

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE

SCIENTIFIQUE



ECOLE NATIONALE POLYTECHNIQUE

Département d'Automatique



Projet de fin d'études

Pour l'obtention du Diplôme

D'ingénieur d'Etat en Automatique

Thème

**Commandes non linéaires Hybrides d'un pendule  
inversé doté d'une carte Dspace**

Présenté par :

DAHMANE Menad

LOUHAIIDIA Oussama

Proposé et dirigé par :

Pr. BOUDJEMAA

Pr. BOUKHETALA

*Juin 2010*

*Laboratoire de Commandes des Processus  
Ecole Nationale Polytechnique, 10 AV Hassen Badi, BP.182, 16200 El Harrach, Algérie.*

ملخص:

في هذا العمل تم تركيب قوانين تحكم هجينة و غير خطية من خلال نموذج لنواس مقلوب مزود ببطاقة ديسبايس 1104. من أهم أهداف هذا العمل هو تغيير قاعدة التحكم الأولية المجهزة مع النموذج, بقاعدة جديدة أكثر مرونة. في مرحلة أولى نطبق 3 قوانين تحكم مختلفة لرفع النواس من حالته المستقرة إلى حالته المضطربة, و من ثم التحويل إلى القانون الذي يسمح بالإبقاء عليه في حالته الغير المستقرة و ذلك باستعمال مجموعة قوانين مختلفة كالتحكم الضبابي, إنزلاق الحالات, إنزلاق الحالات الضبابي, العصبي الضبابي..

الكلمات الرئيسية:

ControlDesk, dSPACE, النواس المقلوب, أنظمة هجينة, تحكم لاخطي, تحكم ضبابي.

## Résumé

Le travail présenté dans ce mémoire porte sur la synthèse, la simulation puis l'implémentation de lois de commandes non linéaires hybrides sur une maquette pendule inversé munie d'une carte dSPACE. L'un des objectifs principaux était de changer la plateforme initiale de commande (DSP\_TMS320C32/CEMtool) par une autre plus pratique (DSP\_dSPACE/ControlDesk). en premier lieu, on applique une commande de basculement sur le pendule, ensuite on switch vers une autre commande de stabilisation et cela tout en respectant les limites de la course de chariot. A cette fin, nous avons appliqué différentes technique de commandes comme le flou, le glissement, le flou glissant...etc en simulation, et le mode de glissement, en expérimentation.

**Mots clés:** Pendule inversé, commandes hybrides, commande Floue, commandes par modes de glissement, commande neuro floue ANFIS, retour d'état, commande optimale, dSPACE DS 1104, Control Desk.

## Abstract :

The work presented in this memory concerns the synthesis, simulation then the implementation of nonlinear hybrid control laws, on n inverted pendulum systeme. One of the main aims was to change the initial platform (DSP\_TMS320C32/CEMtool) by a new one (DSP\_dSPACE/ControlDesk). Indeed, initially we applied a law to swing-up of pendulum, then another to stabilize it in vertical position, and that respecting the limits of the rail. For this purpose, we applied different technique.

Key words: inverted pendulum, swing-up, hybrid control, Fuzzy control, Sliding mode control, neuro fuzzy ANFIS, state feedback, linear quadratic optimal control, dSPACE, DS1104, Control Desk.

# Remerciements

---

Nous tenons à exprimer nos remerciements à Mr BOUDJEMAA et Mr BOUKHETALA pour les orientations qu'ils nous ont donnés, ainsi que Mr BENMANSOUR pour son aide. Nous remercions également tous les professeurs du département de l'Automatique de l'ENP.

# Table des matières

Préface	01
Introduction générale	02
<b>CHAPITRE I : État de l'art</b>	<b>06</b>
<hr/>	
Introduction	07
1. Les Systèmes mécaniques sous actionnée	08
2. Les Commandes Hybrides	10
3. Intérêt de l'étude d'un pendule inversé	13
<b>CHAPITRE II : Élaboration du modèle</b>	<b>16</b>
<hr/>	
Introduction	17
1. Modélisation des Systèmes Mécaniques	17
2. Modélisation par Approche Lagrangienne du Pendule Inverse	17
2.1. Modèle avec action Force	17
2.2. Model avec action Tension	21
3. Simulation en Boucle Ouverte du Model	23
<b>CHAPITRE III : Rappels théoriques</b>	<b>25</b>
<hr/>	
Introduction	26
1. La commande Floue et optimisation par réseau neuro-floue hybride	26
1.1. Introduction à la logique floue	26
1.2. Les ensembles flous dans la théorie de la commande	27
1.3. Le Contrôleur Flou	28
1.4. Optimisation Neuronal des réseaux d'inférences flous, illustration de l'algorithme hybride ANFIS et de l'outil anfisedit	30
1.4.1. Illustration de l'algorithme ANFIS	31
1.4.2. L'outil 'anfisedit'	37
2. La commande par Modes de Glissement et Optimisation par Logique Floue	38
2.1. La commande par MDG	38
2.2. Le phénomène de réticence ( <i>chattering</i> )	42
2.3. Elimination du phénomène du <i>Chattering</i> avec contrainte sur la robustesse	42
2.4. Optimisation du MDG par Correction floue de la couche limite de commande	43

## **CHAPITRE IV: Synthèse des lois de commandes**

47

---

Introduction	48
1. Commande de balancement vers le haut	48
1.1. Le balancement du bras par retour d'état non linéaire	48
1.2. Le balancement du bras par régulation flou	51
2. Commandes de Stabilisation Locale	54
2.1. Commande Standard	54
2.1.1. Commande par retour d'état linéaire	54
2.1.2. Commande par Régulation Floue a deux Entrées	57
2.1.3. Stabilisation par régulation floue à quartes entrées	61
2.1.4. Stabilisation par modes de glissement	64
2.2. Commande Hybride	69
2.2.1. Stabilisation par mode de glissement flou	69
2.2.2. Optimisation NeuroFloue type ANFIS	73
a. Optimisation du régulateur par modes de glissements	74
b. Optimisation du régulateur floue découplé	76
3. Commandes final à expérimenter	78

## **CHAPITRE V : Implémentation et résultats expérimentaux**

80

---

Introduction	81
1. Présentation du système	81
1.1. Constitution du pendule inverse	82
a. La chaîne d'actionneur	83
b. La chaîne d'instrumentation	84
1.2. La Carte dSPACE1104	86
1.3. Le boîtier de connexion PS600	89
1.4. Le logiciel ControlDesk	90
2. Mise au point du nouveau système de commande	95
2.1. Identification des Pins	95
2.2. Branchement des Pins	96
3. Résultats expérimentaux	97
3.1. Expérimentation du système réel en BO et validation du modèle	98
3.2. Implémentation de la commande par modes glissant avec et sans action intégrale et léviation par action résonante	99

