision (échantillonnage et retard). Les méthodes que nous avons développées sont basées sur l'utilisation des SFM et OCM présenté dans les chapitres précédents. L'utilisation de OCM a l'avantage de compenser les effets de retard et d'échantillonnage pour fournir des estimations continues et sans retard.

Pour réaliser la commande d'un pendule 3D illustré par figure 4.1, nous avons réalisé d'abord la commande sur le pendule inverse 2D (chariot-pendule) en utilisant deux

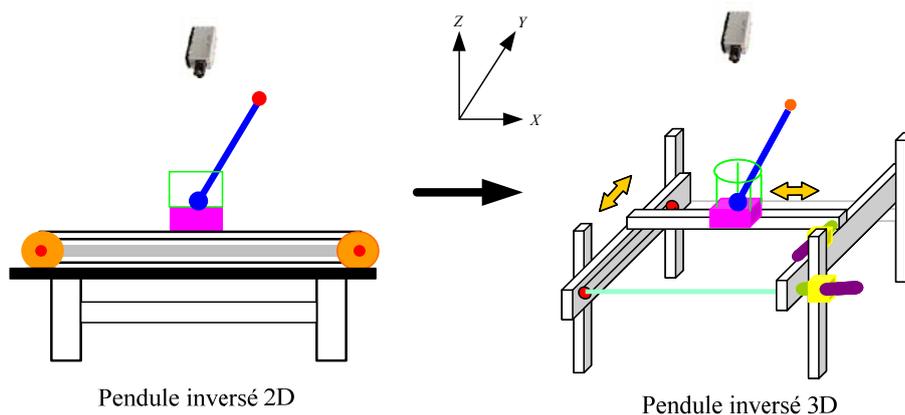


Fig. 4.1. Système à traiter

boucles de retours. La première est une boucle interne d'observation traitant les informations retardée et échantillonnées délivrées par le système de vision. Le traitement vise à réaliser un modèle stable et linéaire du pendule sans aucune approximation et à en reconstruire l'état continu à l'aide d'un OCM. La seconde est une boucle externe de stabilisation visant à stabiliser l'ensemble chariot-pendule selon une commande utilisant une fonction de Lyapunov. Les travaux correspondants sont présentés dans [Wan 08a, Wan 08d, Wan 08f]

Pour le problème 3D, à chaque instant d'échantillonnage de la vision, nous envisageons de choisir un plan de stabilisation, ensuite dans ce plan, il nous amène dans le problème 2D. Ensuite on peut l'appliquer le même principe de commande en deux boucles. Les travaux correspondants sont présentés dans [Wan 08b, Wan 08d, Wan 08f].

## 4.2. Commande d'un pendule inversé 2D

### 4.2.1. Formulation du problème

En se référant à l'annexe I.2, le pendule inversé 2D (un pendule inversé sur un chariot) à retour visuel comme présenté figure 4.2, peut être modélisé comme suit :

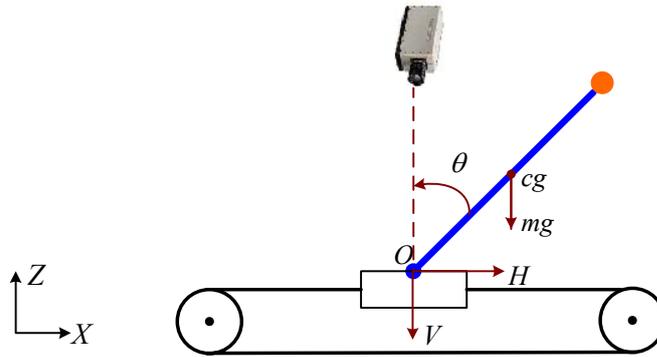


Fig. 4.2. Le pendule inversé 2D dans le plan x-z

$$\ddot{x} = (-\dot{x} + k_x u_x) / \tau_x \quad (4.1a)$$

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} - \omega_n^2 \sin \theta = -K \cos \theta \ddot{x} \quad (4.1b)$$

$$\theta = \sin^{-1}[(x_c - x) / 2l] \quad (4.1c)$$

Avec :  $\omega_n = \sqrt{mgl / (J + ml^2)}$  fréquence naturel du pendule inversé,  $\zeta = B_r / [2\omega_n(J + ml^2)]$  le coefficient d'amortissement,  $K = ml / (J + ml^2)$  le gain du système, dans lesquels :

- $k_x$  et  $\tau_x$  sont respectivement le gain statique et la constante de temps du chariot motorisé selon l'axe x et  $u_x$  sa tension d'entrée.
- $\theta$  angle du pendule avec l'axe verticale,  $l$  longueur du  $1/2$  pendule,  $m$  masse du pendule,  $B_r$  frottement visqueux entre le pendule et le chariot,  $g$  accélération de la pesanteur,  $J = (ml^2) / 3$  moment d'inertie du pendule et  $x$  position du chariot sur l'axe x.
- $x_c$  coordonnée du haut du pendule obtenue par le système de vision. Et  $x$  coordonnée du pied du pendule mesurée par le codeur.

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

$$x_C(kt_e - pt_e) = x_{C,k-p} \quad (4.1d)$$

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

Dans notre système visuel, on dispose  $t_e = 40$  ms et  $p = 1$ , de telle sorte que, à chaque instant d'échantillonnage  $kt_e$ , les équations (4.1c) et (4.1d) fournissent  $\theta_{k-1}$ .

### 4.2.2. Architecture de commande

L'architecture de commande du pendule inversé 2D est présentée figure 4.3. Les retours

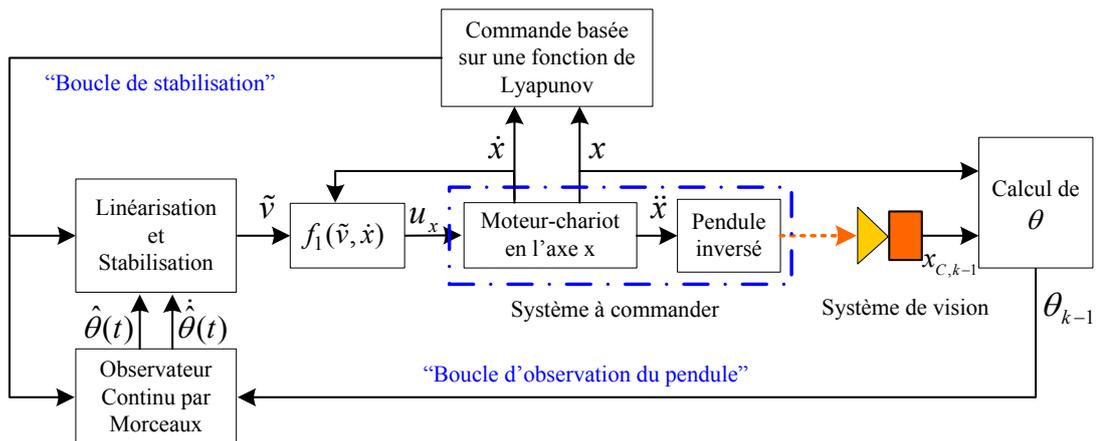


Fig. 4.3. Architecture de commande 2D

fournissent le coordonnée continue  $x$  par le codeur et la position angulaire retardée et échantillonnée  $\theta_{k-1}$  calculé par le système de vision.

La commande de stabilisation sur l'ensemble système : chariot et pendule est réalisée en considérant deux boucles. La première est appelée « boucle d'observation du pendule ». Elle traite les informations échantillonnées et retardées délivrées par le système de vision  $x_{C,k-1}$ . Le traitement vise à réaliser un modèle stable et linéaire du pendule inversé sans aucune approximation et à en reconstruire l'état continu  $[\theta(t) \quad \dot{\theta}(t)]^T$  à l'aide d'un OCM. 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.

Enfin, nous avons validé notre méthode en simulation et en expérimentation temps réel. Il est sûrement valable pour la commande des systèmes avec les retours continus.

### 4.2.3. Boucle d'observation du pendule inversé

#### 4.2.3.1. Linéarisation et stabilisation du pendule inversé

En imposant une commande  $\tilde{v} = (-\dot{x} + k_x u_x) / \tau_x$  aux équations (4.1a) et (4.1b), nous pouvons écrire les équations du système suivant :

$$\ddot{x} = \tilde{v}, \quad (4.2a)$$

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} - \omega_n^2 \sin \theta = -K \cos \theta \tilde{v} \quad (4.2b)$$

avec la relation entre la commande  $u_x$  et  $\tilde{v}$  :

$$u_x = (\tau_x \tilde{v} + \dot{x}) / k_x \quad (4.2c)$$

Afin de stabiliser ce système, une dynamique stable peut être imposée par une nouvelle commande  $v$  avec un nouveau gain  $K_2$ , une nouvelle fréquence naturelle  $\Omega_n$  et l'amortissement  $Z$ . L'équation correspondante qui est vérifié le critère de Routh est défini comme suit :

$$-2\zeta\omega_n\dot{\theta} + \omega_n^2 \sin \theta - K \cos \theta \tilde{v} = -2Z\Omega_n\dot{\theta} - \Omega_n^2 \theta + K_2 v \quad (4.3)$$

ce qui conduit à :

$$\tilde{v} = \left[ -2(\zeta\omega_n - Z\Omega_n)\dot{\theta} + \Omega_n^2 \theta + \omega_n^2 \sin \theta - K_2 v \right] / K \cos \theta, \quad |\theta| < \pi/2 \quad (4.4)$$

Après transformation, le modèle (4.2) devient :

$$\ddot{x} = \left[ -2(\zeta\omega_n - Z\Omega_n)\dot{\theta} + \Omega_n^2 \theta + \omega_n^2 \sin \theta - K_2 v \right] / K \cos \theta \quad (4.5a)$$

$$\ddot{\theta} + 2Z\Omega_n\dot{\theta} + \Omega_n^2 \theta = K_2 v \quad (4.5b)$$

En adoptant pour (4.5b) une représentation d'état, il vient :

$$\dot{\Theta}(t) = A\Theta(t) + Bv(t) \quad (4.6)$$

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

#### 4.2.3.2. Observateur continu par morceaux

Dans la réalité, 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  $\hat{\dot{\theta}}(t)$  de  $\theta(t)$  et  $\dot{\theta}(t)$ , nous utilisons ci-après un OCM réalisant la combinaison d'un Observateur Discret de Luenberger d'Ordre Réduit (ODLOR) et de deux SLCM qui est premièrement présenté dans le chapitre 3.

Le fonctionnement d'un SLCM  $\Sigma[\{kt_e\}, A, B, C, I_2]$  dans le morceau  $q_k = ]kt_e, (k+1)t_e]$ , avec deux entrées : une entrée discrétisée  $\psi(t)$  (ce qu'on peut utiliser pour satisfaire les conditions initiales aux instants d'échantillonnage) et une autre entrée continue  $\varphi(t)$  est défini comme suit :

$$\begin{aligned} x_k(t) &= \exp(A(t - kt_e))\psi(kt_e) + \int_{kt_e}^t \exp A(t - \tau)B\varphi(\tau)d\tau, \forall t \in q_k \\ y_k(t) &= Cx_k(t) \quad \forall t \end{aligned} \quad (4.7)$$

Dans le but d'estimer  $\Theta(t)$ , nous choisissons :  $A$  et  $B$  comme définies dans (4.6) et  $C=I_2$ . Ainsi l'OCM en application sur le pendule inversé 2D présenté figure 4.3 est construit en trois étapes :

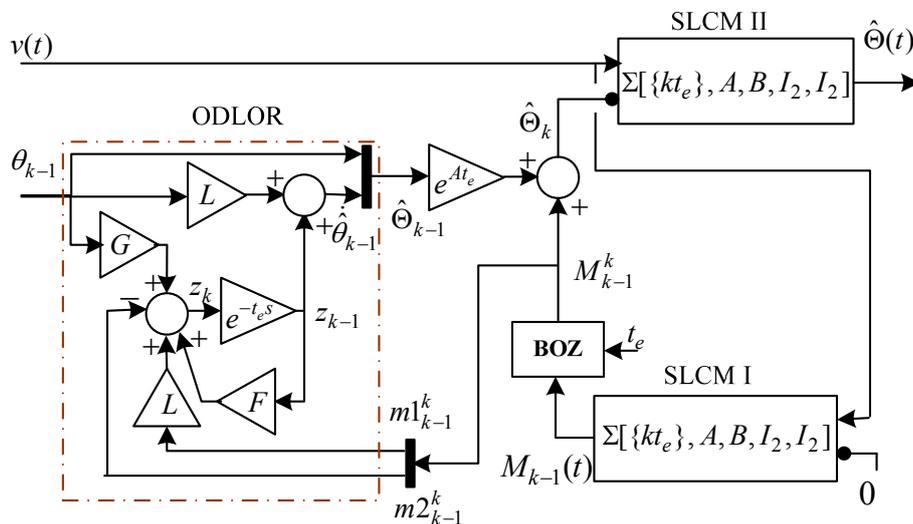


Fig. 4.3. OCM en pendule inverse 2D

- Première étape :

En utilisant le SLCM I, avec  $\varphi(t)=v(t)$  et  $\psi(t)=0$ , nous obtenons :

$$M_{k-1}(t) = \int_{(k-1)t_e}^t \exp A(t-\tau)Bv(\tau)d\tau, \quad (4.8)$$

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

$$M_{k-1}^k = \int_{(k-1)t_e}^{kt_e} \exp A(kt_e - \tau)Bv(\tau)d\tau = \begin{bmatrix} m1_{k-1}^k \\ m2_{k-1}^k \end{bmatrix} \quad (4.9)$$

Le rôle du BOZ est donc de récupérer et de mémoriser cette valeur à  $kt_e$ .

▪ *Deuxième étape :*

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

$$z_k = Fz_{k-1} + G\theta_{k-1} + (m2_{k-1}^k - Lm1_{k-1}^k) \quad (4.10a)$$

$$z_{k-1} = \hat{\theta}_{k-1} - L\theta_{k-1} \quad (4.10b)$$

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

$$F = f_{22} - Lf_{12},$$

$$G = FL + (f_{21} - Lf_{11}),$$

$$L = (f_{22}/f_{12}) \in R \text{ (Valeur imposée pour maximiser la convergence d'ODLOR).}$$

Ainsi, en estimant  $\dot{\theta}_{k-1}$  par  $\hat{\theta}_{k-1} = z_{k-1} + L\theta_{k-1}$ , on obtient  $\hat{\Theta}_{k-1}$ , puis, par intégration de (4.6) sur morceaux  $q_k$ , on obtient  $\hat{\Theta}_k$  :

$$\hat{\Theta}_k = \exp(At_e)\hat{\Theta}_{k-1} + M_{k-1}^k \quad (4.11)$$

▪ *Troisième étape :*

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

$$\hat{\Theta}(t) = \exp(At)\hat{\Theta}_k + \int_{kt_e}^t \exp A(t-\tau)Bv(\tau)d\tau \quad (4.12)$$

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  $\hat{\dot{\theta}}(t)$ .

#### 4.2.4. Boucle de stabilisation

Dans un premier temps, en appliquant aux équations (4.5a) et (4.5b) l'hypothèse de quasi-stationnarité [Isi 94, Gue 05] (cf. développement détaillé en Annexe II) c'est-à-dire  $\ddot{\theta}=\dot{\theta}=0$ , il est possible de définir pour  $\theta$  une trajectoire de référence notée  $\theta = \theta_{ref}$ , satisfaisant les équations suivantes :

$$\ddot{x} = \tilde{v} = \omega_n^2 \tan(\theta_{ref}) / K \quad (4.13a)$$

$$v = \Omega_n^2 \theta_{ref} / K_2 \quad (4.13b)$$

dans lesquels la valeur de  $\theta_{ref}$  doit être ajustée par chaque pas de calcul afin de stabiliser l'ensemble du système : chariot-pendule.

Dans un second temps, nous définissons la fonction de Lyapunov ci-après :

$$V(x, \dot{x}) = (\chi x^2 + \delta \dot{x}^2) / 2, \text{ avec : } \chi > 0 \text{ et } \delta > 0 \quad (4.14)$$

ayant pour dérivée :

$$\dot{V}(x, \dot{x}) = \dot{x}(\chi x + \delta \ddot{x}) = \dot{x}(\chi x + \delta \tilde{v}), \text{ d'après l'équation (4.13a).}$$

Pour garantir la stabilisation du chariot, nous imposons cette dérivée négative, par la relation :

$$(\chi x + \delta \tilde{v}) = -\mu [1 - e^{-(\chi x^2 + \delta \dot{x}^2)/2}] \dot{x}, \quad \mu > 0 \quad (4.15)$$

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

$$\tilde{v} = -[\chi x + \mu(1 - e^{-(\chi x^2 + \delta \dot{x}^2)/2}) \dot{x}] / \delta \quad (4.16)$$

Finalement, en reportant l'équation (5.16) dans l'équation (5.13a), nous obtenons l'expression de  $\theta_{ref}$ , en fonction de  $x$  et  $\dot{x}$  :

$$\theta_{ref}(x, \dot{x}) = \tan^{-1} \left\{ -K [\chi x + \mu(1 - e^{-(\chi x^2 + \delta \dot{x}^2)/2}) \dot{x}] / (\delta \omega_n^2) \right\} \quad (4.17)$$

Par construction d'une fonction de Lyapunov à dérivée négative, nous savons que la commande (4.16) stabilise le chariot ( $x = \dot{x} = 0$ ). Dès lors, d'après (4.17), il apparaît que la stabilisation du chariot conduit à  $\theta_{ref}(0, 0) = 0$  et donc à  $v=0$ , d'après l'équation (4.13b).

Or d'après l'équation (4.5b),  $v=0$  permet d'obtenir  $\theta = \dot{\theta} = 0$ . L'introduction de  $\theta_{ref}$ , tel que défini par les équations (18) et (19), assure donc la stabilité de l'ensemble chariot-pendule.

Finalement, en utilisant l'équation (5.13b), nous obtenons  $v$  :

$$v = \Omega_n^2 \tan^{-1} \left\{ -K[\chi x + \mu(1 - e^{-(\chi x^2 + \delta x^2)/2})\dot{x}] / (\delta \omega_n^2) \right\} / K_2 . \quad (4.18)$$

Ainsi nous bouclons la commande présentée dans l'architecture de figure 4.3.