---

Conclusion générale

101

## Liste des Figures

<b>Figure I.1</b>	Un robot marcheur de Honda	14
<b>Figure I.2</b>	Chaise et Segway	14
<b>Figure I.3</b>	Pendule Gyroscopique inversé	15
<b>Figure II.1</b>	Schéma du pendule inverse	18
<b>Figure II.2</b>	Résultat de simulation du système libre en BO	24
<b>Figure II.3</b>	Résultats de simulation du système forcé en BO	24
<b>Figure III.1</b>	Fonction d'appartenance a la classe des grands hommes	27
<b>Figure III.2</b>	Architecture d'un contrôleur par logique floue	28
<b>Figure III.3</b>	Exemple de fuzzyfication	29
<b>Figure III.4</b>	Architecture d'une commande ANFIS	32
<b>Figure III.5</b>	Exemple ANFIS à 2 entrées avec 9 règles	34
<b>Figure III.6</b>	Méthode d'apprentissage Hybride	36
<b>Figure III.7</b>	GUI anfisedit	38
<b>Figure III.8</b>	Principe des modes glissants	39
<b>Figure III.9</b>	Les étapes de glissement	40
<b>Figure III.10</b>	Phénomène de réticence	42
<b>Figure III.11</b>	Fonction SAT avec un seuil et deux seuils (zone morte)	43
<b>Figure III.12</b>	Fonction « Smooth »	43
<b>Figure III.13</b>	Fuzzy Boundary Layer (Couche limite Floue)	45
<b>Figure III.14</b>	Calculateur flou de $\phi$	45
<b>Figure III.15</b>	Schéma Simulink du régulateur flou glissant	45
<b>Figure III.16</b>	Fonctions d'appartenance des entrées (haut), la sortie floue (bas)	46
<b>Figure IV.1</b>	Les différents niveaux énergétiques de la trajectoire d'état du pendule	49
<b>Figure IV.2</b>	Simulation du balancement par retour d'état non linéaire	51
<b>Figure IV.3</b>	Base des fonctions d'appartenance du régulateur floue de balancement	52
<b>Figure IV.4</b>	Simulation du balancement par logique floue	53
<b>Figure IV.5</b>	Simulations pour $\theta_0 = 0.1rad$ avec retour d'état	55
<b>Figure IV.6</b>	Simulations pour $\theta_0 = 0.1rad$ avec retour d'état optimal	56
<b>Figure IV.7</b>	Simulations avec retour d'état et perturbation impulsionnelle	56
<b>Figure IV.8</b>	Simulations pour $\theta_0 = 0.1rad$ avec variations paramétriques ( $l=60cm$ )	57
<b>Figure IV.9</b>	Surface de commande de $\theta$	58
<b>Figure IV.10</b>	Fonction d'appartenance décrivant un régulateur floue Mamdani	59
<b>Figure IV.11</b>	Simulation d'une régulation floue à deux entrées avec $\theta_0 = 0.2rad$	59
<b>Figure IV.12</b>	Régulation Flou avec Perturbation impulsionnelle sur le pendule	60
<b>Figure IV.13</b>	Flou avec variation paramétrique sur la longueur du pendule	60
<b>Figure IV.14</b>	Base de donnée du régulateur quatre entrées	61
<b>Figure IV.15</b>	Surface de commande de $x$	62
<b>Figure IV.16</b>	Régulation floue 4 entrées $\theta_0 = 0.2rad$	62

<b>Figure IV.17</b>	Flou variation paramétrique de 50% sur la longueur du pendule	63
<b>Figure IV.18</b>	Simulation de poursuite en x	63
<b>Figure IV.19</b>	Simulation des MDG pour $\lambda$ par retour d'état avec $\theta_0 = 0.1rad$	66
<b>Figure IV.20</b>	Simulation pour $\lambda$ par retour d'état optimal en trajectoire $\theta_0 = 0.1rad$	66
<b>Figure IV.21</b>	Mdg, Simulation du système perturbé par force de 1N avec	67
<b>Figure IV.22</b>	Mdg, Simulations Pour variations paramétriques de 50% sur la longueur	67
<b>Figure IV.23</b>	Mdg, Simulation de la poursuite en x, avec retour d'état optimal	68
<b>Figure IV.24</b>	Simulations du mode de glissement flou pour $\theta_0 = 0.3rad$	70
<b>Figure IV.25</b>	Simulation pour rejet de perturbation impulsionnelle d'amplitude	70
<b>Figure IV.26</b>	Simulations pour variations paramétriques de 50% sur $l$	71
<b>Figure IV.27</b>	Simulations des VPs pour différent gains de pondération du traitement flou de la surface	71
<b>Figure IV.28</b>	Simulation des VPs pour $\lambda$ par retour d'état optimal	72
<b>Figure IV.29</b>	Simulation pour poursuite de x d'une dent de scie	72
<b>Figure IV.30</b>	Schéma Simulink montrant la méthode d'acquisition des données	73
<b>Figure IV.31</b>	Simulation pour neuro- floue avec ROI=0.3	75
<b>Figure IV.32</b>	Le réseau d'inférence floue neuronale à 4 entrées pour ROI=0.3	75
<b>Figure IV.33</b>	résultats obtenue pour l'optimisation d'un régulateur floue découplé	77
<b>Figure IV.34</b>	Structure des régulateurs ANFIS utilisé	77
<b>Figure IV.35</b>	Simulation de la léviation par action résonante puis stabilisation par mode de glissement simple	78
<b>Figure IV.36</b>	Simulation de la léviation par action résonante puis stabilisation par mode de glissement avec action intégrale sur le déplacement	79
<b>Figure IV.37</b>	Résultats de simulation de la commande finale	79
<b>Figure V.1</b>	Maquette du nouveau système pendule inversé	81
<b>Figure V.2</b>	Schéma de la maquette pendule inversé	82
<b>Figure V.3</b>	Constitution du pendule inversé	82
<b>Figure V.4</b>	Boitier de contrôle avec illustration de fonctions	83
<b>Figure V.6</b>	Encodeur incrémental	85
<b>Figure V.7</b>	Principe de fonctionnement d'encodeur incrémental.	85
<b>Figure V.8</b>	Carte dSPACE 1104.	86
<b>Figure V.9</b>	Architecture du DS 1104	87
<b>Figure V.10</b>	Exemple de blocs utilisés pour la conception du schéma Simulink	88
<b>Figure V.11</b>	schéma et boitier PS600.	89
<b>Figure V.12</b>	Exemple d'IHM développée sous Control Desk.	90
<b>Figure V.13</b>	Exemple de chargement d'une application dans la carte dSPACE	91
<b>Figure V.14</b>	Identification des pins	96
<b>Figure V.15</b>	Branchement de la commande (A) et les mesures (B)	96
<b>Figure V.16</b>	Schéma de câblage entre la dSPACE et le système	97
<b>Figure V.17</b>	Blocks Simulink d'acquisition de données	97
<b>Figure V.18</b>	Simulation du système réel en BO	98

## Liste des tableaux

<b>Tableau II.1</b>	Paramètres physique du système mathématique	23
<b>Tableau Erreur ! Il n'y a pas de texte répondant à ce style dans ce document..1</b>	de décision standard de Mc.Vicar-Whelan à 5 ensembles flous	Table 29
<b>Tableau III.2</b>	Les différentes couches d'un système ANFIS	34
<b>Tableau III.2</b>	Les paramètres à ajuster d'un système ANFIS	36
<b>Tableau IV.1</b>	Règles de régulateur flou de balancement	52
<b>Tableau IV.2</b>	Règle d'inférence floue d'un régulateur 2 entrées	58
<b>Tableau IV.3</b>	Règles du régulateur flou du déplacement pour quatre entrées	61
<b>Tableau IV.4</b>	Données associées a différent essais de simulations	74
<b>Tableau IV.5</b>	Résultats d'apprentissage pour différente sphères d'influence pour le Régulateur flou numéro 1 pour la régulation du déplacement	76
<b>Tableau IV.6</b>	Résultats d'apprentissage pour différente sphères d'influence pour le régulateur flou numéro 2 pour la régulation de l'inclinaison	76
<b>Tableau V.1</b>	Boutons de la boite de contrôle et de supervision du pendule	84
<b>Tableau V.2</b>	Spécifications techniques du système	86
<b>Tableau V.3</b>	L'identification des Pins	95



# Introduction générale

---

Les différents progrès technologiques que connaît le monde ces dernières années, dans tous les domaines, ont donné un essor considérable à l'automatique ; et l'élaboration de lois de commande de systèmes a fait l'objet de nombreux travaux de recherche. Néanmoins, les systèmes à commander sont devenus de plus en plus complexes et les lois de commandes deviennent elles même complexes et difficiles à mettre en œuvre directement sur le système réel.

En effet, les avancées récentes en informatique et en électronique de puissance permettent la fabrication des systèmes prototypes, qui jouent le rôle de simulateurs réels miniatures. Ces derniers permettent de tester de nouvelles techniques de commande, à l'instar du pendule inversé qu'on dispose au niveau de laboratoire LCP de l'ENP.

Le pendule inversé est bien connu pour l'évaluation, la validation et la vérification de divers types de théories de commande des systèmes mécaniques, car ce système présente des caractéristiques très intéressantes de fait qu'il est non linéaire et sous actionné ; par conséquent il est difficile de le commander par un contrôleur temps réel ; et les méthodes conventionnelles ont montrés leurs limites en termes de stabilisation. En général, un pendule inversé se présente sous plusieurs formes; comme le pendule simple, le double pendule...etc. Différentes méthodes de commandes, standard ou avancé, sont proposées dans la littérature comme par exemples : le retour d'état, commande à énergie minimale, commande par modes de glissant, commande par logique floue, commande robuste, commande linéarisante... et les commandes hybrides pour lesquelles on a opté dans ce mémoire à savoir le mode glissant flou et le neuronal flou. Malgré l'existence de toutes ces méthodes, commander un pendule inversé reste un sujet de recherche très intéressant. Et la solution de ce problème est applicable pour d'autres systèmes non linéaires et sous actionnés dans différents domaines.

Durant notre projet, nous nous sommes particulièrement intéressés à la mise au point d'un nouveau système de commande du banc d'essai dont on dispose, et la configuration de ses différentes Entrées/Sorties sur Matlab/Simulink. En effet, l'ancien système de commande présente beaucoup d'insuffisance et il ne permet pas l'implémentation de lois commandes modernes.

Dans ce mémoire, nous nous sommes intéressés à des commandes hybrides du fait que celles-ci nous permettent de commander le pendule inverse, en éliminant les insuffisances des commandes standards. En effet, Dans une première phase on va mettre en œuvre des commandes standards dans le but de stabiliser le pendule dans le voisinage de l'équilibre instable; et dans une seconde phase on envisage de synthétiser des commandes hybrides pour le même objectif. Par la suite, on va mettre au point le système en effectuant les différents branchements et en configurant ses différentes entrées/sorties sur Matlab/Simulink et puis on va implémenter les commandes déjà synthétisées sur ce même système.

Pour cela nous avons organisé ce mémoire en cinq chapitres comme suit:

**Le premier chapitre** est consacré à l'état de l'art où nous avons parlé des systèmes sous actionnés en générale puisque le pendule inversé appartient à cette classe des systèmes en plus on a donné d'autres exemples de ce genre des systèmes, et enfin nous avons illustré l'intérêt de l'étude du pendule inversé à travers des exemples d'application de son principe dans différents domaines.

**Le second chapitre** porte sur la modélisation de pendule inversé et la simulation en boucle ouverte et montrer ainsi la nature instable de ce système.

**Le troisième chapitre** est consacré à des rappels théoriques des méthodes de commandes utilisées. Nous avons divisé ce chapitre en deux parties : la première est un rappel sur les commandes standards (non hybrides) comme la commande floue ou on a parlé sur sa théorie et les différentes étapes pour la construction d'un régulateur flou et la commande par modes glissants et enfin la synthèse par Lyapunov. La seconde partie est consacrée à la présentation des différentes méthodes d'hybridation des techniques de commandes et les gains qu'on y peut tirer, comme la technique d'hybridation floue-glissant pour éliminer le phénomène indésirable de réticence qui caractérise les modes glissants, puis la technique d'optimisation du système d'inférence de du régulateur flou par de réseaux de neurones ce qu'on appel **ANFIS**.

**Le quatrième chapitre** est consacré à la synthèse des lois de commandes hybrides que nous avons exposées au chapitre précédent puis la simulation et l'interprétation des graphes résultants.

Dans le **dernier chapitre** et le plus intéressant dans ce travail, on va résumer les principales étapes de mise au point du nouveau système de commande utilisant la carte dSPACE DS1104 puis la configuration des entrées/ sorties, ensuite on passe à l'implémentation lois de commandes que nous avons simulées dans le chapitre précédent.

Enfin, une conclusion générale résumant et évaluant les résultats trouvés dans le cadre de ce mémoire sera donnée. Elle inclura aussi les perspectives envisagées.

# Chapitre I

---

État de l'art sur les commandes  
hybrides des Systèmes sous actionnés

## Introduction

Dans ce chapitre premier, on donnera les grandes lignes du travail à venir, on commence par donner une idée sur ce que sont les systèmes sous actionnée, puis après on verra les commandes hybrides, en générale et leur classement, finalement on donne une idée sur l'intérêt d'étudier les pendule inverse.

### 1. Les Systèmes mécaniques sous actionnés

La commande des systèmes mécaniques sous-actionnés constitue un domaine passionnant pour les automaticiens, présentant plusieurs types de problèmes, et pouvant chacun faire appel à des techniques de contrôles différentes. A titre d'exemple on peut citer :

- La stabilisation locale en un point d'équilibre. Elle consiste à stabiliser la partie non actionnée en sa position d'équilibre instable, et la partie actionnée en une position finale désirée partant de conditions initiales suffisamment proche de l'équilibre désirée.

- La stabilisation orbitale où le système en boucle fermée auto-génère une orbite périodique.

- La stabilisation globale qui consiste à amener la partie non actionnée de n'importe quelles conditions initiales à la position d'équilibre instable et la partie actionnée à une position finale désirée.

- La poursuite de trajectoire qui consiste, par exemple, à maintenir la partie non actionnée proche de la position d'équilibre instable et à forcer la partie actionnée à suivre une trajectoire donnée. On peut imaginer que le pendule inversé fait le rôle d'un serveur. Il (le pendule) a pour mission de se déplacer d'un point A à un point B tout en maintenant un plateau (posé au bout de la tige) en équilibre.

Lorsqu'on parle de stabilisation locale, on pense habituellement à une linéarisation du système autour de l'équilibre désiré suivie de la synthèse d'un retour d'état statique linéaire. Cette démarche a deux désavantages :

- La commande synthétisée à la base des équations linéarisé a un domaine d'attraction restreint.

– La présence inévitable des forces de frottements engendre si on utilise un retour d'état statique linéaire, des cycles limites se traduisant par des oscillations du système autour de la position d'équilibre désirée.

L'étude des systèmes mécaniques sous-actionnés est beaucoup plus récente. On peut citer le pendule inversé, le pendubot, comme exemples de systèmes mécaniques sous-actionnés. Inspirés principalement de la structure du corps humain, les systèmes sous-actionnés admettent des degrés de liberté plus nombreux que les actionneurs. Le manque d'actionneur complique la tâche de commande de ce genre de systèmes : à notre connaissance, à part la commande par retour d'état basé sur un linéarisé du système, il existe peu de stratégies de commande conçues à la base des équations non linéaires assurant la stabilisation autour du point d'équilibre instable.

La commande de tels systèmes, lorsqu'on veut la mettre en pratique, se trouve confrontée à la présence inévitable de perturbations, notamment celles issues des phénomènes de frottements. Les forces de frottement ont des dynamiques compliquées souvent mal modélisées. Pour commander des systèmes perturbés mal modélisés, des techniques de commande à structure variables (par modes glissants) sont efficaces à condition que les perturbations et frottements vérifient une condition dite de recouvrement (matching condition en anglais).

Un système mécanique est dit complètement actionné, si le nombre des entrées de commande est égal au nombre de degrés de liberté, Par conséquent, les systèmes mécaniques complètement actionnés sont linéarisables par retour d'état statique.

Un système mécanique est dit sous-actionné s'il admet moins d'actionneurs que de degrés de liberté, soit :  $rg(B(q)) = m < n$ . Cette restriction empêche une linéarisation par bouclage statique de la dynamique complète du système.

La forme générale des équations dynamiques des systèmes sous-actionnés peut être donnée par :

$$\begin{aligned} M_{11}(q)\ddot{q}_1 + M_{12}(q)\ddot{q}_2 + C_1(q, \dot{q}) + G_1(q) &= 0 \\ M_{21}(q)\ddot{q}_1 + M_{22}(q)\ddot{q}_2 + C_2(q, \dot{q}) + G_2(q) &= B(q)u \end{aligned}$$

Dans ce mémoire ; on s'intéresse bien évidemment sur le pendule inversé, mais cela n'empêche pas de donner quelques autres exemples de cette famille de systèmes sous actionnés.