## 4.2.5. Validations expérimentales

Pour le système réel de la figure I.1 dans l'annexe I, le pendule en laiton a une masse de 560 g, une longueur de 32,4 cm. Le frottement visqueux entre le pendule et le chariot est estimé par  $B_r = 0.001$ , et l'accélération de la pesanteur  $g = 9.81 \text{ m/s}^2$ , Cela donne :  $\zeta = 0.0038$  et  $\omega_n = 6.739 \text{ rad/s}$ . Les paramètres du système moteur-chariot sont identifiés par la méthode proposée dans [Cha 05] :  $k_x = 2.92$  et  $\tau_x = 0.008 \text{ s}$ , selon l'axes  $x$ .

Enfin les valeurs des différents paramètres introduits précédemment dans la commande sont :  $K_2 = \omega_n^2$ ,  $\xi = 1.2$ ,  $\Omega_n = \omega_n$ ,  $\chi = 2.5$ ,  $\delta = 10$  et  $\mu = 8$ .

### 4.2.5.1. Résultats de simulation

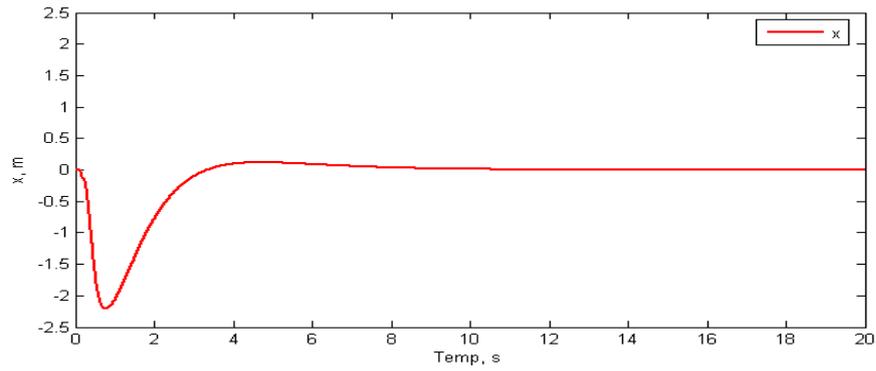
#### 4.2.5.1.1. Sans perturbation

La figure 4.4 illustre la stabilisation avec une condition initiale ( $x_0 = 0.5 \text{ m}$ ,  $\theta_0 = \pi/3 \text{ rad}$ ,  $\dot{x}_0 = 0 \text{ m/s}$ ,  $\dot{\theta}_0 = 0 \text{ rad/s}$ ).

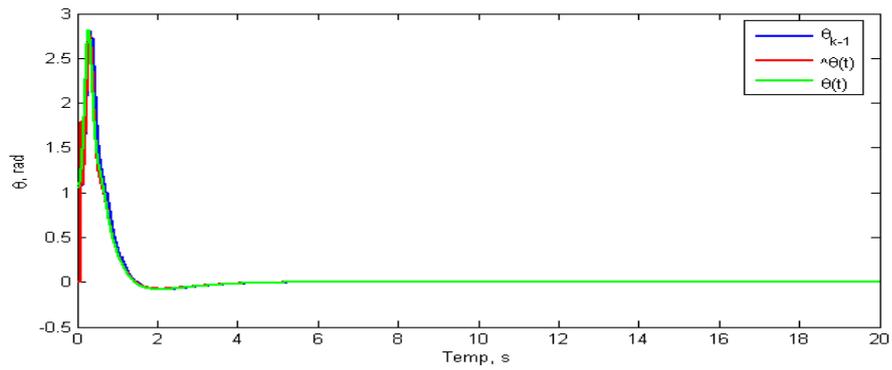
Cette figure fait apparaître : (a) le déplacement  $x$  selon axe  $x$ , (b) les positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ ), (c) les vitesses angulaires du pendule ( $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$ ) et finalement (d) la commande d'axe  $u_x$ .

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 chariot, ni sur le pendule. Les résultats de simulation illustrent parfaitement que  $\theta_{ref}$  assure la stabilité du système global.

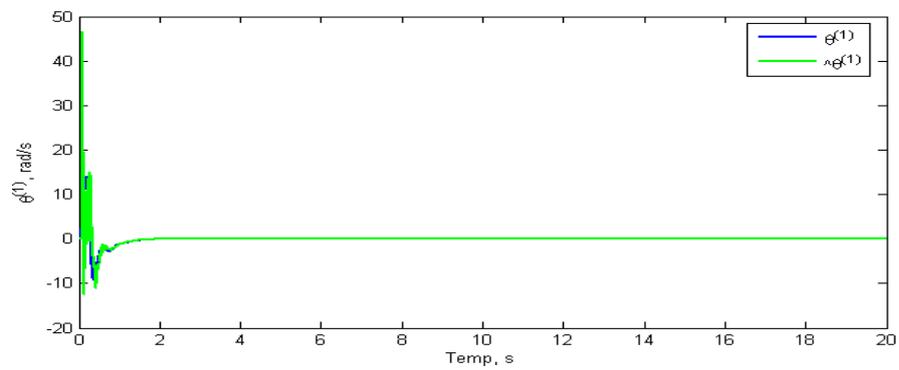
Pour faciliter la lecture des performances de l'OCM, nous avons fait un zoom des figures 3.4b et 3.4c, sur l'intervalle de temps [0.2s, 2.5s]. Les résultats correspondants apparaissent figure 4.5. Après un transitoire très court, obtenu en faisant  $F=0$ , l'OCM donne d'excellentes estimations de  $\theta(t)$  et  $\dot{\theta}(t)$ .



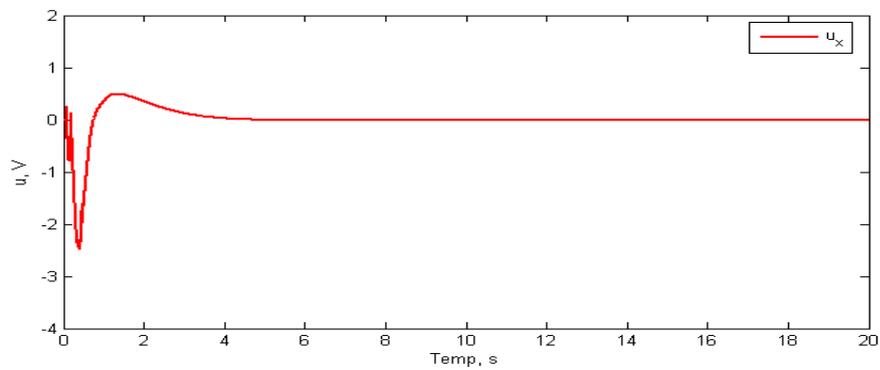
(a). Déplacement  $x$  selon axe  $x$



(b). Positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ )

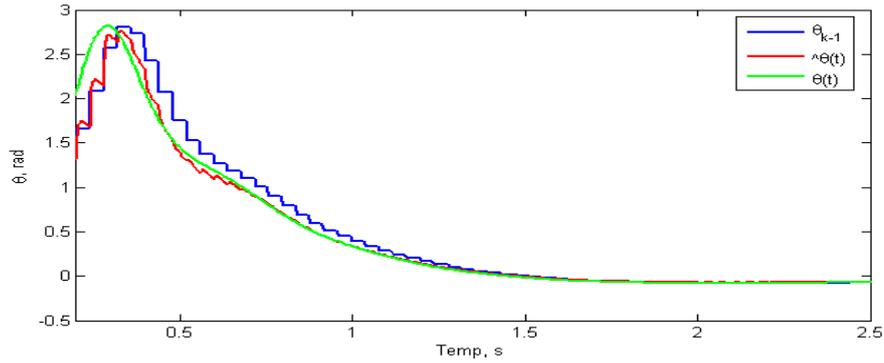


(c). Vitesses angulaires du pendule ( $\hat{\theta}(t)$  et  $\dot{\hat{\theta}}(t)$ )

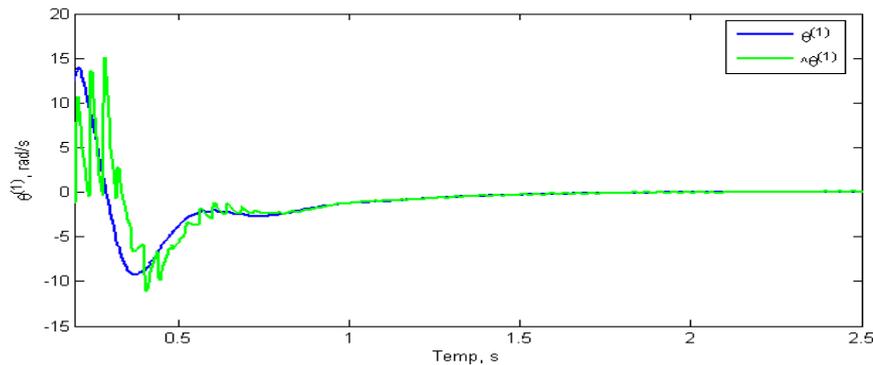


(d). Commande d'axe  $u_x$

**Fig. 4.4.** Stabilisation d'un pendule inversé 2D sans perturbations



(a) Positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ )



(b) Vitesses angulaires du pendule ( $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$ )

**Fig. 4.5.** Zoom sur OCM de la figure 4.4

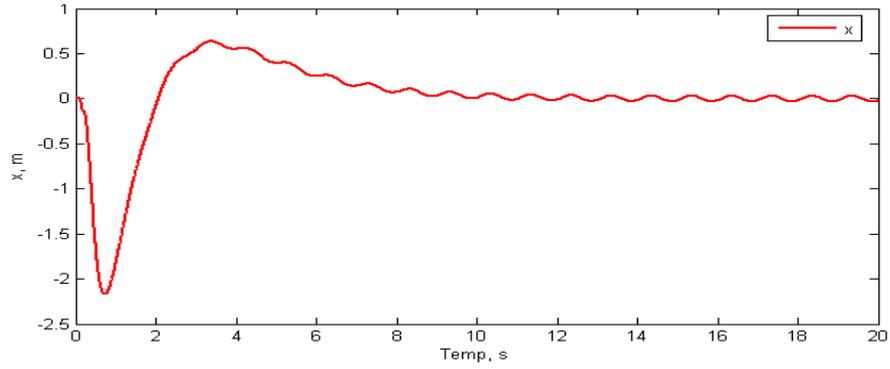
#### 4.2.5.1.2. Avec perturbations

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.

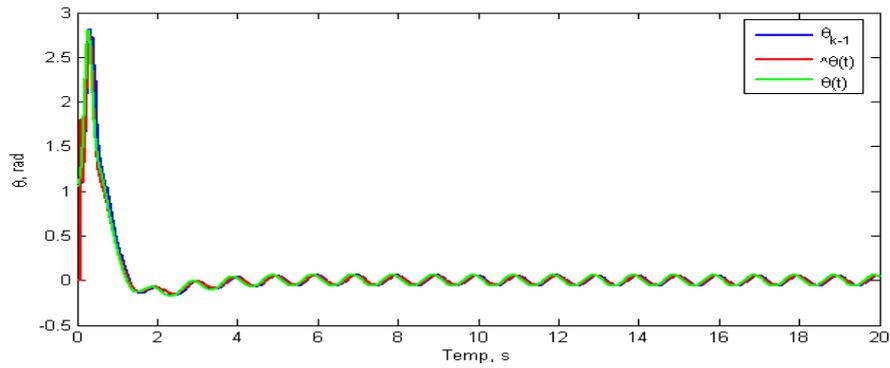
Dans ce sens, pour étudier la robustesse de la stratégie de commande proposée, nous avons ajouté simultanément deux perturbations :

- un terme de variation sur le paramètre  $B_r$  de la forme  $10B_r \sin 2\pi t$
- une autre erreur de mesure sur  $\theta_{k-1}$  de la forme  $0.02 \sin 2\pi t$ , ce qui représente une erreur de mesure environs 1.2 degré. Dans l'asservissement visuel cette somme d'erreur est importante.

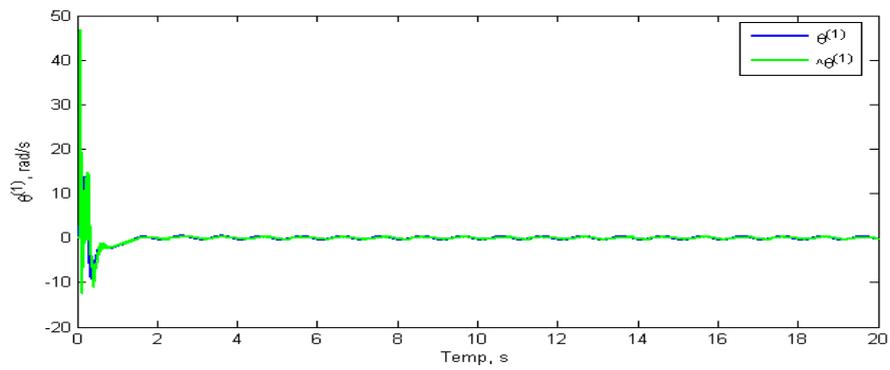
La simulation sous la même condition initiale que figure 4.4 correspondante apparaît figure 4.6. Elle montre que, même dans ce cas, le système chariot-pendule reste stable. L'apparition des oscillations sur les sorties est tout à fait raisonnable. L'OCM donne encore des estimations satisfaisantes illustrées figure 4.7 (zoom de la figure 4.6 dans l'intervalle de temps [0.2s, 2.5s]).



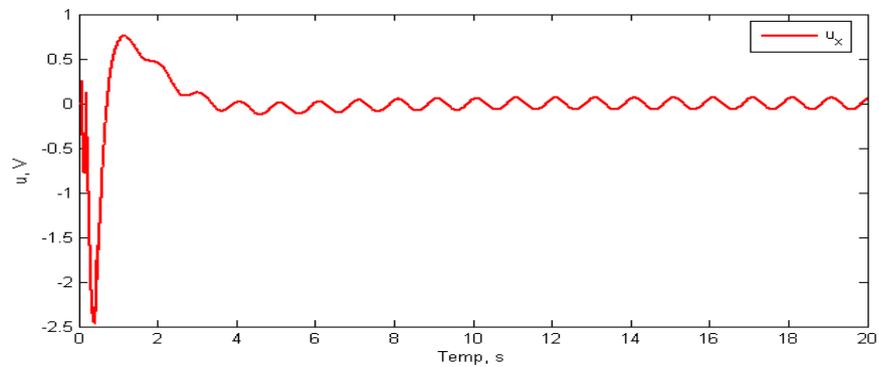
(a). Déplacement  $x$  selon axe  $x$



(b). Positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ )

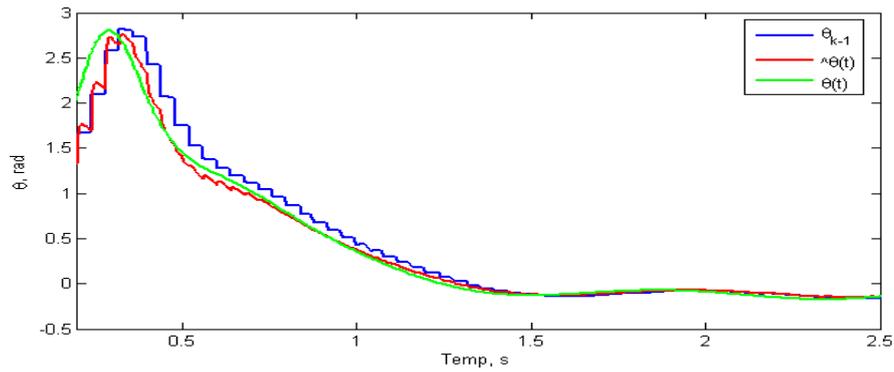


(c). Vitesses angulaires du pendule ( $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$ )

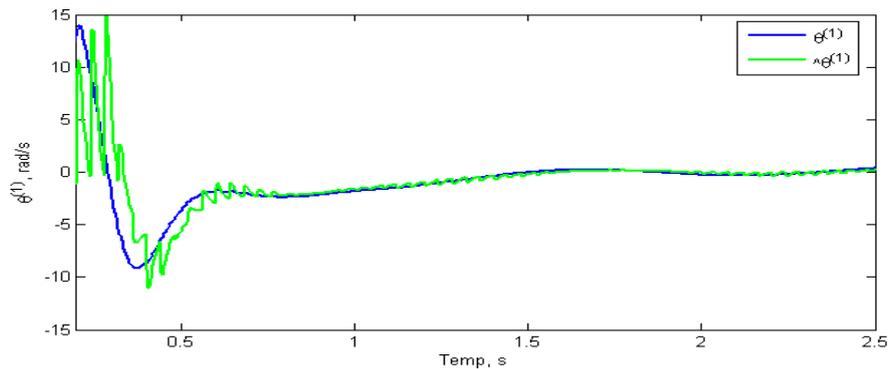


(d). Commande d'axe  $u_x$

**Fig. 4.6.** Stabilisation d'un pendule inversé 2D avec perturbations



(a). Positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ )



(b). Vitesses angulaires du pendule ( $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$ )

**Fig. 4.7.** Zoom sur OCM de la figure 4.6

#### 4.2.5.2. Application au système réel

En raison de contraintes du système mécanique (le support de moteur, la zone autorisée de déplacement etc), la méthode proposée ne peut être implantée directement pour stabiliser le pendule inversé avec une condition initiale de  $\theta_0 = 0.873$  rad (environ 50 degrés), où le pendule repose sur la couronne circulaire (cf. figure I.1 dans l'annexe I).

##### 4.2.5.2.1. Algorithme de démarrage

Pour pallier des difficultés indiquées au dessus, nous avons développé un nouveau algorithme de démarrage, ce qui nous permet d'amener le pendule au voisinage de  $\theta = \dot{\theta} = 0$ . Ensuite la stratégie de stabilisation définie précédemment prend le relais. Ainsi pour commander le système réel, la commande se fait en deux étapes.

Pour la première étape, nous ne pouvons pas appliquer une commande d'énergie, telle que présentée dans la plupart des articles [Ohs 95, Ast 00, Bug 03, Chat 02, Fie 99] et qui utiliserait l'effet des oscillations provoquées au démarrage, pour commuter la commande au voisinage de zéro (moins de 10 degrés). En effet, sur notre système, la couronne circulaire introduit une contrainte supplémentaire de limitation de l'angle à environ 50

degrés. Ainsi, nous ne pouvons pas utiliser l'effet des oscillations, qui conduirait à des collisions avec la couronne.

Nous avons donc développé un nouvel algorithme de démarrage en nous inspirant d'une observation de la vie courante. Considérons un passager adossé au siège d'une voiture. Si on accélère fortement sur un intervalle ( $t \in [0, t_1]$ ), cesse d'accélérer sur ( $t \in ]t_1, t_2]$ ) et finalement freine fortement sur ( $t \in ]t_2, T]$ ), alors le corps du passager est d'abord plaqué sur le dossier (phase d'accélération), puis s'incline vers l'avant (au freinage). Dans notre cas, le chariot joue le rôle de la voiture, le pendule celui de passager et la couronne celui de dossier du siège. Dès lors, par un choix approprié des paramètres  $T$ ,  $t_1$  et  $t_2$ , on parvient à ce que l'angle du pendule passe au voisinage de zéro avec une vitesse faible, pour  $x(T) = 0$  et  $\dot{x}(T) = 0$ , avec une condition initiale  $x(0) \neq 0$ . (réf Annexe III)

Cette stratégie de démarrage a été implantée avec succès dans l'expérimentation réelle.

La seconde étape ne pose aucun problème particulier.

#### 4.2.5.2.2. Résultats temps réel

Dans ce cas nous choisissons  $\alpha = 1$ ,  $T = 0.14$  s,  $t_1 = t_2 = 0.07$  s and  $x_0 = 0.05$  cm pour l'algorithme de démarrage. Et pour la commande de stabilisation, nous imposons :  $K_2 = \omega_n^2$ ,  $\xi = 1.2$ ,  $\Omega_n = \omega_n$ ,  $\chi = 100$ ,  $\delta = 10$  et  $\mu = 4$ .

La figure 4.8 illustre la stabilisation de l'ensemble chariot-pendule avec condition initiale de : (a) stabilisation du chariot, (b) stabilisation du pendule, faisant apparaître  $\theta_{k-1}$  et  $\hat{\theta}(t)$ , (c) commande  $u_x$  et (d) dérivé de la fonction Lyapunov choisi.

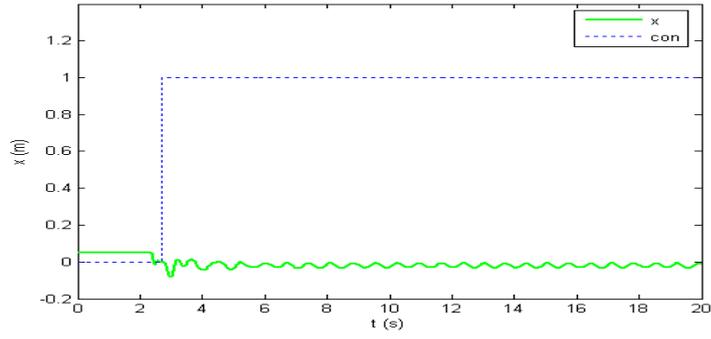
On constate qu'il subsiste des oscillations résiduelles limitées à  $\pm 3^\circ$  pour le pendule et à  $\pm 2$  cm pour le déplacement du chariot. Les essais ont montré que l'ensemble chariot-pendule reste stable plus de 10 heures (nous n'avons pas testé au-delà).

La figure 4.9 illustre le comportement du dispositif perturbé manuellement à partir de 4s. La perturbation consiste à essayer de déséquilibrer le pendule en agissant sur son extrémité. On constate que le chariot se déplace de manière à assurer l'équilibre. Dès que la perturbation cesse (à 9s), l'algorithme maintient l'équilibre du pendule, tout en ramenant le chariot à l'origine.

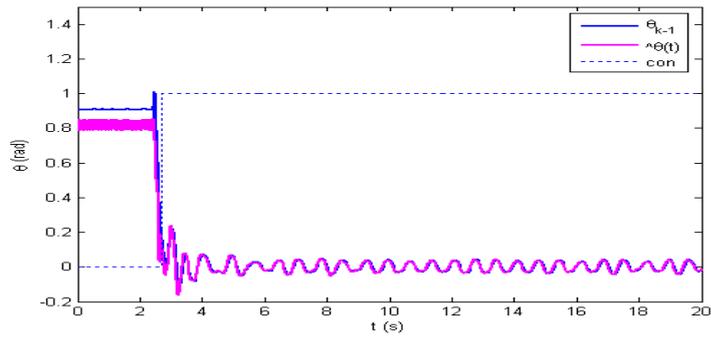
Des vidéos illustratives de ces expériences sont disponibles sur le site :

<http://www-lagis.univ-lille1.fr/~wang/Research.html>

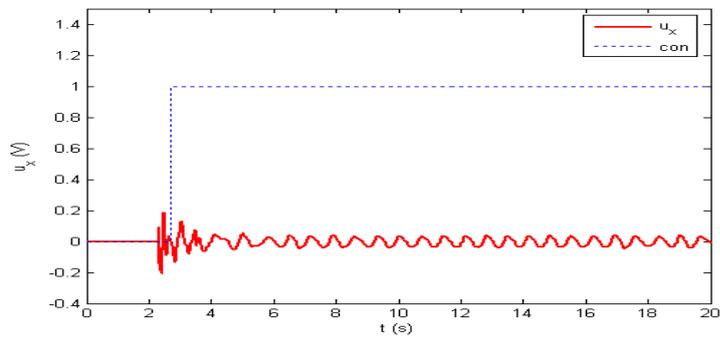
on y constatera également que lorsque l'on rend le système aveugle (en masquant la LED), l'équilibre est rompu.



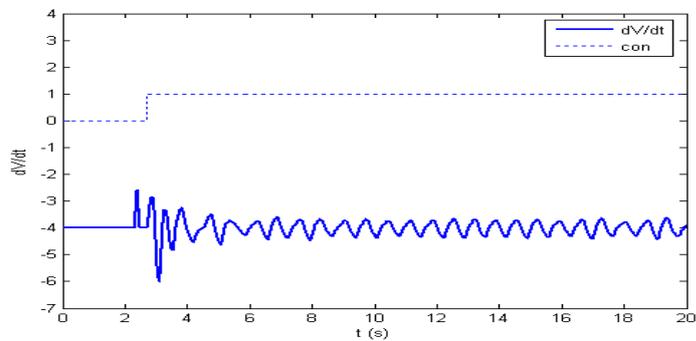
(a). Stabilisation du chariot



(b). Stabilisation du pendule ( $\theta_{k-1}$  et  $\hat{\theta}(t)$ )



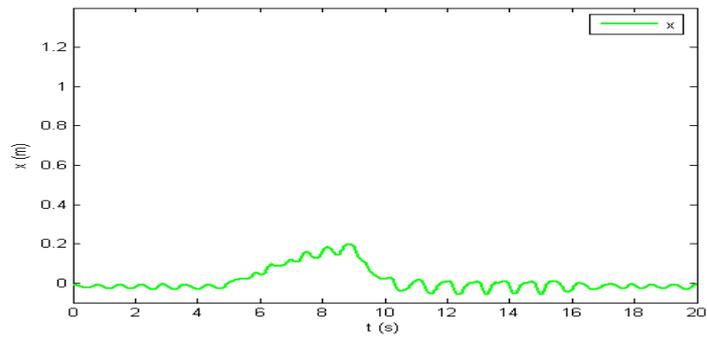
(c). Commande d'axe  $u_x$



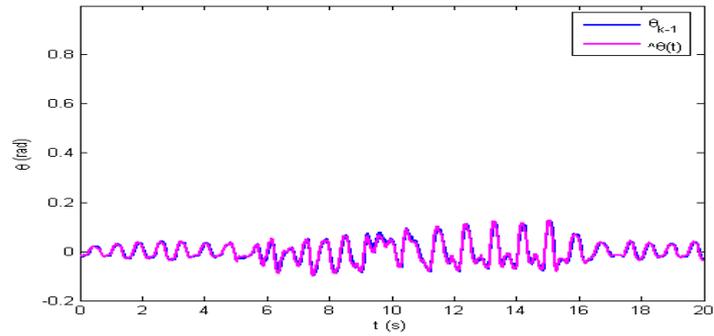
(d). Dérivé de la fonction de Lyapunov choisi

**Fig. 4.8.** Résultat temps réel de pendule inverse 2D

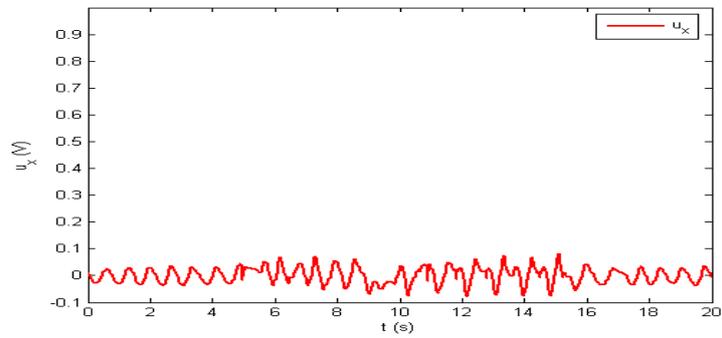
(Signal con représente l'instant où la commande commute depuis l'algorithme de démarrage à la commande de stabilisation ou plus simplement, depuis première étape à la deuxième étape)



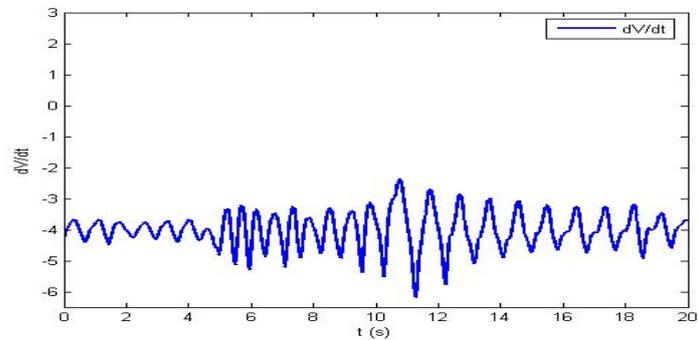
(a). Stabilisation du chariot



(b). Stabilisation du pendule ( $\theta_{k-1}$  et  $\hat{\theta}(t)$ )



(c). Commande d'axe  $u_x$



(d). Dérivé de la fonction de Lyapunov choisi

**Fig. 4.9.** Résultat temps réel de pendule inverse 2D avec perturbation manuelle

Les résultats expérimentaux montrent l'efficacité de notre commande, en particulier pour un système mécanique sous actionné ayant une dynamique élevée à déphasage non minimal.

Il subsiste toutefois de petites oscillations résiduelles, pour lesquelles deux hypothèses peuvent être faites :

- les incertitudes d'identification et les variations des paramètres (notamment le coefficient de frottement) en fonction de  $\theta(t)$ ,  $\dot{\theta}(t)$  et  $x(t)$ . sont inévitables. En particulier, il existe probablement des frottements secs difficilement identifiables, ce qui est susceptible de perturber toute la chaîne d'observation et donc la commande.
- la dynamique du pendule est très élevée. En même temps, l'utilisation d'un système de vision à bas coût implique un cycle d'acquisition d'image de 40ms. Dès lors, les incertitudes d'identification peuvent avoir une incidence non négligeable sur le fonctionnement de l'OCM, notamment dans sa composante prédictive.

### 4.3. Modélisation et commande d'un pendule inversé 3D

Dans le cas un pendule inverse 3D contrairement au 2D comme illustré figure 5.9, nous avons remplacé un joint de cardan vertical par un joint de cardan universel.



(a). Joint de cardan vertical



(b). Joint de cardan universel

**Fig. 4.10.** Joint de cardan

#### 4.3.1. Méthodes de modélisation classiques [Yan 00]

Dans un système coordonnées cartésiennes, un pendule inversé 3D peut être projeté en deux plans orthogonaux, un plan x-z et un autre plan y-z. Les deux Axes x et y considérés

sont parallèles avec les deux directions des mouvements de chariot x-y illustré par figure 4.11.

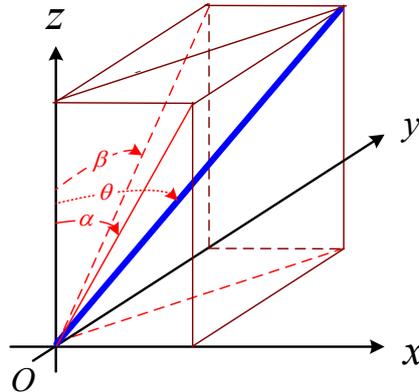


Fig. 4.11. Projection d'un pendule inversé planaire

Ces projections sont considérées en tant qu'un pendule inversé 2D couplés à l'intérieur de deux plans orthogonaux (plan x-z et plan y-z). L'angle d'inclinaison  $\theta$  est la fonction d'angles de  $\alpha$  et  $\beta$  dans des deux plans projetés. Leur moment de l'inertie par rapport à l'origine  $O$  sont décrits  $J_{xz}$  et  $J_{yz}$  respectivement. Il est évident que  $J_{xz}$  et  $J_{yz}$  sont les fonctions de leurs angles  $\alpha$  et  $\beta$  :

$$J_{xz} = \frac{l^2}{l_{xz}^2} J = \frac{1}{1 + \frac{\cos^2 \alpha}{\cos^2 \beta} \sin^2 \beta} J \quad (4.19a)$$

$$J_{yz} = \frac{l^2}{l_{yz}^2} J = \frac{1}{1 + \frac{\cos^2 \beta}{\cos^2 \alpha} \sin^2 \alpha} J \quad (4.19b)$$

avec :  $l$  longueur de pendule et  $J$  le moment de l'inertie de pendule.

Dans [Yan 00, Cho 03, Cho 03a, Jun 04], les auteurs ont proposé une méthode simplifiée pour modéliser le pendule inversé 3D. L'idée principale est présentée au dessous :

Sous l'hypothèse que l'angle d'inclinaison initiale  $\theta$  est petit  $\pm 10^\circ$ , on peut alors déduire que les angles de  $\alpha$  et  $\beta$  sont aussi petits et dans l'intervalle  $]-10^\circ \quad +10^\circ[$ . On peut faire la simplification sur les équations (4.19a) et (4.19b) en tant que considérer :  $\cos \alpha = \cos \beta = 1$ ,  $\sin \alpha = \sin \beta = 0$ , on a :

$$l_{xz} \approx l_{yz} \approx l \quad (4.20a)$$

$$J_{xz} \approx J_{yz} \approx J \quad (4.20b)$$

Après une telle simplification, le pendule 3D en 2-DOF est considéré en tant que deux pendules inversés découplés dans des deux plans orthogonaux différents, et peut être commandé individuellement. En fait, chaque système est considéré comme un pendule inversé relié avec son support de chariot, qui alternativement est conduit par une force horizontale.

Les équations de mouvement correspondant sont décrits comme suit :

$$(M_x + m)\ddot{x} + ml\ddot{\alpha} \cos \alpha - ml\dot{\alpha}^2 \sin \alpha + c_{M_x} \dot{x} = u_x \quad (4.21a)$$

$$J\ddot{\alpha} + c_{m_x} \dot{\alpha} - mgl \sin \alpha = -ml\ddot{x} \cos \alpha \quad (4.21b)$$

$$(M_y + m)\ddot{y} + ml\ddot{\beta} \cos \beta - ml\dot{\beta}^2 \sin \beta + c_{M_y} \dot{y} = u_y \quad (4.22a)$$

$$J\ddot{\beta} + c_{m_y} \dot{\beta} - mgl \sin \beta = -ml\ddot{y} \cos \beta \quad (4.22b)$$

Dans lesquelles : les équations (4.21a) et (4.21b) sont la projection sur le plan x-z, alors que (4.22a) et (4.22b) sont pour celui sur l'avion de y-z;  $M_x$  et  $M_y$  sont la masse de plateforme en respectant à chaque axe;  $m$ ,  $l$  et  $J$  sont la masse, la longueur, et le moment de l'inertie du pendule inversé, respectivement;  $c_{M_x}$  et  $c_{M_y}$  sont des coefficients de frottement visqueux selon les axes x et y;  $c_{m_x}$  et  $c_{m_y}$  sont coefficients de frottement visqueux de rotation (dans le plan x-z et le plan y-z) et  $u_x$  et  $u_y$  sont des force entrée fournie par deux moteurs sur l'axe x et l'axe y.

Ce type de modélisation simplifiée et approximative a une forte désavantage : il n'est valide que quand l'angle d'inclinaison initiale  $\theta$  est petit  $|\theta| \leq 10^\circ$ . Le pendule 3D ne peut être plus considéré en tant que deux pendules inversés découplés dans des deux plans orthogonaux différents.

### 4.3.2. Modélisation et commande

Contrairement à la méthode Nous développons ici une méthode de modélisation valable quel que soit l'angle d'inclinaison du pendule  $\theta$ .

L'idée principale consiste à travailler, à chaque instant d'échantillonnage de vision, dans un plan du pendule de stabilisation. Ainsi, on transforme le problème 3D en un problème 2D dans le plan du pendule.

On applique ensuite la méthode en doubles boucles développées précédemment en tenant compte des difficultés d'utilisation un retour visuel avec une cadence lente. La méthode que nous avons proposée a une large convergence.