#### a) Le pendubot

Le pendubot est constitué de deux tiges qui peuvent tourner autour de leurs axes respectifs. La tige 1, de masse  $m_1$  et de longueur  $l_1$ , est actionnée par un couple de contrôle  $\tau$  tandis que la tige 2, de masse  $m_2$  et de longueur  $l_2$ , est en rotation libre autour de la tige 1. A vitesse nulle, le pendubot admet une infinité de points d'équilibre instables donnés par  $\dot{q}_1 = \dot{q}_2 = \dot{q}_2 = 0; q_1 = cst$  correspondant à la position haute de la tige 2 pour toute position de la première tige, et une infinité de points d'équilibre stables donnés par  $\dot{q}_1 = \dot{q}_2 = 0; q_2 = \pi$  correspondant à la position basse de la tige numéro 2.

#### b) L'acrobot

L'acrobot est similaire au pendubot à la différence que c'est l'articulation joignant les 2 tiges qui est actionnée par un couple  $\tau$ .

#### c) Le pendule de Furuta

Le pendule de Furuta a été conçu par K. Furuta pour contrer le handicap de la course limitée (en translation horizontale) du pendule inversé classique. Le bras actionné en rotation dans le plan horizontal permet une course infinie, ce qui facilite la conception du contrôle. A l'autre extrémité du bras vient s'ajouter un pendule libre en rotation dans le plan vertical orthogonal au bras.

#### d) Le pendule à roue inertielle

Le pendule à roue inertielle, représenté sur la, est constitué d'un pendule libre en rotation autour d'un axe lié au sol, l'autre extrémité du pendule étant reliée à un disque actionné qui ne peut que tourner.



### **e)Le pendule inversé**

Le pendule inversé est un système classique très intéressant et largement étudié dans la communauté automatique, vu sa nature non linéaire et instable. Il a toujours constitué un défi intéressant pour le contrôle et a servi à la compréhension des notions de l'automatique comme à l'élaboration des lois de commande. Par ailleurs, son principe se retrouve dans plusieurs applications (par exemple le Segway, ou les fusées). Depuis 1950, les pendules inversés sont des plates-formes d'expérimentation classiques dans les laboratoires d'automatique. Ils ont été utilisés pour illustrer des idées de commande linéaire comme la stabilisation des systèmes instables. Les pendules sont aussi utilisés pour illustrer des idées émergentes du contrôle non linéaire. Des exemples typiques sont la stabilisation par retour de sortie, la commande par passivité, les techniques backstepping et forwarding et la commande robuste à structure variable. Plusieurs études de cas ("task oriented controls") ont concerné le balancement de bas en haut et le rattrapage. C'est justement sur ce dernier titre qu'on va faire notre étude de simulation et d'implémentation de lois de commandes dans la suite de ce mémoire.

## **2. Les Commandes Hybrides**

L'une des tâches de base de l'automaticien est de développer des régulateurs pour piloter des systèmes. En fonction des erreurs de sortie des systèmes (qui sont les variables d'observation), l'organe de commande génère automatiquement les entrées du système (variables de commande) de façon à obtenir le comportement souhaité des sorties du processus.

L'augmentation des capacités de calcul des ordinateurs a conduit à l'intégration de nouvelles fonctions plus complexes, combinant plusieurs techniques de base, afin d'obtenir des résultats plus performants et plus fiables. Le problème global posé à l'automaticien est de gérer des processus complexes en développant diverses fonctions automatisées, qui à la base présentent des insuffisances. Des structures plus complexes

sont alors considérés, se sont les systèmes hybrides de control où ce qu'on appelle l'Intelligence Artificielle (IA). [1]

En effet la pluparts des approches conventionnelles (classiques) utilisent parfois des hypothèses trop restrictives, par exemple, la nécessité d'un modèle mathématique précis du système régulé. Ainsi, si la structure du modèle choisie pour un problème de régulation adaptative de types *self tuning* n'est pas suffisamment proche de la réalité, la méthode d'adaptation ne fonctionnera pas. Tous les problèmes de régulation ne peuvent donc pas être résolus efficacement, en particulier ceux pour lesquels la modélisation analytique du procédé est difficile, voir impossible ; or, ces cas de figure se rencontrent pour beaucoup de systèmes industriels. De façon générale, les méthodes conventionnelles sont précises, toutefois, le codage procédural ne permet pas beaucoup de généralisation, il est spécifique à une application donnée et hautement spécialisé. Des limites apparaissent donc clairement pour l'intégration des fonctions d'automatisation avancée avec seulement des fonctions mathématiques.

Le deuxième type de limitations des approches classiques est le nombre réduit de degrés de liberté de certaines lois de commande, ce qui les rend surtout efficaces pour les systèmes linéaires, un exemple est le régulateur PID, qui est très largement utilisé pour les applications industrielles (à 90 %). Il possède l'avantage d'être facile à utiliser et sûr pour des conditions de fonctionnements nominales ; mais en contrepartie, il possède la limitation majeure d'être réglé sous-optimalement dans la pluparts des autres conditions de fonctionnement (telles des variations importantes de facteurs de la boucles de régulation – ces facteurs pouvant être des dynamique non modélisées et très influentes du système, du bruit ou le niveau de consigne). Des ajustements supplémentaires sont alors souvent nécessaires dans les nouveaux contextes cités précédemment, mais aussi, parfois, après la phase de réglage des gains du PID. En effet, les méthodes de réglages couramment utilisées, comme les formules de *Ziegler & Nichols*, conduisent à des résultats souvent inacceptables en pratique. C'est pourquoi, dans l'industrie, plus de 50% des régulations sont menés en mode manuel car les

opérateurs sont insatisfaits de performances atteintes après l'application de procédure d'autorégulation.

En plus, certaines méthodes conventionnelles sont parfois difficiles à utiliser et trouvent peu de succès auprès des industriels, qui ne disposent pas toujours des compétences nécessaires. Il s'agit surtout de techniques de régulation des systèmes non linéaires; ces régulateurs, possédant de nombreux paramètres, sont flexibles. Mais, pour les régulateurs conventionnels, flexibilité et transparence (clarté des liens entre les paramètres de régulateur et performances en boucle fermée) s'opposent souvent.

Comme les approches conventionnelles possédant certaines limitations, il est intéressant de se tourner vers d'autres approches, telles que les techniques appelées « non-conventionnelles ». Ces outils regroupent les domaines de l'intelligence artificielle -incluant les techniques floues, les approches neuronales et les méthodes de raisonnement probabiliste incluant les algorithmes génétiques.

C'est dans ce cadre que se situe le travail que nous présentons dans ce mémoire. En effet, nous nous sommes particulièrement intéressés à des commandes hybrides du fait que celles-ci nous permettent de commander le pendule inverse, en éliminant les insuffisances des commandes standards. En effet, Dans une première phase on va mettre en œuvre des commandes standards dans le but de stabiliser le pendule dans le voisinage de l'équilibre instable; et dans une seconde phase on envisage de synthétiser des commandes hybrides pour le même objectif.

Plusieurs classes d'hybridation de commandes existent. Globalement on pourrait les classées dans trois classes distincts, les commandes hybrides continue-continue où une technique ou plusieurs optimisent une autre. Dans la seconde classes, on trouve les hybridations continue-discrète où un réseau de Petrie organisera les enchainements de commandes, en fonction de l'état du système. Et puis finalement, la commande multi-modèles, ou autour de chaque point d'équilibre, on construit un model linéarisé et on

lui associe une commande locale, et finalement la commande totale sera le centre de gravité de toutes les commandes locales.

### 3. Intérêt de l'étude d'un pendule inversé [2]

L'intérêt en ingénierie de l'étude du pendule inverse sont diverse et remarquables.