#### 4.3.2.1. Modélisation

##### 4.3.2.1.1. Pendule inversé 3D par chariot x-y

En analysant la structure de notre pendule inversé 3D illustré figure I.1 (dans l'annexe I), on désigne par  $P$  le point de liaison du pendule avec le chariot (cardan) et par  $O$  l'origine du plan  $x$ - $y$  de la table. Dans un premier temps, on considère le repère  $Px'y'z'$  parallèle au repère  $Oxyz$  lié à la table. Comme l'illustre la figure 4.12, le pendule inversé peut être

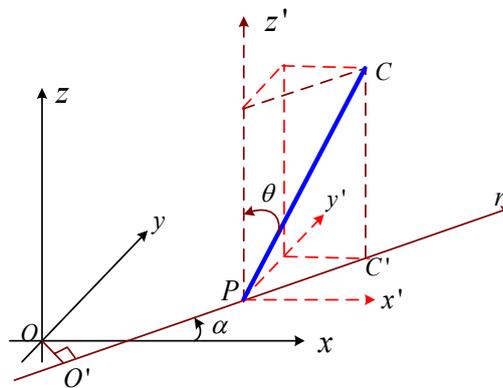


Fig. 4.12. Projection sur le plan  $r$ - $z'$  et  $x$ - $y$ - $z$

projeté, à tout instant, sur deux plans orthogonaux : le plan  $x$ - $y$  (plan de la table) 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 à la modélisation du pendule inversé 2D par, le pendule inversé 3D peut être modélisé comme suit, dans le plan  $r$ - $z'$  :

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} - \omega_n^2 \sin \theta = -K \cos \theta \ddot{r} \quad (4.23a)$$

où :  $\omega_n = \sqrt{mgl/(J + ml^2)}$  le fréquence naturel,  $\zeta = B_r/[2\omega_n(J + ml^2)]$ , le coefficient d'amortissement,  $K = ml/(J + ml^2)$  le gain du système, dans lesquelles :

- $\theta$  angle du pendule avec l'axe  $Pz'$ ,  $l$  longueur du  $\frac{1}{2}$  pendule,  $m$  masse du pendule,  $B_r$  frottement visqueux entre le pendule et le chariot,  $g$  accélération de la pesanteur,  $J = ml^2/3$  moment d'inertie du pendule et  $r$  position du chariot sur l'axe  $r$ , l'origine de cet axe étant la projection  $O'$  de  $O$  sur le plan  $r$ - $z'$ .

Naturellement, le déplacement du chariot sur l'axe  $r$  résulte d'une combinaison de mouvements selon les axes  $x$  et  $y$ .

Comme le montre la figure 4.13, en considérant la projection du pendule sur le plan  $x$ - $y$ , nous pouvons trouver facilement la relation reliant  $x$ ,  $y$  et  $r$  :

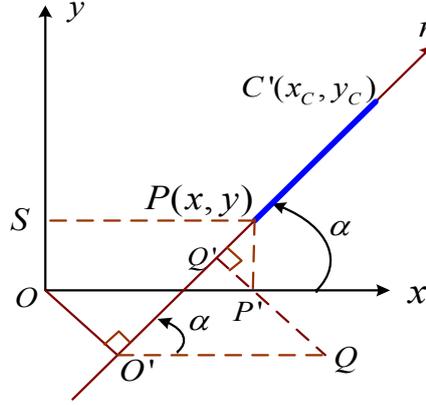


Fig. 4.13. Projection sur le plan  $x$ - $y$

$$r = x \cos \alpha + y \sin \alpha \quad (4.23b)$$

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

D'après I.2.1, le chariot  $x$ - $y$  est modélisé comme :

$$\ddot{x} = (-\dot{x} + k_x u_x) / \tau_x \quad (4.23c)$$

$$\ddot{y} = (-\dot{y} + k_y u_y) / \tau_y \quad (4.23d)$$

où :  $k_x$  ( $k_y$ ) et  $\tau_x$  ( $\tau_y$ ) sont respectivement le gain statique et la constante de temps du chariot motorisé selon l'axe  $x$  (l'axe  $y$ ) et  $u_x$  ( $u_y$ ) sa tension d'entrée.

#### 4.3.2.1.2. Traitement des signaux visuels

D'après la structure de notre plateforme, les informations que nous pouvons utiliser sont les coordonnées  $(x_C, y_C)$  du haut du pendule obtenues par le système de vision. Et les coordonnées  $(x, y)$  du pied du pendule mesurées par les codeurs. Ainsi selon les figures 4.12 et 4.13, les expressions de  $\alpha$  et  $\theta$  peuvent s'écrire :

$$\alpha = \tan^{-1} [(y_C - y) / (x_C - x)] \quad (4.24)$$

$$\theta = \sin^{-1} \{ [(x_C - x) \cos \alpha + (y_C - y) \sin \alpha] / 2l \} \quad (4.25)$$



En comparant les deux architectures de commande, la différence se trouve l'entrée de la *commande basée sur une fonction de Lyapunov* dans la « Boucle de stabilisation » et la sortie de *linéarisation et stabilisation* dans la première boucle de « Boucle d'observation du pendule » (tous les  $x$ ,  $\dot{x}$  et  $\ddot{x}$  sont remplacés par  $r$ ,  $\dot{r}$  et  $\ddot{r}$ ).

Alors, d'après l'équation (4.16), nous avons la commande correspondante ici :

$$\tilde{v} = -[\chi r + \mu(1 - e^{-(\chi r^2 + \delta \dot{r}^2)/2})\dot{r}]/\delta \quad (4.27)$$

Les restes à déterminer sont les relations entre  $\tilde{v}$  et  $u_x$  ou  $u_y$ , notés par les fonctions  $f_1(\tilde{v}, \dot{x})$  et  $f_2(\tilde{v}, \dot{y})$  dans la figure 4.4.

Pour cela, en considérant (4.23a) et (4.23b) et en réalisant des boucles locales appropriées, nous obtenons  $k_x=k_y=k$  et  $\tau_x=\tau_y=\tau$ .

Dès lors, pour équilibrer le pendule inversé dans son plan de stabilisation, il suffit d'imposer que le rapport des accélérations selon les axes x et y soit égal à  $\tan \alpha$  :

$$\ddot{y}/\ddot{x} = \tan \alpha \quad (4.28)$$

Cette relation supplémentaire permet de calculer les relations entre  $\tilde{v}$ ,  $u_x$  et  $u_y$  comme illustré sur la figure 4.14 :

$$u_x = f_1(\tilde{v}, \dot{x}) = (\dot{x} + \tau \cos \alpha \tilde{v})/k \quad (4.29a)$$

$$u_y = f_2(\tilde{v}, \dot{y}) = (\dot{y} + \tau \sin \alpha \tilde{v})/k \quad (4.29b)$$

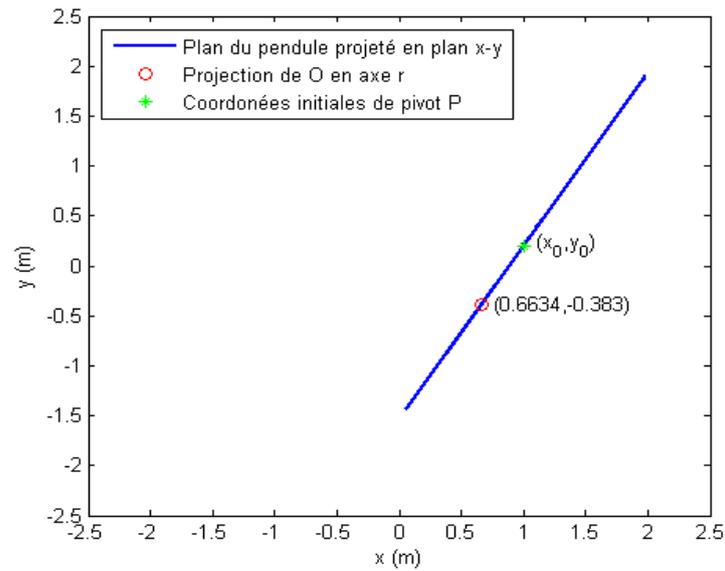
Nous avons ainsi fini notre commande.

### 4.3.3. Validations expérimentales

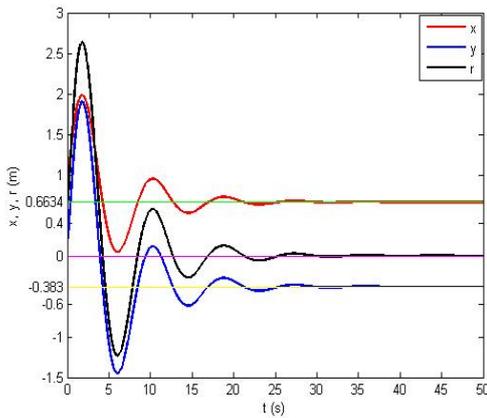
Dans ce cas pour le pendule inversé qu'on utilise les même paramètres que §4.2.5. Les paramètres du système moteur-chariot sont imposés par les boucles locales à  $k = 2.92$  et  $\tau = 0.008$  s, selon les axes x et y. Enfin les valeurs des différents paramètres introduits précédemment dans la commande sont :  $K_2 = \omega_n^2$ ,  $\xi = 1.2$ ,  $\Omega_n = \omega_n$ ,  $\chi = 2.5$ ,  $\delta = 10$  et  $\mu = 8$ .

#### 4.3.3.1. Sans perturbation

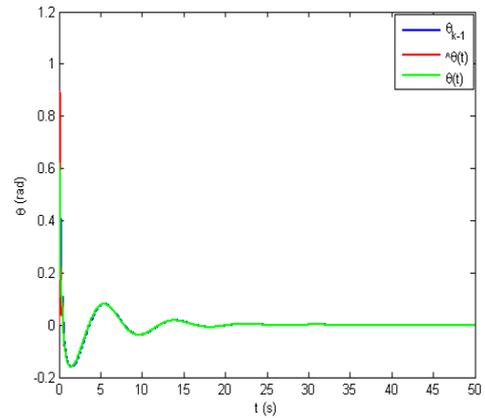
La figure 4.15 illustre la stabilisation dans un plan quelconque.



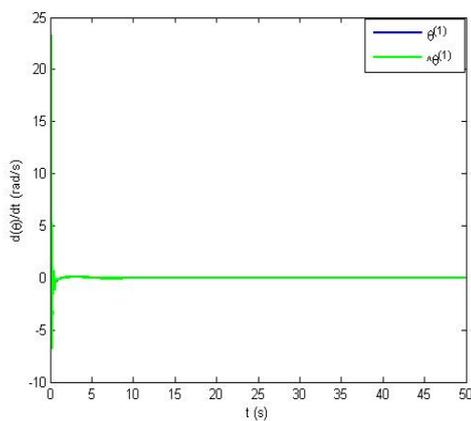
(a). Projection du plan du pendule sur plan x-y



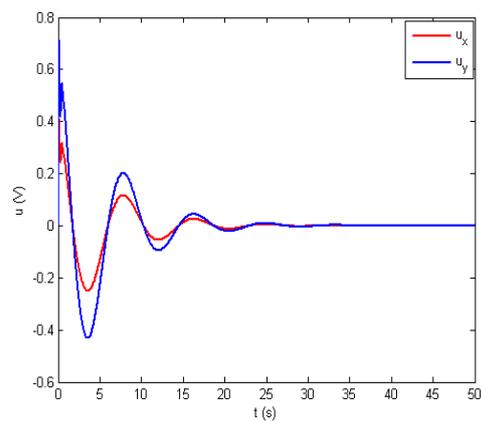
(b). Déplacements selon x, y et r



(c).  $\theta_{k-1}$ ,  $\hat{\theta}(t)$  et  $\theta(t)$



(d).  $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$



(e). Commandes  $u_x$  et  $u_y$

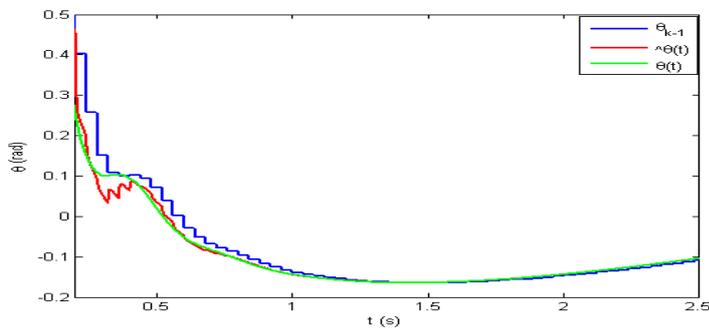
**Fig. 4.15.** Résultats de simulation 3D sans perturbation

$$\{ \alpha_0 = \pi/3 \text{ rad}, \theta_0 = \pi/6 \text{ rad}, x_0 = 1 \text{ m}, y_0 = 0.2 \text{ m} \}$$

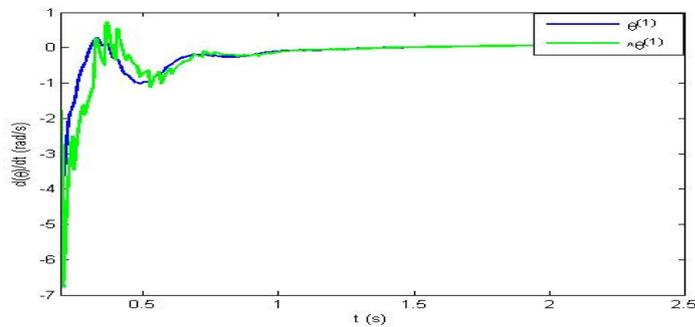
Cette figure fait apparaître : (a) la projection du plan du pendule sur le plan x-y, (b) les déplacements selon  $x$ ,  $y$  et  $r$ , (c) les positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ ), (d) les vitesses angulaires du pendule ( $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$ ) et finalement (e) les commandes d'axes ( $u_x$ ,  $u_y$ ).

Il faut souligner que la stabilisation de l'ensemble chariot-pendule, à l'intérieur de la zone autorisée, se fait bien à l'origine  $O$  de l'axe  $r$  d'après figure 4.15a et figure 4.15b.

Enfin, il faut souligner également que même si en 3D, l'application de l'hypothèse de quasi-stationnarité n'apporte aucune restriction. Ainsi, aucune oscillation permanente n'apparaît, ni sur le déplacement du chariot, ni sur le pendule.



(a). Positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ )



(b). Vitesses angulaires du pendule ( $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$ )

**Fig. 4.16.** Zoom sur OCM de la figure 4.15

Pour faciliter la lecture des performances de l'OCM, nous avons de plus fait un zoom des figures 4.15c et figure 5.15d, sur l'intervalle de temps  $[0.2s, 2.5s]$ . Les résultats correspondants apparaissent figure 4.16. L'OCM donne pareillement d'excellentes estimations de  $\theta(t)$  et  $\dot{\theta}(t)$ .

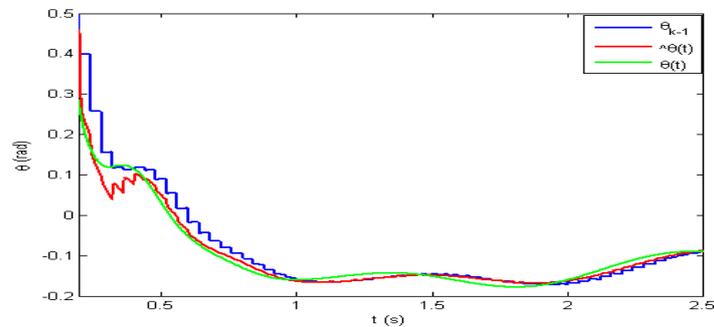
### 4.3.3.2. Avec perturbations

Dans ce cas, nous considérons le même type de perturbations que §4.2.5.1.2 pour  $B_r$  et  $\theta_{k-1}$ . Par ailleurs nous avons ajouté simultanément : une erreur de mesure sur  $\alpha_{k-1}$  de la forme  $0.03 \cos 2\pi t$ .

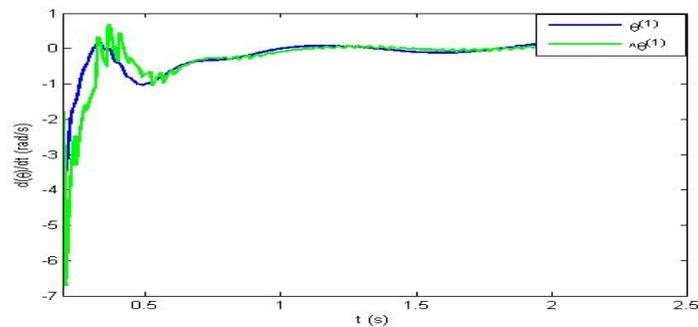
Les résultats de simulation correspondants apparaissent figure 4.17 (cf. Page suivante).

Il apparaît que, même dans ce cas, le système chariot-pendule reste stable.

L'OCM donne encore des estimations tout à fait satisfaisantes (figure 4.18, zoom de la figure 4.17d et figure 4.17e).

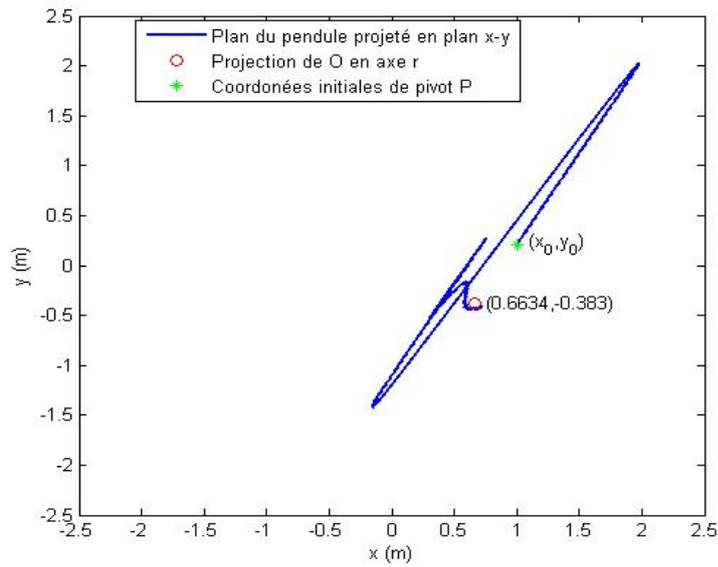


(a) Positions angulaires du pendule ( $\theta_{k-1}$ ,  $\theta(t)$  et  $\hat{\theta}(t)$ )

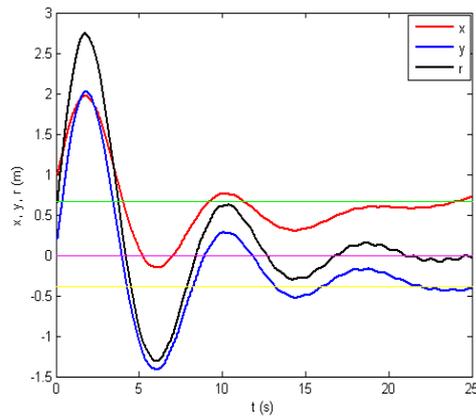


(b) Vitesses angulaires du pendule ( $\dot{\theta}(t)$  et  $\dot{\hat{\theta}}(t)$ )

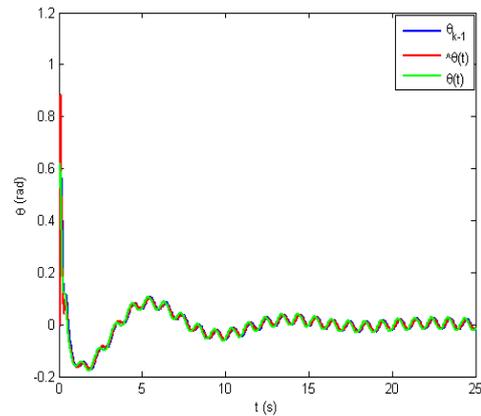
**Fig. 4.18.** Zoom sur OCM de la figure 4. 17



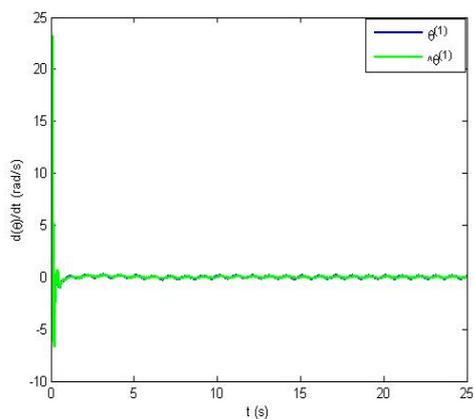
(a) Projection du plan du pendule sur plan x-y



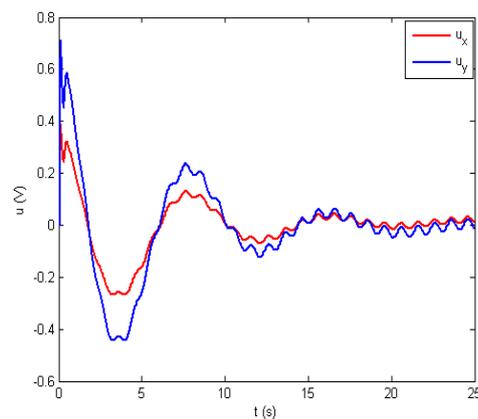
(b) Déplacements selon x, y et r



(c)  $\theta_{k-1}$ ,  $\hat{\theta}(t)$  et  $\theta(t)$



(d)  $\dot{\theta}(t)$  et  $\hat{\dot{\theta}}(t)$



(e) Commandes  $u_x$  et  $u_y$

**Fig. 4.17.** Résultats de simulation 3D avec perturbation

$$\{ \alpha_0 = \pi/3 \text{ rad}, \theta_0 = \pi/6 \text{ rad}, x_0 = 1 \text{ m}, y_0 = 0.2 \text{ m} \}$$

## 4.4. Conclusion

Ce chapitre présente une stratégie de commande d'un pendule inversé 2D puis 3D en utilisant un retour fourni par une caméra à bas coût et par deux codeurs. 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 et les résultats expérimentaux 2D montrent l'efficacité de notre commande, en particulier pour un système mécanique sous actionné ayant une dynamique élevée à déphasage non minimal.

Il subsiste toutefois de petites oscillations résiduelles dans les résultats expérimentaux pour le pendule 2D, pour lesquelles trois hypothèses peuvent être faites :

- les incertitudes d'identification et les variations des paramètres (notamment le coefficient de frottement) en fonction de  $\theta(t)$ ,  $\dot{\theta}(t)$  et  $x(t)$ . sont inévitables. En particulier, il existe probablement des frottements secs difficilement identifiables, ce qui est susceptible de perturber toute la chaîne d'observation et donc la commande.
- la dynamique du pendule est très élevée. En même temps, l'utilisation d'un système de vision à bas coût implique un cycle d'acquisition d'image de 40ms. Dès lors, les incertitudes d'identification peuvent avoir une incidence non négligeable sur le fonctionnement de l'OCM, notamment dans sa composante prédictive.
- La discrétisation spatiale introduite par les imperfections du système de vision utilisé (résolution de la caméra...) est également susceptible de perturber le bon fonctionnement de l'algorithme de commande et de générer des oscillations lors de la phase finale de commande. Cependant, l'objectif de notre stratégie de commande a été justement d'assurer un niveau de performances acceptable, même en utilisant des boucles de retour immatérielles de faible qualité (low cost).

En dépit de ces difficultés, la stratégie de commande que nous avons proposée permet de compenser et de stabiliser l'ensemble chariot-pendule. Cela est corroboré par une étude de robustesse, qui prend en compte aussi bien des incertitudes sur  $\alpha_{k-1}$  et  $\theta_{k-1}$ , que sur les paramètres du système.



# Conclusion générale et perspectives

## Conclusion générale

Le travail effectué dans ce mémoire montre l'importance des capteurs numériques sans contact (caméras...) intégrés dans la boucle de rétroaction. De plus, le contexte industriel actuel impose dans la plupart des cas l'utilisation de capteurs à bas coût (low cost) impliquant manque de précision, dynamique plus lente et donc fréquence d'échantillonnage limitée et retard de mesure. Les lois de commandes doivent cependant assurer une bonne qualité et la possibilité de pilotage en temps réel.

Les nouvelles méthodes développées dans le cadre de cette thèse de doctorat se basent essentiellement sur une classe particulière de systèmes hybrides : les SFM (Systèmes à fonctionnement par Morceaux), conçus et développés par Vasseur et Koncar dans [Kon 01, Kon 02, Kon 03].

Dans ce sens...

- Le chapitre 1, au travers de la définition des systèmes hybrides donnée par SANTIS, est consacré à la présentation de la plupart des systèmes hybrides existants. L'étude globale des systèmes hybrides nous permet de situer les SFM et de les généraliser dans un sens plus large pour modéliser les systèmes hybrides multi-modèles. Ils se caractérisent principalement par l'existence de deux espaces temps et deux espaces commandes. De manière générale, selon que la dynamique du système est continue ou discrète, on les appelle : Systèmes Continus à Fonctionnement par Morceaux (SCFM) ou Systèmes Echantillonnés à Fonctionnement par Morceaux (plus simplement Systèmes Bi-Echantillonnés (SBE)). Les SFM sont bien adaptés et facilement transposables sur les calculateurs numériques ou les architectures temps réels. Toutefois, les propositions hybrides théoriques sont souvent trop compliquées pour une implantation directe sur des calculateurs numériques destinés à la modélisation des systèmes réels complexes naturellement hybrides. Dans notre cas, les systèmes hybrides sont conçus

artificiellement afin d'identifier, linéariser, estimer, prédire ou commander des procédés industriels complexes.

Dans les trois chapitres suivants, les SFM sont utilisés dans le cadre de la modélisation et de la linéarisation des systèmes à commander (continus, discrets ou hybrides), dans le développement de méthodes d'estimation, de prédiction et de commande des systèmes réels complexes y compris des systèmes non linéaires, sous actionnés et à déphasage non minimal :

- Ainsi dans le chapitre 2, sur la base du formalisme des SFM, nous avons réalisé, d'abord des CFM qui garantissent la poursuite échantillonnée d'une trajectoire d'état à partir de l'état du processus à commander. Selon que la dynamique du système associé est continue ou échantillonnée entre les instants de commutation, on définit les CCM ou les CBE. Ensuite nous avons développé un contrôleur spécifique appelé Contrôleur Récursif (CR) assurant la poursuite d'une trajectoire de sortie à partir de la sortie continue du processus à commander. Afin de prendre en compte des éventuelles discontinuités sur la consigne, nous avons également proposé un amortisseur qui permet d'éviter une commande excessive. L'intégration de cet amortisseur dans l'architecture de commande avec CFM, et surtout CR, fait apparaître des propriétés intéressantes. Des exemples de simulation et d'implantation temps réel sur la plateforme de chariot x-y illustrent la robustesse de ce type de contrôleur, même en présence de perturbations et de variations des paramètres du processus.

L'intérêt de l'asservissement visuel réside dans les nombreuses applications potentielles. Ainsi, il peut s'avérer impossible de mettre le capteur en contact direct avec l'objet commandé. Les systèmes de vision artificielle permettent de pallier facilement 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. Dans les travaux de recherche présentés dans cette étude, l'enjeu est d'utiliser une caméra 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 des temps de calcul (traitement d'images) et de transmission des signaux dans la boucle.

- Le chapitre 3 commence par une analyse des difficultés liées à la commande par rétroaction visuelle. Cela 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). La mise en œuvre de

l'observateur OCM est basée sur l'utilisation de deux SFM et d'un observateur de Luenberger d'ordre réduit. Cet observateur présente l'avantage de compenser les effets de retard et d'échantillonnage pour fournir des estimations continues et sans retard. La validation de cet observateur pour réaliser une poursuite robuste par retour visuel d'un robot X-Y est présentée. Les vidéos illustrant cette validation sont accessibles sur le site : <http://www-lagis.univ-lille1.fr/~wang/Research.html>. Elles montrent que les méthodes proposées sont robustes vis à vis des perturbations ou des pertes de mesures.

Enfin dans le chapitre 4...

- Pour les systèmes non linéaires mécaniques sous actionnés ayant une dynamique à déphasage non minimal, la commande d'un pendule inversé 2D a été développée, selon une architecture à deux boucles :
  - La première boucle de retour traite les informations échantillonnées et retardées délivrées par le système de vision. Le traitement vise à réaliser un modèle stable et linéaire du pendule bouclé sans aucune approximation et à en reconstruire l'état continu à l'aide d'un OCM,
  - La seconde boucle vise à stabiliser l'ensemble chariot-pendule, selon une commande utilisant une fonction de Lyapunov.
- Cette méthode est appliquée avec succès sur la plate-forme réelle, même si l'on ajoute une perturbation manuelle persistante. La perturbation consiste à essayer de déséquilibrer le pendule en agissant sur son extrémité. On constate que le chariot se déplace de manière à assurer l'équilibre. Dès que la perturbation cesse, l'algorithme maintient l'équilibre du pendule, tout en ramenant le chariot à l'origine. Les vidéos correspondantes sont également accessibles sur le site : <http://www-lagis.univ-lille1.fr/~wang/Research.html>. On y constatera que lorsque l'on rend le système aveugle (en masquant la LED), l'équilibre est rompu. Les essais ont montré que l'ensemble chariot-pendule reste stable plus de 10 heures (nous n'avons pas testé au-delà). En analysant les méthodes existantes de commande du même type de système, il apparaît que notre méthode possède de forts avantages en termes de robustesse, de réduction des oscillations ( $\pm 2^\circ$ ) et de simplicité d'implantation. De plus, pour commander les systèmes ayant une structure de pendule inverse 3D, les autres auteurs utilisent normalement 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 avons développé 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. Nous utilisons ainsi les deux boucles de commande proposées précédemment pour commander le pendule 3D

## **Perspectives de recherche**

### **Théoriques**

Dans le cadre du développement des contrôleurs hybrides, des études sont prévues pour analyser la robustesse des méthodes proposées en utilisant les SFM ce qui permettra de déterminer rigoureusement la marge de tolérance des variations de paramètres du processus, des perturbations extérieures ou des erreurs de mesures provoquées par des capteurs à bas coût avec faible précision, (caméra intégrée dans la maquette 2D d'asservissement visuel).

Par ailleurs, nous envisageons d'étudier les instants de commutation dans les SFM dans le but d'améliorer les performances des contrôleurs. Pour l'instant, les instants de commutation sont imposés à une cadence fixe. Un mécanisme de commutation qui détermine les instants de commutation à une cadence variable dépendant principalement de la stabilité, de la robustesse et des perturbations du système à commander peut être envisagé.

### **Applicatives**

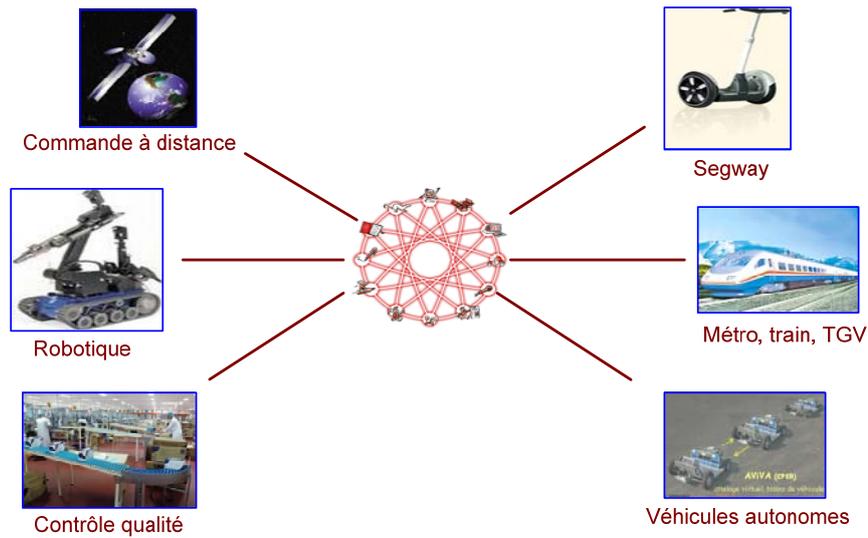
La validation des méthodes de poursuite par retour visuel permettra à terme une mise en œuvre au sein du projet « Attelage Virtuel » (suite du projet AViVA) qui consiste à « accrocher » un véhicule meneur (avec conducteur) et un suiveur (autonome) grâce à des capteurs « immatériels » ou « sans contact ».

Par ailleurs, l'intégration de systèmes de navigation tels que GPS (Etats Unis), GLONASS (Russie), GALILEO (Europe) et BEIDOU (Chine) permettra de réaliser la commande à distance, la commande via réseaux de transports autonomes (guidage, détection et localisation des défaillances).

La figure 5.1 illustre des applications potentielles des méthodes que nous avons développées.

### **Pédagogiques**

En présentant ma recherche sous forme de vidéos dans mon enseignement de TD/TP à l'ENSAIT, afin de fournir des manipulations démonstratives de l'automatique, j'ai remarqué que les étudiants s'intéressaient fortement aux applications réelles.



**Fig. 5.1.** Applications potentielles

J'envisage donc, dans un futur proche, de créer des projets, TD ou TP utilisant la technologie Matlab/Simulink/dSpace, en espérant contribuer à la motivation et à la connaissance des étudiants<sup>1</sup>.

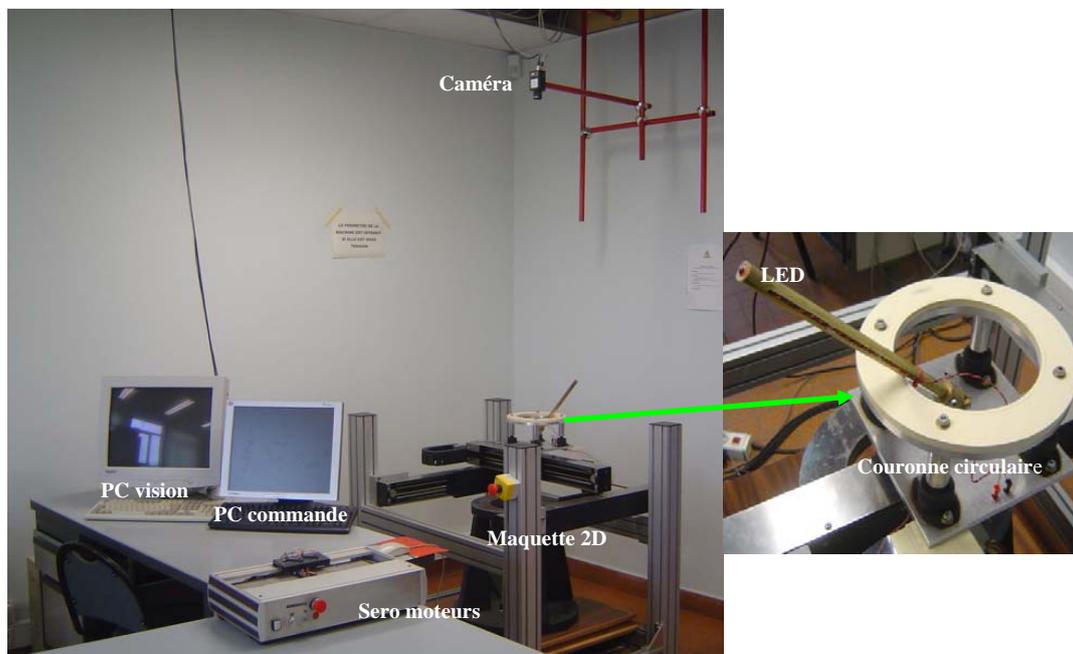
<sup>1</sup> Je suis actuellement ATER à temps plein à l'UFR d'IEEA, USTL.

# Annexe I

## Plate-forme de pendule inversé 3D à retour visuel

Les méthodes de poursuite échantillonnée, d'estimation et de commande par retour visuel développées dans les chapitres 2, 3 et 4 ont été implantées sur la plate-forme d'essai présentée figure I.1.