#### 3.1. En médecine

L'homme est vu comme un double pendule inversé en série, dont les deux axes de rotations principales sont les chevilles et les hanches. Quand nous sommes en position debout, nos articulations travaillent sans arrêt pour nous y maintenir. Les spécialistes qui travaillent à la réalisation de prothèses (remplacement chirurgical d'un organe ; la pièce ou l'appareil de remplacement : prothèse dentaire) pour les hanches sont amenés à utiliser le modèle de double pendule inversé pour calculer l'ensemble des contraintes qui sont soumises à la prothèse, comme le montre la figure suivante. Le premier pendule est articulé à la cheville et représente les membres inférieurs considérés groupés. Le second pendule est articulé à la hanche et représente la partie supérieure du corps. On accélère en se penchant en avant et on ralentit en se penchant en arrière. Le système est le même que le pendule inversé.

#### 3.2. En robotique

Dans le même ordre d'idée, un robot est vu comme un pendule inversé. Pour maintenir le robot en équilibre il suffit de stabiliser le pendule inversé correspondant à sa position verticale au cours de son déplacement et en présence d'autres perturbations. A l'occasion, nous citons quelques exemples des robots qui sont déjà étudiés et conçus à base de ce principe:

- **Les robots marcheurs** : peut se présenter par deux doubles pendules inversés en série, dont les deux pendules sont couplés par une tige avec deux articulations.



*Figure I.1 : Un robot marcheur de Honda.*

- **La chaise sur deux roues/Le Segway:** elles sont vues comme un pendule inversé simple (une seule tige) se déplaçant à l'aide de deux roues.



*Figure I.2 : Chaise et Segway.*

### 3.3. Dans l'aérospatiale

Dans ce domaine aussi, l'étude des pendules inversés a une grande importance, par exemple pour commander et stabiliser l'attitude du satellite, le lancement des fusées...etc.

- a) Pour stabiliser l'attitude d'un satellite par les actionneurs gyroscopiques

Les actionneurs gyroscopiques également appelés girodynes (CMG : Control Moment-Gyro) constituent de nouveaux systèmes d'actionneurs qui permettent de générer des couples dynamiques de commande de basculement d'attitude d'un satellite. L'expérience consiste à le rappeler vers la position d'équilibre (instable), il est donc vu comme un pendule inversé.



*Figure I.3 : pendule gyroscopique inversé.*

b) Pour le lancement des fusées

Lors de son lancement, une fusée est vue comme un pendule inversé simple, donc il faut maintenir sa position verticale par rapport à la terre, et pour cela, des moteurs à propulsion sont installés dans les deux cotés de la fusée (gauche et droite).

### **3.4. Dans la recherche**

Vu sa complexité, le pendule inversé est considéré comme l'un des meilleurs systèmes pour tester des lois de commande modernes.

# Chapitre II

---

Élaboration du modèle

## Introduction

Le système simple pendule inversé est un système multi-variable non linéaire instable avec des constantes de temps très rapides. Ce système est reconnu pour être une plateforme difficilement stabilisable. Le problème de commande exige la recherche du modèle mathématique du système à commander, donc une phase de modélisation est nécessaire pour permettre l'étude en simulation. Pour ce faire, on trouve dans la littérature différentes méthodes comme celle basée sur la physique Newtonienne ou celle basé sur le formalisme de Lagrange. Nous avons optés pour la méthode de Lagrange-Euler

### 1. Modélisation des Systèmes Mécaniques

Afin de pouvoir modéliser un système mécanique par des équations mathématiques, on commence par choisir un ensemble de coordonnées permettant de le décrire. Ces coordonnées sont généralement nommées coordonnées généralisées. Cette dénomination provient de l'époque où l'utilisation des coordonnées cartésiennes était la plus naturelle.

L'intérêt du choix des coordonnées généralisées réside dans la simplification de la démarche de la modélisation puis de l'analyse de ces systèmes. Une fois les coordonnées généralisées choisies, on peut procéder de deux manières :

- Soit on fait une étude des forces agissant sur le système, et on applique la seconde loi de la dynamique de Newton, disant que la somme des forces appliquées est égale à la masse multipliée par l'accélération.

- Soit on étudie les différentes énergies échangées par le système, puis on exprime le lagrangien avant d'appliquer la formule d'Euler-Lagrange.

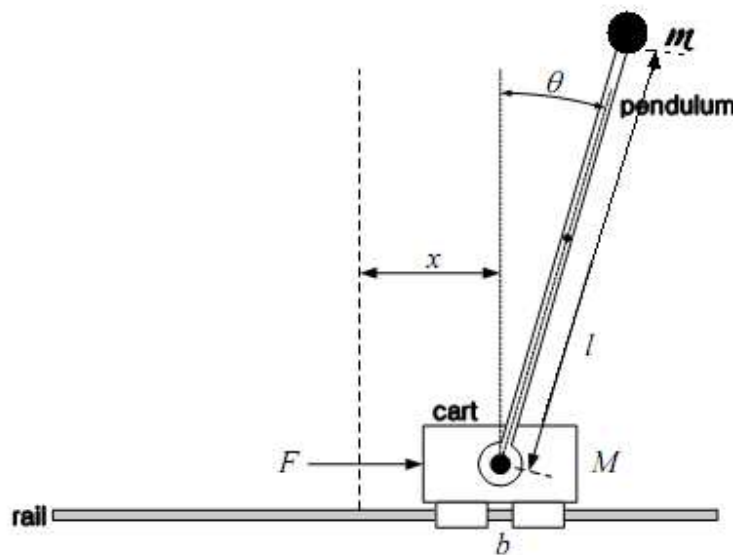
Cette deuxième méthode, en général plus simple à mettre en œuvre, est décrite dans ce qui suit.



## 2. Modélisation par Approche Lagrangienne du Pendule Inverse

### 2.1. Modèle avec action Force

Le système pendule inversé est composé comme indique la figure III.1 d'un chariot mobile en translation sur un axe horizontal et d'un pendule libre et fixé verticalement sur le chariot. En exerçant une force horizontale  $\vec{F}(t)$  sur le chariot, il résulte une translation de  $x$  mètres de celui-ci ainsi que une rotation de  $\theta$  radian du pendule. Le système simple pendule inversé est un système instable en boucle ouverte, non linéaire multi variable et sous actionné.



**Figure II.1 :** Schéma du pendule inverse

Pour le schéma de la Figure1 on aura :

$$T = T_c + T_p$$

$$T_c = \frac{1}{2} m_c \dot{x}^2$$

$$T_p = \frac{1}{2} m_p v_p^2$$

$$\vec{v}_p = \dot{x} \vec{i} + l \dot{\theta} \vec{j}$$

$$v_p^2 = \dot{x}^2 + l^2 \dot{\theta}^2 + 2l\dot{x}\dot{\theta}\vec{i}\cdot\vec{j}$$

$$\vec{i}\cdot\vec{j} = \cos(\theta)$$

$$v_p^2 = \dot{x}^2 + l^2 \dot{\theta}^2 + 2l\dot{x}\dot{\theta}\cos(\theta)$$

$$T_p = \frac{1}{2}m_p\dot{x}^2 + \frac{1}{2}m_p l^2 \dot{\theta}^2 + m_p l\dot{x}\dot{\theta}\cos(\theta)$$

Finalement :

$$T = \frac{1}{2}m_c\dot{x}^2 + \frac{1}{2}m_p\dot{x}^2 + \frac{1}{2}m_p l^2 \dot{\theta}^2 + m_p l\dot{x}\dot{\theta}\cos(\theta) \quad II.1$$

Concernant l'énergie potentielle :

$$U = U_c + U_p$$

Avec un chariot à déplacement horizontal uniquement, et un pendule à énergie nulle dans sa position la moins élevée on aboutit à l'équation :

$$U = U_p = m_p gh = m_p gl \cdot (\cos(\theta) - 1)$$

Alors :

$$L = T - U$$

$$L = \frac{1}{2}m_c\dot{x}^2 + \frac{1}{2}m_p\dot{x}^2 + \frac{1}{2}m_p l^2 \dot{\theta}^2 + m_p l\dot{x}\dot{\theta}\cos(\theta) - m_p gl \cdot \cos(\theta) + cst$$

Quelque simplification et on aura :

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_p l^2 \dot{\theta}^2 - m_p l (g - \dot{x}\dot{\theta}) \cos(\theta) + cst \quad (II.2)$$

Où :  $M = m_p + m_c$

On considérant des frottements de type visqueux entre le chariot et ses rails, et avec une liaison pivot parfaitement lisse on écrit :

$$D = \frac{1}{2}\alpha\dot{x}^2$$

Les équations dynamiques de Lagrange du système seront :

$$\begin{cases} \frac{d}{dt} \left[ \frac{\delta L}{\delta \dot{x}} \right] + \frac{\delta D}{\delta \dot{x}} - \frac{\delta L}{\delta x} = F_x \\ \frac{d}{dt} \left[ \frac{\delta L}{\delta \dot{\theta}} \right] + \frac{\delta D}{\delta \dot{\theta}} - \frac{\delta L}{\delta \theta} = F_\theta \end{cases}$$

Alors :

$$\begin{cases} M\ddot{x} - \frac{d}{dt} [m_p l \dot{\theta} \cos(\theta)] + \alpha \dot{x} = F_x \\ m_p l^2 \ddot{\theta} - \frac{d}{dt} [m_p l \dot{x} \cos(\theta)] - m_p l (g + \dot{x} \dot{\theta}) \sin(\theta) = F_\theta \end{cases} \quad (II.3)$$

Donc on réaménageant les deux équations précédentes, et sachant qu'aucune force de commande n'est appliquée directement sur le pendule, et on faisant la simplification au niveau de la 2<sup>nd</sup> équation on aboutira à ce nouveau system :

$$\begin{cases} M\ddot{x} + m_p l \ddot{\theta} \cos(\theta) - m_p l \dot{\theta}^2 \sin(\theta) + \alpha \dot{x} = F \\ m_p l^2 \ddot{\theta} + m_p l \ddot{x} \cos(\theta) - m_p l \dot{x} \dot{\theta} \sin(\theta) - m_p l (g - \dot{x} \dot{\theta}) \sin(\theta) = 0 \end{cases} \quad (II.4)$$

D'où :

$$\begin{cases} M\ddot{x} + m_p l \ddot{\theta} \cos(\theta) - m_p l \dot{\theta}^2 \sin(\theta) + \alpha \dot{x} = F \\ l \ddot{\theta} + \ddot{x} \cos(\theta) - g \sin(\theta) = 0 \end{cases} \quad (II.5)$$

L'équation (1) donnera directement :

$$\ddot{x} = -\frac{m_p l}{M} \ddot{\theta} \cos(\theta) + \frac{m_p l}{M} \dot{\theta}^2 \sin(\theta) - \frac{\alpha}{M} \dot{x} + \frac{1}{M} F$$

On substitue (II.4.2) dans (II.4.1) pour avoir:

$$\ddot{\theta} = \frac{Mg \sin(\theta) - m_p l \dot{\theta}^2 \sin(\theta) \cos(\theta) + \alpha \dot{x} \cos(\theta)}{l.(M - m_p \cos^2(\theta))} - \frac{\cos(\theta)}{l.(M - m_p \cos^2(\theta))} F \quad (II.6)$$

Ou  $\frac{F_x}{l}$  représente le couple appliqué sur la masse  $m_p$ .

De même on partant du system (II.4), on écrit (II.4.2) sous la forme :

$$l\ddot{\theta} = -\ddot{x} \cos(\theta) + g \sin(\theta)$$

Qu'on remplace dans (II.4.1), on aura :

$$\ddot{x} = \frac{-m_p g \sin(\theta) \cos(\theta) + m_p l \dot{\theta}^2 \sin(\theta) - \alpha \dot{x}}{M - m_p \cos^2(\theta)} + \frac{1}{M - m_p \cos^2(\theta)} F \quad (II.7)$$

Les équations (4) et (5) représente les équations de la dynamique du system qui sous

forme d'état deviennent, pour le vecteur d'état :  $\underline{x}^T = [x \quad \dot{x} \quad \theta \quad \dot{\theta}]$

$$\dot{\underline{x}} = \begin{bmatrix} x_2 \\ \frac{-m_p g \sin(x_3) \cos(x_3) + m_p l x_4^2 \sin(x_3) - \alpha x_2}{M - m_p \cos^2(x_3)} \\ x_4 \\ \frac{Mg \sin(x_3) - m_p l x_4^2 \sin(x_3) \cos(x_3) + \alpha x_2 \cos(x_3)}{l.(M - m_p \cos^2(x_3))} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M - m_p \cos^2(x_3)} \\ 0 \\ \frac{-\cos(x_3)}{l.(M - m_p \cos^2(x_3))} \end{bmatrix} F$$

La sortie est  $\underline{y} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$ .

## 2. 2. Model avec action Tension

La force  $F_x$  étant générer par un moteur asynchrone dont l'équation du couple est :

$$C = K_m i$$

Avec :

$K_m$  Constante mécanique.

$i$  Courant dans le moteur.

La force appliquer au chariot sera :

$$F = \frac{2\pi}{r} C$$

$r$  Rayon du pignon de transfert.

$$i = \frac{v}{R} - \frac{2\pi K_b}{Rr} \dot{\theta}$$

$v$  Tension de commande du moteur.

$R$  Résistance de l'armature.

$K_b$  Constante électrique du moteur.

Alors :

$$F = \frac{1}{R} \frac{2\pi K_m}{r} \left( v - \frac{2\pi K_b}{r} \dot{x} \right) \quad (II.8)$$

Ou bien :

$$F = H_1 v - H_2 x_2$$

Avec :

$$H_1 = \frac{2\pi K_m}{rR}$$

$$H_2 = \left( \frac{2\pi}{r} \right)^2 \frac{K_m K_b}{R}$$

$F$  Dans l'équation

d'état donne

$$\dot{\underline{x}} = \begin{bmatrix} x_2 \\ \frac{-m_p g \sin(x_3) \cos(x_3) + m_p l x_4^2 \sin(x_3) - x_2 (\alpha + H_2)}{M - m_p \cos^2(x_3)} \\ x_4 \\ \frac{Mg \sin(x_3) - m_p l x_4^2 \sin(x_3) \cos(x_3) + x_2 (\alpha + H_2) \cos(x_3)}{l.(M - m_p \cos^2(x_3))} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{H_1}{M - m_p \cos^2(x_3)} \\ 0 \\ \frac{-H_1 \cos(x_3)}{l.(M - m_p \cos^2(x_3))} \end{bmatrix} v$$

Avec  $v$  tension de commande du moteur, la sortie  $\underline{y}$  reste inchangée.