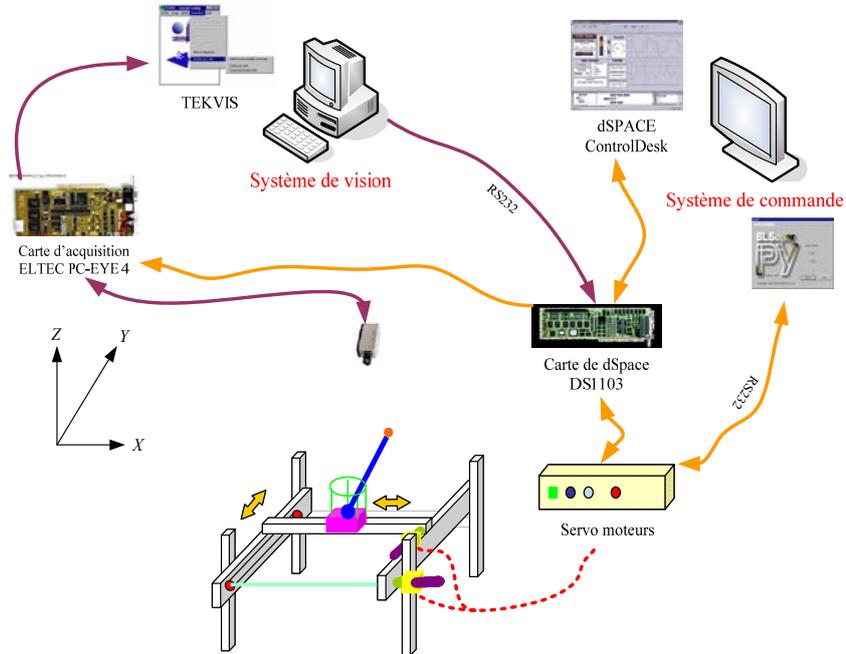
Cette plate-forme est située au LAGIS, bâtiment P2-314, USTL. Elle peut être déclinée selon plusieurs versions : chariot x-y à retour continu, chariot x-y à retour visuel et pendule inversé 2D à retour visuel.



**Fig. I.1.** Plate-forme d'essai temps réel

## I.1. Plate-forme expérimentale de pendule inversé 3D à retour visuel

La plateforme d'essai est constituée de quatre sous ensembles comme illustré figure I.2 :



**Fig. I.2.** Plate-forme de pendule inversé 3D

- 1) une table X-Y entraînant un chariot sur lequel le pendule inversé est fixé par un joint de cardan,
- 2) une caméra CCD associée à un système de vision (TEKVIS) utilisée pour capturer la position angulaire du pendule,
- 3) un contrôleur implanté sur une carte de technologie dSpace (DS1103) reliée à un superviseur ControlDesk intégré dans les logiciels Matlab/Simulink,
- 4) deux actionneurs (servomoteurs) commandés par la carte dSpace.

### I.1.1 Système mécanique chariot-pendule

Le système se compose d'une table X-Y, dont la zone de déplacement est un carré de 0,48 m de côté, et d'un pendule inversé (tube de plastique ou tube de laiton) fixé au chariot par un joint de cardan. 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.

L'axe x se déplace le long de l'axe y par un guidage à billes. Les axes de x et y sont motorisés par deux servomoteurs (voir figure I.3).

### **I.1.2 Système de vision**

Contrairement aux caméras numériques de haute résolution (jusqu'à  $4000 \times 4000$  pixels) travaillant jusqu'à 2000 images/sec avec un rapport signal/bruit très élevé, la caméra que nous utilisons est une caméra ordinaire infra rouge de faible coût (Jai M50 IR), dont la cadence d'échantillonnage est de 25 images/sec et la résolution de  $640 \times 480$  pixels. Cette caméra fixée à environ 2 mètres au-dessus du centre de la table X-Y est configurée en mode non entrelacé.

Le système de vision associé à la 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  $(x_C, y_C)$  (en pixels) de la LED fixée à l'extrémité haute du pendule et les transmet au système de commande par une liaison RS-232. La prise d'image de la caméra est déclenchée et synchronisée par 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 construite lors d'un calibrage préalable (méthode "TSAI" présentée dans [Hei 97]). 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, les coordonnées de la LED sont disponibles avec une période de 40 ms.

### **I.1.3 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 quasiment en temps réel les variables de la carte. Les signaux de commande sont, quant à eux, envoyés aux amplificateurs de puissance par un CAD ( $\pm 10$  V).

### **I.1.4 Actionneurs**

Le chariot de la table X-Y est motorisé à l'aide de deux servomoteurs SANYO DENKI PY2A015A3-A3, dont les configurations peuvent être modifiées et paramétrées par le

logiciel PY, à partir de l'ordinateur de commande. L'entraînement du chariot se fait par courroies crantées.

De plus, deux codeurs (un par axe) mesurent la position en continu du chariot ( $x, y$ ). L'amplificateur alimenté en 240 V mono offre un courant maximal de 15 A. Les motoréducteurs brushless délivrent un couple nominal de 3.0 Nm avec une puissance de 200 W.

## I.2. Modélisation du système de pendule inversé 3D

### I.2.1 Chariot motorisé x-y

Si, en faisant un zoom sur figure I.1, on démonte le pendule inversé et la couronne circulaire et si l'on fixe une LED infrarouge sur le chariot, on obtient un simple système

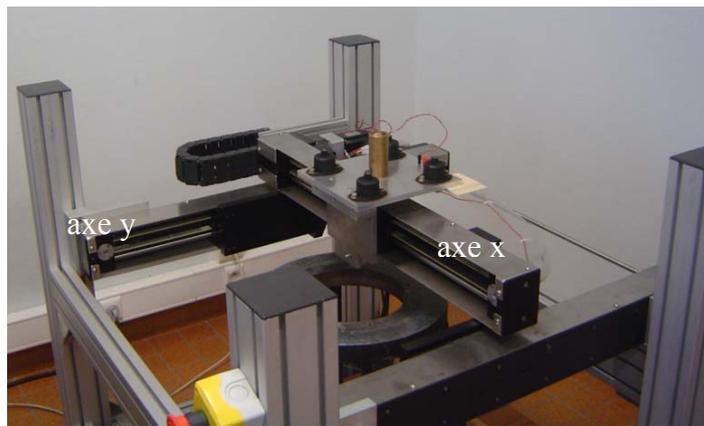


Fig. I.3. Chariot x-y

de chariot x-y à retour visuel, tel qu'illustré figure I.3. Le mouvement du chariot résulte de la combinaison des mouvements sur chaque axe. On peut donc le modéliser selon deux axes indépendants comme suit :

L'ensemble amplificateur-moteur-chariot selon l'axe x est modélisé par un système linéaire du second ordre défini par :

$$\ddot{x} = (-\dot{x} + k_x u_x) / \tau_x \quad (I.1)$$

où :  $k_x$  et  $\tau_x$  sont respectivement le gain statique et la constante de temps du chariot motorisé selon l'axe x et  $u_x$  sa tension d'entrée.

Il en va de même pour l'axe y :

$$\ddot{y} = (-\dot{y} + k_y u_y) / \tau_y \quad (\text{I.2})$$

## I.2.2 Pendule inversé 2D

La figure I.3 représente un pendule inversé 2D, dans lequel :

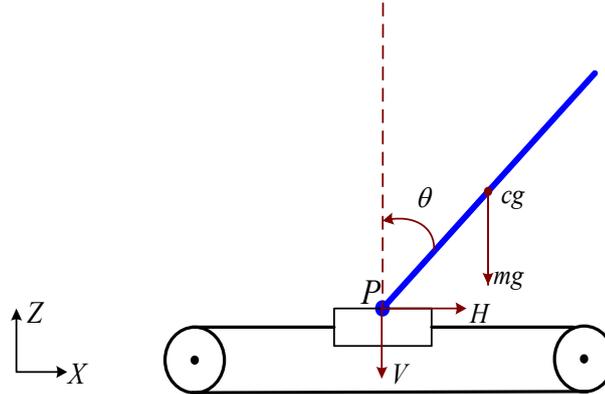


Fig. I.3. Le pendule inversé 2D dans le plan  $x$ - $z$

$cg$  : Centre de gravité,

$P$  : point bas du pendule, pivot du chariot,

$x_{cg}$  : Abscisse du centre de gravité selon l'axe  $x$ ,

$z_{cg}$  : Ordonnée du centre de gravité selon l'axe  $z$ ,

$x$  : Abscisse du pivot (les  $X$  positifs sont vers la droite),

$z$  : Ordonnée du pivot (les  $Z$  positifs sont vers le haut),

$H$  : Force horizontale,

$V$  : Force verticale,

$l$  : 1/2 longueur du pendule,

$m$  : Masse du pendule,

$\theta$  : Angle du pendule avec la verticale,

$B_r$  : Constant de viscosité,

$I$  : Moment d'inertie du pendule ( $ml^2/3$  pour une barre simple)

$g = 9.81 \text{ m/s}^2$  : Accélération de la pesanteur.

En appliquant la seconde loi du mouvement de Newton sur le centre gravité du pendule [Bug 03] ou [Del 07], on obtient :

$$\sum F_x = m\ddot{x}_{cg} \quad \text{avec : } x_{cg} = x + l \sin(\theta)$$

$$\sum F_z = m\ddot{z}_{cg} \quad \text{avec : } z_{cg} = z + l \cos(\theta) \quad \text{et } \dot{z} = 0$$

soit :

$$H = m[\ddot{x} + l \cos(\theta)\ddot{\theta} - l \sin(\theta)(\dot{\theta})^2]$$

$$V = -m[l \sin(\theta)\ddot{\theta} + l \cos(\theta)(\dot{\theta})^2 - g]$$

Puis, on écrit les sommes des moments autour de centre de gravité :

$$\sum M_{cg} = I\ddot{\theta} + B_r\dot{\theta}$$

$$\sum M_{cg} = Vl \sin(\theta) - Hl \cos(\theta)$$

On obtient finalement la relation entre  $x$  et  $\theta$  comme suit :

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} - \omega_n^2 \sin(\theta) = -K \cos(\theta)\ddot{x} \quad (I.3)$$

avec :  $2\zeta\omega_n = \frac{3B_r}{4ml^2}$ ,  $\omega_n^2 = \frac{3g}{4l}$  et  $K = \frac{3}{4l}$ .

### I.2.3 Pendule inversé 3D

Dans [Cho 03, Cho 03a, Jun 04, Yan 00], 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 dans 4.3.2.1 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 dans le plan du pendule.

### I.2.4 Traitement des signaux visuels

D'après la structure de notre plateforme, les informations que nous pouvons utiliser sont les coordonnées  $(x_C, y_C)$  du haut du pendule obtenues par le système de vision et les coordonnées  $(x, y)$  du pied du pendule mesurées par les codeurs. Ainsi selon 4.3.2.1.2 en utilisant deux types d'information mentionnés ci-dessus et illustrés par les équations (4.24) et (4.25), on peut calculer les angles  $\alpha_{k-1}$  et  $\theta_{k-1}$  retardés d'une période  $t_e$ .

### I.2.5 Conclusion

En récapitulatif, le banc essais de pendule inversé 3D à retour visuel nous donne trois différents types de retour :

- Premier type : retour continu mesuré par codeur. C'est ce retour qui est utilisé pour la poursuite échantillonnée présentée au chapitre 2,
- Deuxième type : retour échantillonné par système de vision  $(x_{C,k-1}, y_{C,k-1})$ . Ce retour est utilisé pour la poursuite échantillonnée à retour visuel présentée au chapitre 3,
- Troisième type : retour hybride, c'est à dire la position continue du chariot  $(x, y)$  et l'angle retardé et échantillonné de  $\alpha_{k-1}$  et  $\theta_{k-1}$ . Ce retour est utilisé pour la commande de stabilisation des pendules inversés 2D et 3D présentée au chapitre 4.



## Annexe II

# Systèmes mécaniques sous actionnés à déphasage non minimal et hypothèse de quasi-stationnarité

## II.1 Systèmes mécaniques sous actionnés à déphasage non minimal

Les systèmes mécaniques sous actionnés possèdent plus de degrés de liberté que d'entrées de commande. La stabilisation de ce type de systèmes est encore un défi [Fan 00] [Ort 02] [Ria 07]. Dans cette annexe, nous présentons la synthèse des systèmes mécaniques sous actionnés à déphasage non minimal. Ces systèmes mécaniques évoluent dans un espace de dimension  $n$  de coordonnées  $(q_1, \dots, q_n)$ , avec un espace de commande à  $m$  actionneurs ( $m < n$ ). Ils peuvent être décrits sous la forme générale suivante :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_a = B\tau \quad (\text{II.1})$$

d'où,  $q$  est le vecteur de position des articulations,  
 $\dot{q}$  et  $\ddot{q}$  sont respectivement les vecteurs de vitesse et d'accélération,  
 $\tau \in R^m$  avec  $m < n$  est la commande d'entrée,  
 $M(q)$  est la matrice d'inertie,  
 $C(q, \dot{q})$  est la matrice centrifuge et de Coriolis,  
 $F(\dot{q})$  sont les forces de frottements,  
 $G(q)$  est le vecteur de gravité,  
 $\tau_a$  représente des perturbations ou des parties de modélisations inconnues,  
 $B$  est le vecteur d'entrée.

Le système (II.1) peut être réécrit sous forme suivante :

$$\ddot{q} = M(q)^{-1} [-C(q, \dot{q})\dot{q} - F(\dot{q}) - G(q) - \tau_d] + M(q)^{-1} B\tau, \quad (\text{II.2})$$

Ensuite, si l'on introduit un vecteur d'état  $x \in R^n$  et une commande d'entrée  $u \in R^m$ , on peut obtenir un modèle non linéaire affine en la commande, selon l'équation suivante :

$$\dot{x} = f(x) + g(x)u, \quad x(0) = x_0 \quad (\text{II.3})$$

Pour des raisons de simplicité on considère que la sortie  $y = h(x) \in R$  est mesurable, avec  $h(0) = 0$ . En utilisant les outils de la géométrie différentielle, on examine l'ordre  $r$  correspondant à la première dérivée de  $y$  faisant apparaître explicitement la commande  $u$ .

Si la valeur de  $r$  est constante et  $r \leq n$ , le système (II.3) peut être linéarisé par la méthode de découplage entrées-sorties, en adoptant la forme de Byrnes-Isidori à deux étapes [Isi 94, Gue 05] :

1. On utilise un retour d'état qui peut compenser les non linéarités dans la dynamique du système :

$$u = \frac{v - L_f^r h(x)}{L_g L_f^{r-1} h(x)}, \quad (\text{II.4})$$

d'où  $L_f N(x) = \frac{\partial N}{\partial x} f(x)$  désigne la dérivée de Lie dans la direction du champ de vecteurs  $f$ . D'après la définition de  $r$ , on a :  $L_g L_f^{r-1} h(x) \neq 0$  et  $L_f^{i-1} h(x) = 0, \forall i < r - 1$ .

2. On applique par la suite une transformation non linéaire de coordonnées  $z = \Phi(x)$ , dans laquelle on note  $z = [z_1 \ z_2 \ \dots \ z_n]^T$  et  $\Phi(x) = [\phi_1 \ \phi_2 \ \dots \ \phi_n]^T$ . Et on suppose que  $z = [y \ \dot{y} \ \dots \ y^{(r-1)} \ \eta^T]^T$ , où  $\eta \in R^{n-r}$  est définie de telle sorte que la matrice jacobienne de  $\Phi(x)|_{x_0}$  soit inversible. Le système (III.3) est alors décrit par :

$$y^{(r)} = v, \quad (\text{II.5})$$

$$\dot{\eta} = Q(y, \dot{y} \dots y^{(r-1)}, \eta), \quad (\text{II.6})$$

avec  $y(0) = y_0, \eta(0) = \eta_0$ .

Le vecteur  $\eta$  est utilisé pour représenter les états correspondant aux dynamiques internes du système.

Le système est à déphasage non minimal si le système représentée par les équations (II.5) et (II.6) a des valeurs propres positive, à condition que toutes ses entrées soient nulles et que ses conditions initiales soient telles que sa sortie soit nulle,. Dans ce cas, il suffit de vérifier si l'équation suivante :

$$\dot{\eta} = Q(0, \eta), \quad (\text{II.7})$$

a des valeurs propres positives.

En général, nous choisissons les vecteurs d'état avec une dynamique lente pour  $\eta$ . Dans le chapitre 4, en comparant les dynamiques du chariot et du pendule inversé, nous adoptons comme vecteur  $\eta$  l'état du chariot et comme sortie  $y$  la position angulaire  $\theta$  du pendule inversé.

## II.2 Hypothèse de quasi-stationnarité

Une fois le système (II.3) linéarisé, une commande de retour peut être appliquée sur l'équation linéarisée d'entrée-sortie définie par (II.5) :

$$y^{(r)} = v, \quad v = K_1(y_{ref} - y) - \sum_{i=1}^{r-1} K_{i+1}y^{(i)} \quad (\text{II.8})$$

où  $y_{ref}$  est la consigne de référence pour la sortie  $y$ , et  $K_1, \dots, K_n$  sont les coefficients de l'équation polynomiale de Hurwitz. Ces paramètres sont choisis tels que le système bouclé de l'équation II.8 soit stable avec une dynamique rapide. Pour un bon fonctionnement,  $y_{ref}$  doit évoluer lentement, afin que la sortie  $y$  suive facilement.

En introduisant la notation  $\xi_i = y^{(i-1)}$ ,  $i=1, \dots, r$ , le système (II.5) peut être écrit sous la forme suivante :

$$\dot{\eta} = Q(\xi, \eta), \quad (\text{II.9})$$

$$\dot{\xi}_i = \xi_{i+1}, \quad i = 1, \dots, r-1 \quad (\text{II.10})$$

$$\dot{\xi}_r = K_1(y_{ref} - \xi_1) - \sum_{i=1}^{r-1} K_{i+1}\xi_{i+1}, \quad (\text{II.11})$$

avec  $\xi = [\xi_1 \quad \xi_2 \quad \dots \quad \xi_r]^T$ .

L'hypothèse de quasi-stationnarité donne l'équation interne du système (II.9) sous la forme suivante :

$$\dot{\eta} = Q(\bar{\xi}, \eta) \equiv \bar{Q}(y_{ref}, \eta), \quad \eta(0) = \eta_0 \quad (\text{II.12})$$

où  $\bar{\xi} = \xi|_{\xi_1=y_{ref}, \xi_2=\xi_3=\dots=\xi_r=0} = [y_{ref} \quad 0 \quad \dots \quad 0]^T$  est le vecteur de quasi-stationnarité de  $\xi$ .

Le système de (II.10) et (II.11) peut donc être écrit sous la forme canonique suivante :

$$\frac{d\tilde{\xi}}{dt} = A\tilde{\xi}, \quad \tilde{\xi}(0) = \tilde{\xi}_0 \quad (\text{II.13})$$

où  $\tilde{\xi} = \xi - \bar{\xi}$ , avec :

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -K_1 & -K_2 & -K_3 & \dots & -K_r \end{bmatrix} \quad (\text{II.14})$$

Le système (II.12) et (II.13) représente les sous systèmes à complexité réduite issus du système initial (II.3).