Et ayant la tension  $v$ , on peut avoir le courant  $i$ , et la force  $F$ , avec les équations des contraintes suivantes:

$$i = \frac{v}{R} - \frac{2\pi K_b}{r.R} x_2 \quad F = \frac{2\pi K_m}{R.r} \left( v - \frac{2\pi K_b}{r} x_2 \right) \quad (II.9)$$

### 3. Simulation en Boucle ouverte du modèle

Après avoir obtenu les équations générales de la dynamique d'un pendule à tige de masse négligeable, on va spécifier notre cas avec les paramètres physiques du Tableau II.1

**Tableau II.1:** Paramètres physique du système mathématique

Les paramètres	Signification	Valeur
$m_c$	Poids du chariot	1 Kg
$m_p$	Poids de la masse	0.2 Kg
$l$	Longueur du pendule	0.4 m
$\alpha$	Coefficient de frottement	0.5 Kg/s
$r$	Distance parcourue par tours de pignon	0.0127 m
$K_m$	Constante mécanique du moteur	0.049 N m/A
$K_b$	Constante électromagnétique	0.0507 V/ rad/s
$g$	Accélération de la pesanteur	9.81m/s/s
$R$	Resistance d'armature du moteur	0.3 $\Omega$

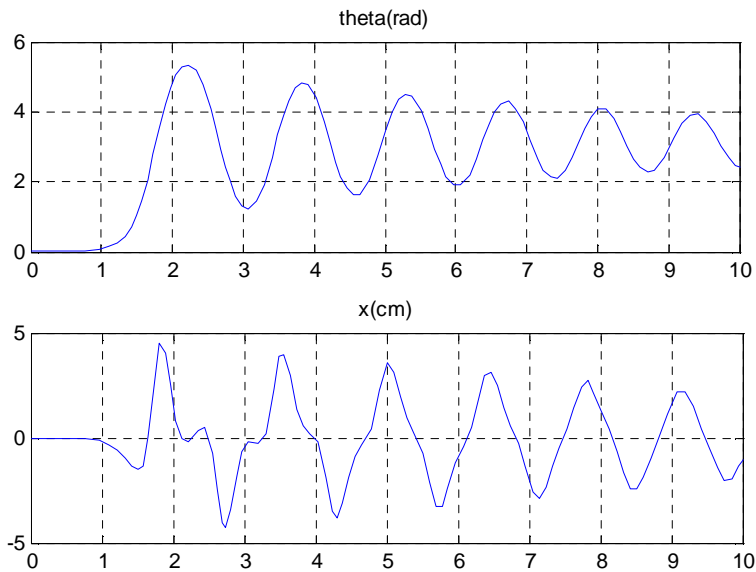
Les constantes  $K_b$  et  $K_m$ , peuvent changés par de petites variations, l'ajustement de ces deux paramètres se fera en comparant entre les résultats de simulation du system réel et du système mathématique.

En fixant les conditions initiales du pendule aux valeurs :  $x_0^T = [0 \ 0 \ 0.001 \ 0]$ , et en utilisant les paramètres physiques précisés dans le tableau (II.1) on obtient les graphiques de la figure (II.2) sans excitation extérieurs sur le système qui évoluera de façon libre. Dans le premier cas, on voit que le système regagne son équilibre stable après une série d'oscillations a la fréquence naturelle qui peut être tirée du graphe II.2, avec 2 périodes en 3 seconde on dira que la pulsation naturelle du système est de

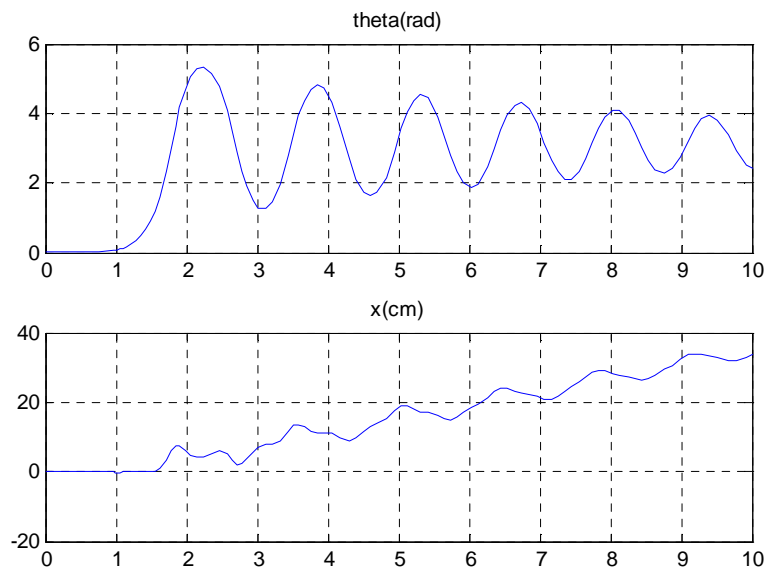
$\omega \approx \frac{2\pi}{T} \approx 4.2 \text{ rad/s}$ , ce qui peut être vérifiée sur la maquette, néanmoins on verra dans

le chapitre 5, que le model présente des insuffisances.

D'autre part en appliquant une entrée en échelon au niveau de l'actionneur électromécanique, et pour la condition initiale précédente, on aura les graphiques de la figure III.3.



**Figure II.2 :** Résultat de simulation du système libre en BO



**Figure II.3:** repense de simulation du système forcé en BO

# Chapitre III

---

Rappels théoriques



## Introduction

Après avoir obtenu le modèle mathématique du système dans le chapitre précédent, On a jugé utile de faire un rappel théorique sur les commandes qu'on va appliquer sur le modèle et puis sur le système réel, et ça sera l'objet de ce présent chapitre.

### 1. La commande Floue et optimisation par réseau neuro-floue hybride

#### 1.1. Introduction à la logique floue [8]

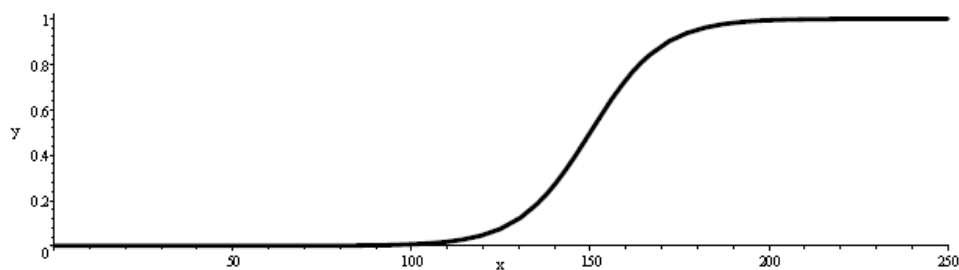
Il y a une imprécision inhérente dans notre langage naturel quand nous décrivons des phénomènes qui n'ont pas de frontières bien définies. Des phrases telles que « Moh est futé » ou « Akila est jeune » sont des exemples simples. Les ensembles flous sont les objets mathématiques modelant cette imprécision du langage. Notre principale préoccupation est de représenter, manœuvrer, et dessiner des inférences de tels rapports imprécis. La théorie des ensembles flous fournit les outils mathématiques pour suivre des processus approximatifs de raisonnement quand l'information disponible est incertaine, inachevée, imprécise, ou vague.

En employant le concept des degrés d'appartenance pour donner une définition mathématique des ensembles flous, nous augmentons le nombre de circonstances produites dans le raisonnement humain qui peut être soumis à la recherche scientifique. Les humains font beaucoup de choses qui peuvent être classées comme commande. Les exemples incluent, monter une bicyclette, frapper une boule avec une batte, et donner un coup de pied à un ballon de foot entre les poteaux. Comment faisons-nous ces choses ? Nous n'avons pas des mesures précises, ou un système d'équations, pour nous indiquer comment commander notre mouvement, mais les humains peuvent néanmoins devenir très habiles à effectuer des tâches très compliquées. Une explication est que nous apprenons par l'expérience, le bon sens, et l'entraînement pour suivre un nombre incalculable de règles de base du type « **si...alors...** ».

L'utilisation des règles de base de cette forme est l'idée fondamentale derrière la commande floue. Des variables linguistiques telles que rapide, lent, grand, moyen, et petit sont traduites en ensembles flous; puis la version mathématique du « si...alors... » est constituée en combinant ces ensembles flous.

## 1.2. Les ensembles flous dans la théorie de la commande [8], [14]

Il est facile d'exprimer des règles en mots, mais la notion linguistique du manque de netteté comme illustré ci-dessus dans les besoins des règles d'être représentées d'une manière mathématique afin de se servir de cette notion dans la théorie de commande. Comment pouvons-nous faire cela? La modélisation mathématique des concepts flous était premièrement présentée par professeur **Lotfi Zadeh** en 1965 pour décrire, mathématiquement, les classes des objets non définis avec une précision d'un critère d'adhésion. Sa controverse est que la signification dans le langage naturel est une question de degré. La notion de grand peut être dépeinte par un graphique tel que la figure II.1, où l'axe des abscisses représente la taille en centimètres, et l'axe des ordonnées représente le degré d'appartenance à l'ensemble des grands hommes, sur une échelle de 0 à 1, attribuée à cette taille.