Les deux boucles de retour utilisées au chapitre 4 permettent d'appliquer le principe énoncé ci-dessus : la première boucle correspond au retour de linéarisation représenté par l'équation (II.8), la seconde boucle correspond à la commande développée (par exemple, à partir d'une fonction de Lyapunov) pour le sous système résultant de l'hypothèse de quasi-stationnarité traduite par l'équation (II.12).

## Annexe III

### Algorithme de démarrage

Cet algorithme permet de pallier la difficulté de démarrage, lorsque le pendule est appuyé sur sa couronne avec un angle initial important. Dans ce cas, en effet, le modèle non linéaire du second ordre n'est plus valide.

Pour comprendre l'algorithme, nous considérons un passager adossé au siège d'une voiture. Dans notre cas, le chariot joue le rôle de la voiture, le pendule celui du passager et la couronne celui du dossier du siège. D'après les équations (4.2a) et (4.2c), l'équation du chariot selon l'axe  $x$  est définie par :  $\ddot{x} = \tilde{v}$ .

Nous appliquons alors une commande  $\tilde{v}$  en trois étapes :

- *Première étape* : on accélère fortement sur l'intervalle ( $t \in [0, t_1]$ ) quand  $t \in [0, t_1]$ ,  $x_0 \neq 0$  et  $\tilde{v} = -\varepsilon M$ ,  $M > 0$ .

Par l'intégration de  $\ddot{x} = \tilde{v}$ , on a :

$$x_{t_1} = x_0 - \varepsilon M t_1^2 \text{ et } \dot{x}_{t_1} = -\varepsilon M t_1 \quad (\text{III.1})$$

- *Deuxième étape* : on cesse d'accélérer pour  $t \in ]t_1, t_2]$  ( $\tilde{v} = 0$ ) :

$$x_{t_2} = x_0 + \varepsilon M t_1^2 / 2 - \varepsilon M t_1 t_2 \text{ et } \dot{x}_{t_2} = \dot{x}_{t_1} \quad (\text{III.2})$$

- *Troisième étape* : on freine fortement sur  $t \in ]t_2, T]$ ,  $x_T = 0$  et  $\tilde{v} = \varepsilon \alpha M$ , avec :  $\alpha > 0$ . On obtient les relations suivantes par intégration de l'équation  $\ddot{x} = \tilde{v}$  :

$$t_1 = \alpha(T - t_2), \text{ et } M = \varepsilon x_0 / [T(t_1 + \alpha t_2) - t_1^2 + \alpha t_2^2 + \alpha T^2 / 2] > 0. \quad (\text{III.3})$$

Lors de la phase d'accélération, le pendule est plaqué sur la couronne, puis il s'incline vers l'avant lors de la phase de freinage. Dès lors, par un choix approprié des paramètres  $T$ ,  $t_1$  et  $t_2$ , on parvient à ce que l'angle du pendule passe au voisinage de zéro avec une vitesse faible, pour  $x(T) = 0$  et  $\dot{x}(T) = 0$ , avec une condition initiale  $x(0) \neq 0$ .

Dès que la position angulaire du pendule arrive dans la zone  $|\theta| \leq 0.2$ , on commute sur la commande de stabilisation.

# Annexe IV

## A. Bibliographie personnelle

### REVUES INTERNATIONALES

H. P Wang, C. Vasseur, V. Koncar et A. Chamroo, Commande d'un pendule inversé 3D par asservissement visuel, *JESA : Journal Européen des Systèmes Automatisés*, Vol. X, No.9, Octobre 2008.

H. P. Wang, C. Vasseur, A. Chamroo and V. Koncar, Sampled tracking for linear delayed plants using piecewise functioning controller, *WSEAS Transactions on Systems and Control*, Issue 8, Vol. 2 : 405 – 410, August 2007.

### CHAPITRE D'OUVRAGE

H.P. Wang, A. Chamroo, C. Vasseur, V. Koncar and N. Christov., A new trajectory tracking approach for a vision based X-Y robot, *Jubilee Collection dedicated to the 60th Anniversary of Mihail Konstantinov*, Sofia, Bulgarian Academy of Sciences.

### CONFERENCES INTERNATIONALES AVEC ACTES & COMITE DE LECTURE

H. P. Wang, C. Vasseur, V. Koncar et A. Chamroo, Modélisation et commande d'un pendule inversé sur un robot X-Y par l'asservissement visuel, *CIFA'08, 5<sup>e</sup> IEEE Conférence Internationale Francophone d'Automatique*, UPB, Bucarest, 3 – 5 septembre 2008.

H. P. Wang, A. Chamroo, C. Vasseur, V. Koncar et N. Christov., Suivi de trajectoire avec retour visuel par l'approche des systèmes à fonctionnement par morceaux, *CIFA'08, 5<sup>e</sup> IEEE Conférence Internationale Francophone d'Automatique*, UPB, Bucarest, 3 – 5 septembre 2008.

H. P Wang, A. Chamroo, C. Vasseur and V. Koncar, Hybrid control for vision based Cart-inverted Pendulum system, *2008 American Control Conference*, Regular paper. FrA07.1 : 3845 – 3850, Seattle, Washington, USA, June 11 – 13, 2008.

H. P. Wang, A. Chamroo, C. Vasseur and V. Koncar, Stabilization of a 2-DOF inverted pendulum by a low cost visual feedback, *2008 American Control Conference*, Regular paper. FrA07.2 : 3851 – 3856, Seattle, Washington, USA, June 11 – 13, 2008.

H. P. Wang, A. Chamroo, C. Vasseur, V. Koncar and N. Christov., Trajectory tracking of a vision based x-y robot by Piecewise continuous systems”, *MECATRONICS’08, 7th France-Japan / (5th Europe-Asia) Congress on Mechatronics*, Le Grand-Bornand, FRANCE, May 21 – 23, 2008.

H. P. Wang, C. Vasseur and V. Koncar, Piecewise continuous systems used in trajectory tracking of a vision based x-y robot, *IEEE CISSE-IETA’07, International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering*, University of Bridgeport, USA, December 3 – 12, 2007.

H. P. Wang, C. Vasseur, A. Chamroo and V. Koncar, Sampled tracking for delayed systems using piecewise functioning controller, *CESA’06, IMACS-IEEE Multiconference on Computational Engineering in Systems Applications*, Vol. 2 : 1326 – 1333, Beijing, China, October 4 – 6, 2006.

A. Chamroo, A. Seuret, C. Vasseur, J.-P. Richard and H. P. Wang, Observing and controlling plants using their delayed and sampled outputs, *CESA’06 IMACS-IEEE Multiconference on Computational Engineering in Systems Applications*, Vol. 1 : 851 – 857, Beijing, China, October 4 – 6, 2006.

V. Koncar, H. P. Wang, A. Chamroo and C. Vasseur, Piecewise continuous systems used for control and identification, *Invited Lecture in 3rd International Conference of Applied Mathematics – Included in International Journal of Pure and Applied Mathematics (IJPAM, ISSN 1311 – 8080)*, Plovdiv, Bulgaria, August 15 – 16, 2006.

H. P. Wang, A. Chamroo, C. Vasseur and V. Koncar, Poursuite échantillonnée à partir de l’état retardé et échantillonné, *CIFA’06, 4<sup>e</sup> IEEE Conférence Internationale Francophone d’Automatique*, Bordeaux, France, 30 mai – 1er juin 2006.

A. Chamroo, C. Vasseur and H. P. Wang, Plant control using digital sensors that introduce a delayed and sampled output, *ELMA’05, 11th International Conference on Electrical Machines, Drives and Power Systems (Supported by IEEE Bulgaria Section)*, Sofia, Bulgaria, Vol. 1 : 119 – 124, September 15 – 16, 2005.

### DISSERTATION DE MEMOIRE

H. P. Wang, Suivi de trajectoire par la commande modifiée, Master. dissertation, Université Lille 1, Villeneuve d’Ascq, France, Juin, 2005.

## B. Bibliographie

- [Alu 93] R. Alur, C. Courcoubetis, T. A. Henzinger and P. H. Po, Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems, in Hybrid Systems, *Lecture Notes in Computer Science*, Vol. 736 (R. L. Grossman, A. Nerode, A. R. Ravn and H. Rischel, Eds.), Springer-Verlag, pp. 209-229, New York, 1993.
- [Ant 98] P. Antsaklis, X. Koutsoukos and J. Zaytoon, On hybrid control of complex systems: A survey, *APII-JESA*, Vol. 32, No. 9 : 1023 – 1045, 1998.
- [Ast 00] K. J. Astrom and K. Furuta, Swinging up a pendulum by energy control, *Automatica*, Vol. 36 : 287 – 295, 2000.
- [Arm 05] A. R. Armando, P. M. Richard, C. Oguzhan and D. Thanate, 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 : 359 – 374, Aug. 2005.
- [Bai 89] D. D. Bainov and P. S. Simeonov, *Systems with impulse effect*, Chichester, UK, Ellis Horwood, 1989.
- [Bac 93] A. Back, J. Guckenheimer and M. Myers, A dynamical simulation facility for hybrid systems, *Hybrid Systems, Lecture Notes in Computer Science*, Vol. 736 (R. L. Grossman, A. Nerode, A. R. Ravn and H. Rischel, Eds.), Springer-Verlag, New York : 255 – 267, 1993.
- [Bem 99] A. Bemporad and M. Morari, Control of systems integrating logic, dynamics and constraints, *Automatica*, Vol. 35, No. 3 : 407 – 427, 1999.
- [Ben 84] A. Bensoussan and J.-L. Lions, *Impulse control and quasi-variational inequalities*, Paris, Gauthier-Villars, 1984.
- [Bes 07] G. Besançon, *Nonlinear Observers and applications*, Springer, 2007.
- [Bir 06] A. Birouche, *Contribution sur la synthèse d'observateurs pour les systèmes dynamiques hybrides*, Ph.D. dissertation, Ecole doctorale IAEM Lorraine, Nancy, France, 2006.
- [Bod 45] H.W. Bode, *Network analysis and feedback amplifier design*, New York, Van Nostrand, 1945
- [Bor 90] P. Borne, C. Dauphin-Tanguy, J. -P. Richard, F. Rotella and I. Zambetakis, *Commande et optimisation des processus*, Editions Technip, 1990.

- [Bor 05] F. Borrelli, M. Baotic, A. Bemporar and M. Morari, Dynamic programming for constrained optimal control of discrete-time linear hybrid system, *Automatica*, Vol. 41 : 1709 – 1721, 2005.
- [Bou 00] R. Boukezzoula, *Commande floue d'une classe de systèmes non linéaires : application au problème de suivi de trajectoire*, Ph.D. dissertation, Université de Savoie, France, 2000.
- [Bra 94] M. S. Branicky, V. Borkar and S. K. Mitter, A unified framework for hybrid control, *Proceedings of the IEEE Conference of Decision Control*, Lake Buena Vista, FL., USA : 4228 – 4234, December 1994.
- [Bra 95] M. S. Branicky, *Studies in hybrid systems: Modeling, analysis, and control*, Ph.D. dissertation, Department of Electrical and Computer Engineering, Massachusetts Institute of Technology, 1995.
- [Bra 98] M. S. Branicky, V. S. Borkar and S. K. Mitter, A unified framework for hybrid control: Model and optimal control theory, *IEEE Transaction on Automatic Control*, Vol. 43, No. 1 : 31 – 45, 1998.
- [Bra 98a] M. S. Branicky, Multiple Lyapunov functions and other analysis tools for switched and hybrid systems, *IEEE Transaction on Automatic Control*, Vol. 43, No. 4 : 475 – 482, 1998.
- [Bug 03] M. Bugeja, Non-linear swing-up and stabilizing control of an inverted pendulum system, *EUROCON*, Ljubljana, Slovenia, 2003.
- [Bus 99] K. K. Busawon and M. Saif, A state observer for nonlinear systems *IEEE Transactions on Automatic Control*, Vol. 44, No. 11 : 2098 – 2103, Nov. 1999.
- [Cha 05] A. Chamroo, C. Vasseur and V. Koncar, HAOPI: hybrid adaptive online plant identification, 17th IMACS World Congresss, Paris, France, July 2005.
- [Cha 05a] A. Chamroo, C. Vasseur and H. P. Wang, Plant control using digital sensors that introduce a delayed and sampled output, *11th ELMA World Congress*, Vol. 1 : 119 – 124, Sofia, Bulgaria, Sep. 2005.
- [Cha 06] A. Chamroo, *Contribution à l'étude des systèmes à fonctionnement par morceaux: application à l'identification en ligne et à la commande en temps réel*, Ph.D. dissertation, Université Lille 1, Villeneuve d'Ascq, France, Juin 2006.
- [Cha 06a] A. Chamroo, A. Seuret, C. Vasseur, J.-P. Richard and H. P. Wang, Observing and controlling plants using their delayed and sampled outputs, *CESA'06, IMACS-IEEE Multiconference on Computational Engineering in Systems Applications*, Vol. 1 : 851 – 857, Beijing, China, October 2006.
- [Chat 02] D. Chatterjee, A. Patra and H. Joglekar, Swing-up and stabilization of a cart-pendulum system under restricted cart track length, *System and Control Letters*, Vol. 47, No. 4 : 355 – 364, Nov 2002.

- [Chau 07] F. Chaumette and N. Andreff, Quoi de neuf en asservissement visuel depuis les JNRR'07, *Journées nationales de la recherche en robotique 2007*, Obernai, Oct. 2007.
- [Che 97] L. Chen and R. Smith, Closed-loop model validation for an inverted pendulum experiment via a linear matrix inequality approach, *Proc. Of the 36th CDC* : 2565 – 2566, San Diego, USA, Dec 1997.
- [Che 04] C. K. Chen and J. Hwang, Iterative learning control for position tracking of a pneumatic actuated X-Y table, *Proceedings of the 2004 IEEE International Conference on Control Applications*, Vol. 1 : 388- 393, Sep 2004.
- [Cho 96] H. H. Choi and M. J. Chung, Observer-based  $H_\infty$  controller design for state delayed linear systems, *Automatica*, Vo. 32, No. 7 : 1073-1075, July 1996.
- [Cho 03] H. K. Cho and S. Jung, 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* : 1210 – 1215, 2003, .
- [Cho 03a] H. K. Cho and S. Jung, 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* : 181 – 186, Las Vegas, Oct 2003.
- [Cur 98] E. R. Cury José, H. K. Bruce and N. Toshihiko, Synthesis of supervisory controllers for hybrid systems based on approximating automata, *IEEE Transaction on Automatic Control*, Vol. 43, No. 4 : 564 – 568, April, 1998.
- [Dar 01] M. Darouach, Linear functional observers for systems with delays in state variables. *IEEE Transactions on Automatic Control*, Vo. 46, No. 3 : 491 – 496, 2001.
- [Dar 05] M. Darouach, Linear functional observers for systems with delays in state variables: the discrete-time case, *IEEE Transactions on Automatic Control*, Vo. 50, No. 2 : 228 – 233, Feb 2005.
- [Dec 00] R. A. Decarlo, M. S. Branicky, S. Peterson and B. Lennartson, Perspective and results on the stability and stabilization of hybrid systems, *Proceedings of the IEEE*, Vol. 88, No. 7, July 2000.
- [Def 07] M. Defoort, *Contributions à la planification et la commande pour les robots mobiles coopératifs*, Ph.D. dissertation, Ecole centrale de Lille, Villeneuve d'Ascq, France, Oct 2007.
- [Dej 94] B. de Jager, Acceleration assisted tracking control, *IEEE Control Systems Magazine*, Vol. 14, No. 5 : 20 – 27, 1994.
- [Del 07] D. W. Deley, Controlling an inverted pendulum: example of a digital feedback control system, Available: <http://members.cox.net/srice1/pendulum/index.htm>, Jan 4, 2007.

- [Esp 06] E. S. Espinoza-Quesada and L. E. Ramos-Velasco, Vision based control of an underactuated system using a reduced observer, *Electronics, Robotics and Automotive Mechanics Conference*, Vol. 1 : 9 – 14, Sep. 2006.
- [Fan 00] I. Fantoni, R. Lozano and M. Sponge, Energy based control of the pendubot, *IEEE Trans. Aut. Contr.*, Vol. 45, No. 4 : 725-729, 2000.
- [Fan 05] H. Fang, R. Lenain, B. Thuilot and P. Martinet, Trajectory tracking control of farm vehicles in presence of sliding, *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* : 58 – 63, Aug. 2005.
- [Fie 99] R. Fierro, F. L. Lewis and A. Lowe, 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 : 649 – 654, 1999.
- [For 77] T. E. Fortmann and K. L. Hitz, *An introduction to linear control systems*, New York, Marcel Dekker, 1977.
- [Fri 04] E. Fridman, A. Seuret and J.-P. Richard, Robust sampled-data stabilization of linear systems: An input delay approach, *Automatica*, Vol. 40, No. 8 : 1441 – 1446, 2004.
- [Fuk 06] 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* : 4016 – 4021, Busan, Korea, Oct. 2006.
- [Gey 03] T. Geyer, F. Torrisi and M. Morari, Efficient mode enumeration of compositional hybrid models, *Hybrid Systems: Computation and Control*, Vol. 2623 : 216 – 232, 2005.
- [Gue 05] K. Guemghar, B. Srinivasan, P. Mullhaupt and D. Bonvin, Analysis of cascade structure with predictive control and feedback linearization, *IEE Control Theory & Applications*, Vol. 152, No. 3 : 317 – 324, May 2005.
- [Hei 97] J. Heikkila and O. Silven, A four-step camera calibration procedure with implicit image correction, *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* : 1106 – 1112, Jun 1997.
- [Hor 63] I. M. Horowitz, *Synthesis of feedback systems*, New York Academic, 1963.
- [Hut 96] S. Hutchinson, G.D. Hager and P.I. Corke, A tutorial on visual servo control, *IEEE Trans. on Robotics and Automation*, Vol. 12, No. 5 : 651 – 670, Oct. 1996.
- [Isi 94] Isidori A., *Nonlinear control systems*, Springer, Third edition, 1994.
- [Jia 07] Z. H. Jiang, T. Ishida, Trajectory tracking control of industrial robot manipulators using a neural network controller, *2007. ISIC. IEEE International Conference on Systems, Man and Cybernetics* : 2390 – 2395, Oct. 2007.

- [Jun 04] S. Jung and H. T. Cho, 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 : 92 – 99, March 2004.
- [Kab 87] P. T. Kabamba, Control of linear systems using generalized sampled-data hold functions, *IEEE Transaction on Automatic Control*, Vol. 32, No. 9 : 772 – 783, 1987.
- [Kai 80] T. Kailath, *Linear systems*, Englewood Cliffs, NJ: Prentice-Hall, 1980.
- [Kal 61] R. E. Kalman and R. S. Bucy, *New results in linear filtering and prediction theory*, *Trans. ASME, Series D, Journal of Basic Engineering*, 83 : 95 – 108, March 1961
- [Khe 02] A. Kheddar and P. Coiffet, *Téléopération et réalité virtuelle*, Hermès-Lavoisier, 2002
- [Kim 81] H. Kimura and Y. Tanaka, Minimum-time minimum-order deadbeat regulator with internal stability, *IEEE Transactions on Automatic Control*, Vol. 26 : 1276 – 1282, December 1981
- [Kon 00] V. Koncar and C. Vasseur, Tracking by compound control, *Studies in Informatics and Control Journal*, Vol. 9, No. 4, December 2000.
- [Kon 01] V. Koncar et C. Vasseur, Systèmes à fonctionnement par morceaux et poursuite échantillonnée, *APII-JESA*, Vol. 35, No. 5 : 665 – 689, 2001.
- [Kon 02] V. Koncar and C. Vasseur, Piecewise functioning systems: bi-sampled controllers, *Studies in Informatics and Control*, Vol. 11, No. 2 : 185 – 198, 2002.
- [Kon 03] V. Koncar and C. Vasseur, Control of linear systems using piecewise continuous systems, *IEE Control Theory & Applications*, Vol. 150, No. 6 : 565 – 576, 2003.
- [Kon 06] V. Koncar, H. P. Wang, A. Chamroo and C. Vasseur, Piecewise continuous systems used for control and identification, *Invited Lecture in 3rd International Conference of Applied Mathematics*, Plovdiv, Bulgaria, August 2006.
- [Kor 82] Y. Koren, and O. Masory, Reference-word circular interpolation for CNC system, *Trans. ASME J. Eng. Ind.*, Vol. 104 : 400–405, 1982.
- [Lan 88] I. D. Landau, *Identification et commande des systèmes*, Hermès, Paris, 1988.
- [Lau 72] F. Laurent, Sur la commande d'un filtre linéaire par des impulsions multimodulées, *C.R. Acad. Sc, Paris*, Vol. 270 : 288 – 289, 1972.
- [Lel 01] A. Leleve. P. Fraisse and P. Dauchez, Telerobotics over IP networks: Towards a low-level real-time architecture, *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2 : 643 – 648, 2001.

- [Lib 03] D. Liberzon, *Switching in systems and control*, Birkhauser Boston, 2003.
- [Lin 04] F.J. Lin, R.J. Wai, and P.K. Huang, Two-axis motion control system using wavelet neural network for ultrasonic motor drives, *IEE Proc.-Electr. Power Appl.*, Vol. 151, No. 5 : 613 – 621, Sep. 2004.
- [Lin 06] F. J. Lin, H. J. Shieh, P. H. Shieh and P. h. Shen, An adaptive recurrent-neural-network motion controller for X-Y table in CNC machine, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 36, No. 2 : 286 – 299.
- [Loz 00] R. Lozano, I. Fantoni and D. J. Block, Stabilization of the inverted pendulum around its homoclinic orbit, *Systems & Control Letters*, Vol. 40 : 194 – 204, 2000.
- [Lue 66] D. G. Luenberger, Observers for multivariable systems, *IEEE Transactions on Automatic Control*, Vol. 11, No. 2 : 190 – 197, 1966
- [Lue 71] D. G. Luenberger, An Introduction to observers, *IEEE Transactions on Automatic Control*, Vol. AC-16, No. 1 : 596 – 602, 1971.
- [Mag 98] M. E. Magana and F. Holzapfel, Fuzzy-logic control of an inverted pendulum with vision feedback, *IEEE Trans. On Education*, Vol. 41, Issue 2 : 165 – 170, May 1998.
- [Mat 04] T. Matsuo, R. Yoshino, H. Suemitsu and K. Nakano, Nominal performance recovery by PID+Q controller and its application to antisway control of crane lifter with visual feedback, *IEEE Trans. on Control System Technology*, Vol. 12, No.1 : 156 – 166, Jan. 2004.
- [Mic 99] A. N. Michel and B. Hu, Toward stability theory of general hybrid dynamical systems, *Automatica*, Vol. 35, No. 3 : 371 – 384, 1999.
- [Ner 93] A. Nerode and W. Kohn, Models for hybrid systems: automata, topologies, controllability, observability, in Hybrid Systems, *Lecture Notes in Computer Science*, Vol. 736 (R. L. Grossman, A. Nerode, A. R. Ravn and H. Rischel, Eds.), Springer-Verlag, pp. 317 – 356, New York, 1993.
- [Ohs 95] A. Ohsumi and T. Izumikawa, Nonlinear control of swing-up and stabilization of an inverted pendulum, *Proc. Of the 34th CDC*, New Orleans : 3873 – 3880, Dec. 1995.
- [Olf 99] R. Olfati-Saber, Fixed point controllers and stabilization of the cart-pole system and the rotating pendulum, *Proc. Of the 38th CDC*, Phoenix : 1174 – 1181, Dec 1999.
- [O'Re 81] J. O'Reilly, The discrete linear time invariant time-optimal control problem – An overview, *Automatica*, Vol. 17, No. 2 : 363 – 370, 1981.
- [Ort 02] R. Ortega, M. W. Spong, F. Gomez-Estern and G. Blankenstein, stabilization of a class of underactuated mechanical systems via interconnection and damping assignment, *IEEE Trans. Aut. Contr.*, Vol. 47, No. 8 : 1218 – 1233.

- [Plo 04] N. J. Ploplys, P. A. Kawka and A. G. Alleyne, Closed-loop control over wireless networks, *IEEE Control Systems Magazine*, Vol. 24, No. 3 : 58 – 71, Jun 2004.
- [Ria 07] S. Riachy, T. Floquet et J. P. Richard, Stabilisation de systèmes sous-actionnés par commande quasi-homogène, *e-STA, Revue électronique Science et technologies de l'Automatique*, Vol. 4, No. 3, 2007.
- [Ric 04] J.-P. Richard, Time delay systems: an overview of some recent advances and open problems, *Automatica*, Vol. 39 : 1667 – 1694, 2004.
- [Rot 95] F. Rotella and P. Borne, *Théorie et pratique du calcul matriciel*, Editions Technip, Paris, 1995.
- [Sal 04] T. Salgado-Jimene, Contribution à la commande d'un robot sous-marin autonome de type torpille, Ph.D. dissertation, Université Montpellier II, France, Décembre, 2004.
- [San 04] J. Sanchez, S. Dormido R. Pastor and F. Morilla, A Java/Matlab-based environment for remote control system laboratories: Illustrated with an inverted pendulum, *IEEE Trans. on Education*, Vol. 47, No. 3 : 321 – 329, Aug. 2004.
- [San 03] E. D. Santis, M. D. Benedetto, S. D. Gennaro and P. Giordano, *Hybrid observer design methodology*, Technical Report ST-2001-32460 of European Commission, Distributed control and stochastic analysis of hybrid systems supporting safety critical real-time system design (HYBRIDGE), 2003.
- [Sel 02] R. R. Selmic and F. L. Lewis, Neural-network approximation of piecewise continuous functions: Application to friction compensation, *IEEE Transactions on Neural Networks*, Vol. 13, No. 3, 745 – 751, May 2002.
- [Sen 01] O. Sename, New trends in design of observers for time-delay systems, *Kybernetika*, 37(4), 427 – 458, 2001.
- [Seu 06] A. Seuret, *Commande et observation des systèmes à retards variables: théorie et applications*, Ph.D. dissertation, Ecole centrale de Lille, Villeneuve d'Ascq, France, Oct 2006.
- [Tor 04] F. Torrisi and A. Bemporad, Hysdel-a tool for generating computational hybrid models for analysis and synthesis problems, *IEEE Transactions on Control System Technology*, Vol. 12 : 235 – 249, 2004.
- [Tri 97] H. Trinh, and M. Aldeen, A memoryless state observer for discrete time-delay systems, *IEEE Transactions on Automatic Control*, Vo. 42, No. 11 : 1572 – 1577, Nov. 1997.
- [Uri 87] S. Urikura and A. Nagata, Ripple-free deadbeat control for sampled-data systems, *IEEE Transaction on Automatic Control*, Vol. AC-32, No. 6 : 474 – 482, 1987.

- [Vas72] C. Vasseur, *Contribution à l'étude des systèmes échantillonnés commandés par impulsions multimodulées*, Ph.D. dissertation, Université Lille 1, Villeneuve d'Ascq, France, 1972
- [Vid 02] R. Vidal, A. Chiuso and S. Soatto, Observability and identifiability of jump linear systems, *In proceeding of 41nd IEEE conference on decision and control* : 3614 – 3619, Las Vegas NV, USA.
- [Wan 04] H. R. Wang, X. Wang and C. D. Jiang, Adaptive fuzzy friction compensation for X-Y Table,” *Proc. of the 23rd Chinese Control Conference* : 1165-1168, 2004.
- [Wan 05] H. P. Wang, Suivi de trajectoire par la commande modifiée, Master. dissertation, Université Lille 1, Villeneuve d'Ascq, France, Juin, 2005.
- [Wan 06] H. P. Wang, A. Chamroo, C. Vasseur and V. Koncar, Poursuite échantillonnée à partir de l'état retardé et échantillonné, *CIFA'06, 4<sup>e</sup> IEEE Conférence Internationale Francophone d'Automatique*, Bordeaux, France, Mai, 2006.
- [Wan 06a] H. P. Wang, C. Vasseur, A. Chamroo and V. Koncar, Sampled tracking for delayed systems using piecewise functioning controller, *CESA'06, IMACS-IEEE Multiconf. on Computational Engineering in Systems Applications*, Vol. 2 : 1326 – 1333, Beijing, China, Oct. 2006.
- [Wan 07] H. P. Wang, C. Vasseur, A. Chamroo and V. Koncar, Sampled tracking for linear delayed plants using piecewise functioning controller, *WSEAS transactions on systems and control*, Issue 8, Vol. 2 : 405 – 410, August 2007.
- [Wan 07a] H. P Wang, C. Vasseur and V. Koncar, Piecewise continuous systems used in trajectory tracking of a vision based x-y robot, *IEEE CISSE-IETA'07, International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering*, December, 2007.
- [Wan 08] H. P. Wang, A. Chamroo, C. Vasseur, V. Koncar and N. Christov., Trajectory tracking of a vision based x-y robot by Piecewise continuous systems”, *MECATRONICS'08, 7th France-Japan / (5th Europe-Asia) Congress on Mechatronics*, Le Grand-Bornand, FRANCE, May, 2008.
- [Wan 08a] H. P Wang, A. Chamroo, C. Vasseur and V. Koncar, Hybrid control for vision based Cart-inverted Pendulum system, *2008 IEEE American Control Conference*, Seattle, Washington, USA, June 2008.
- [Wan 08b] H. P Wang, A. Chamroo, C. Vasseur and V. Koncar, Stabilization of a 2-DOF inverted pendulum by a low cost visual feedback, *2008 IEEE American Control Conference*, Seattle, Washington, USA, June 2008.
- [Wan 08c] H. P. Wang, A. Chamroo, C. Vasseur, V. Koncar et N. Christov., Suivi de trajectoire avec retour visuel par l'approche des systèmes à fonctionnement par morceaux, *CIFA'08, 5<sup>e</sup> IEEE Conférence Internationale Francophone d'Automatique*, UPB, Bucarest, Septembre 2008.

- [Wan 08d] H. P. Wang, C. Vasseur, V. Koncar et A. Chamroo, “Modélisation et Commande d’un pendule inversé sur un robot X-Y par l’asservissement visuel”, CIFA’08, *5<sup>e</sup> IEEE Conférence Internationale Francophone d’Automatique*, UPB, Bucarest, Septembre 2008.
- [Wan 08e] H.P. Wang, A. Chamroo, C. Vasseur, V. Koncar and N. Christov., A new trajectory tracking approach for a vision based X-Y robot, *Jubilee Collection dedicated to the 60th Anniversary of Mihail Konstantinov*, Sofia, Bulgarian Academy of Sciences (Sous presse)..
- [Wan 08f] H. P Wang, C. Vasseur, V. Koncar et A. Chamroo, Commande d’un pendule inversé 3D par asservissement visuel, *JESA: Journal Européen des Systèmes Automatisés*, No. 9, Octobre 2008.
- [Wei 95] Q. Wei, W. P. Dayawansa and W. S. Levine, Nonlinear controller for an inverted pendulum having restricted travel, *Automatica*, Vol. 31, No. 6 : 841 – 850, 1995.
- [Wen 00] L. Wenzel, N. Vazquez, D. Nair and R. Jamal, Computer vision based inverted pendulum, *Proc. of the 17th IEEE Instrumentation and Measurement Technology Conference*, Vol. 3 : 1319 – 1323, May 2000.
- [Wit 66] H. S. Witsenhausen, A Class of Hybrid-State Continuous-Time Dynamic Systems, *IEEE Transaction on Automatic Control*, Vol. 11, No. 2 : 161 – 167, 1966.
- [Wol 83] W. A. Wolovich, Deadbeat error control of discrete multivariable systems, *International Journal of Control*, Vol. 37 : 567 – 582, 1983.
- [Yam 94] Y. Yamamoto, A function space approach to sampled data control systems and tracking problems, *IEEE Transaction on Automatic Control*, Vol.39, No. 4 : 703 – 713, 1994.
- [Yan 00] R. Yang, Y. Kuen and Z. Li, Stabilization of a 2-DOF spherical pendulum on X-Y table, *Proc. Of the 2000 IEEE International Conference on Control Applications* : 724 – 729, USA, Sep 2000.
- [Ye 98] H. Ye, A. N. Michel and L. Hou, Stability theory for hybrid dynamical systems, *IEEE Transaction on Automatic Control*, Vol. 43, No. 4 : 461 – 474, 1998.
- [Zab 73] J. Zabczyk, Optimal control by means of switching, *Studia Mathematica*, Vol. 65 : 161 – 171, 1973.
- [Zay 01] J. Zaytoon, *Systèmes dynamiques hybrides*, Collection Hermès, Paris France.
- [Zha 03] Y. L. Zhang, J. H. Chung, S. A. Velinsky, Variable structure control of a differentially steered wheeled mobile robot, *Journal of Intelligent and Robotic Systems*, Vol. 36 : 301 – 314, 2003

- [Zha 06] Y. M. Zhang, H. S. Yu and L. Guo, Using Guaranteed Cost Filters for Fault Detection of Discrete-Time Stochastic Distribution Systems with Time Delays, *The Sixth World Congress on Intelligent Control and Automation*, Vol. 2 : 5521 – 5525, Jun 2006.



## **Observation et Commande par Systèmes à Fonctionnement par Morceaux : Application aux Asservissements Visuels**

*Le travail de recherche présenté dans ce mémoire concerne l'étude et le développement de méthodes de linéarisation, d'estimation d'état et de commande pour les systèmes complexes. Le chapitre 1 est consacré au concept et à la classification des systèmes hybrides. L'origine et à la nature des Systèmes à Fonctionnement par Morceaux (SFM) se situent dans ce contexte. Le chapitre 2 traite des contrôleurs de poursuite et de la linéarisation des systèmes réels. Les chapitres 3 et 4, quant à eux, visent à réaliser la poursuite d'une trajectoire à partir d'un retour issu d'un capteur numérique délivrant une information sous forme retardée et échantillonnée. Dans le chapitre 3, un observateur continu par morceaux basé sur la théorie des SFM et sur l'observateur de Luenberger est développé. Le chapitre 4 présente une commande de stabilisation de pendule inversé (2D ou 3D) utilisant deux boucles de retour. A chaque instant d'échantillonnage du capteur, la commande agit dans le plan du pendule, ce qui permet de transformer le problème 3D en un problème 2D. Par rapport aux autres méthodes, les méthodes présentées ont l'avantage d'assurer la stabilité sans utiliser un modèle linéarisé. Dans ce mémoire, chaque méthode proposée est fournie avec des exemples de simulation et d'implantation temps réel.*

**Mots clés :** *Systèmes à dynamiques hybrides, systèmes à fonctionnement par morceaux, systèmes à sortie retardée et échantillonnée, poursuite échantillonnée, linéarisation, observateur continu par morceaux, commande des systèmes complexes, systèmes non linéaires sous actionnés à déphasage non minimal, commande temps réel, asservissement visuel*

---

## **Observation and Control by Piecewise Functioning Systems: Application to Visual Servoing Systems**

*This research work deals with the study and the development of the linearization, state estimation and complex systems' control. Chapter 1 is devoted to the concept and classification of hybrid systems and to the origin and nature of the Piecewise Functioning Systems (PFS). Thus, based on the PFS, chapter 2 proposes controllers for trajectories tracking and linearization of the real time plans. Chapters 3 and 4 aim at realizing the trajectory tracking or stabilization control in the case where the available feedback comes from a digital sensor delivering the plant's output in a delayed and sampled format. In chapter 3 a piecewise continuous observer which is essentially based on PFS theory and on discrete Luenberger observer is developed. In chapter 4, a stabilization control of a 2D or 3D inverted pendulum by using two feedbacks is proposed. At each sensor's sampling period, the control acts in the pendulum's balancing plan, which makes it possible to transform the 3D problem into a 2D one. Compared to other methods, the referred methods guarantee the stability without being based on a linearized model. In this thesis, all proposed methods are illustrated with examples of simulation and real-time implementations.*

**Keywords:** *Dynamic hybrid systems, piecewise functioning systems, delayed and sampled systems, sampled tracking, linearization, piecewise continuous observer, complex systems' control, under actuated non minimum phase mechanical systems, real time control, visual servoing control*