**Figure III.1:** Fonction d'appartenance à la classe des grands hommes

Avant de définir un ensemble flou mathématiquement on commence par définir un ensemble normal, de façon à pouvoir étendre cette définition vers les ensembles flous. Un ensemble  $A$  d'éléments  $X$  peut être identifiée par une fonction des éléments de  $X$  dans l'ensemble binaire  $\{0,1\}$  comme suit:

$$f(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

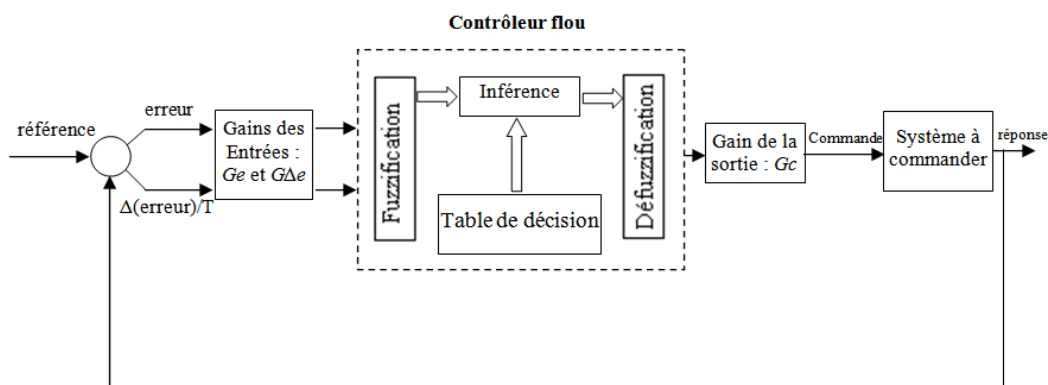
Cette fonction s'appelle la fonction caractéristique ou fonction d'indication de l'ensemble flou A. En revanche, les éléments des sous-ensembles flous de X peuvent avoir des divers niveaux d'adhésion de 0 à 1.

*Definition 1.1:* Un ensemble flou A d'éléments X est une fonction d'appartenance  $A(x)$  définit de X vers l'intervalle [0,1].

### 1.3. Le Contrôleur Flou

L'architecture typique d'un contrôleur logique flou est présentée sur la Figure III.2. L'erreur  $e(i)$  est prise à chaque i-ème instant d'échantillonnage. La variation de l'erreur est calculée par la formule suivante:

$$\Delta e = \frac{e(n) - e(n-1)}{T}$$



**Figure III.2 :** Architecture d'un contrôleur par logique floue

$e(n)$  étant la valeur de l'erreur au  $n^{\text{ième}}$  instant d'échantillonnage et  $e(n-1)$  la valeur de l'erreur au  $(n-1)^{\text{ième}}$  instant. T est la période d'échantillonnage.

Les variables linguistiques sont l'erreur ( $e$ ), la variation de l'erreur ( $\Delta e$ ) et la commande ( $c$ ). Chacune d'elles a ses propres valeurs linguistiques et son univers de discours.

La fuzzyfication assure la transformation des grandeurs physiques (entrées) en valeurs linguistiques qui peuvent être traitées, ensuite, par le mécanisme d'inférence et ce grâce aux fonctions d'appartenances des valeurs linguistiques concernées par la valeur physique.

Si l'erreur physique dans la figure III.3 en entrée est égale à 0.1, après fuzzyfication l'erreur est égal à Z avec un degré 0.8 et est égale aussi à PM avec un degré 0.2.

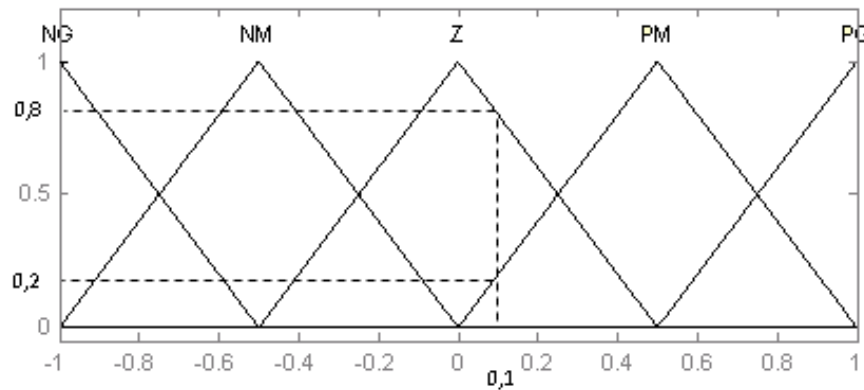


Figure III.3. Exemple de fuzzyfication

La table de décision contient l'ensemble des règles floues nécessaire à la commande du système. Le choix des règles à utiliser peut être difficile à déterminer, néanmoins, la table standard diagonale de Mc.Vicar-Whelan (Tableau III.1) peut être appliquée à plusieurs problèmes de commande floue.

Tableau III.1 : Table de décision standard de Mc.Vicar-Whelan à 5 ensembles flous

$\Delta e \backslash e$	NG	NM	Z	PM	PG
NG	NG	NG	NM	NM	Z
NM	NG	NM	NM	Z	PM
Z	NM	NM	Z	PM	PM
PM	NM	Z	PM	PM	PG
PG	Z	PM	PM	PG	PG

Les règles floues de cette table peuvent être écrites sous la forme suivante :

**Si  $e=NG$  ET  $\Delta e=NG$  Alors  $c=NG$**

**SI  $e=NG$  ET  $\Delta e=NM$  Alors  $c=NG$**

.....

**Si  $e=PG$  ET  $\Delta e=PG$  Alors  $c=PG$** 

A partir des valeurs fuzzyfiées, le mécanisme permet d'obtenir en utilisant les fonctions d'appartenance de chaque variable linguistique (entrées et sortie) ainsi que les règles floues, la fonction d'appartenance du sous ensemble flou solution de la commande. En d'autres termes, il établit avec tel degré une commande satisfait la situation courante. Il existe plusieurs méthodes d'inférence qu'on trouve dans la littérature.

#### **1.4. Optimisation Neuronal des réseaux d'inférences flous, illustration de l'algorithme hybride ANFIS et de l'anfisedit [19], [14]**

Dans ce chapitre, nous discutons des techniques pour combiner des méthodes neuronales et floues pour construire des contrôleurs. Il y a beaucoup de manières dont ces méthodes peuvent être combinées.

En général tous les systèmes issus d'une combinaison des techniques neuronales et floues sont appelée système NeuroFloue, les différentes combinaisons de ces techniques peuvent être divisées en les classes suivantes :

- Système neuro flou coopérative
- Système neuro flou concourant
- Système neuro flou hybride

Un système Neuro Flou coopératif se caractérise par un bloc préprocesseur où le mécanisme d'apprentissage du réseau neurologique détermine les fonctions d'appartenance du système d'inférence flou, et les règles floues à partir de données d'apprentissage. Une fois que les paramètres de système d'inférence flou sont déterminés, l'algorithme du réseau de neurones s'arrête et seul le système d'inférence floue continue son exécution. C'est l'initialisation par réseaux de neurones.

Dans un système NeuroFloue concourant, on trouve un bloc post processeur. Le réseau de neurones aide le système d'inférence flou continuellement pour déterminer les paramètres exigé, et particulièrement si les variables d'entrée du contrôleur ne peuvent

pas être mesurées directement. Dans certains cas les sorties du système d'inférence flou ne pourraient pas être directement applicables au processus.

Et puis finalement dans les systèmes neuro flous Hybride, le réseau neurologique est utilisé pour apprendre de façon itérative les paramètres du système d'inférence flou. Lorsqu'on évoquera les systèmes NeuroFlous de façon générale, c'est à ce type de système qu'on fait référence. Ce type de système est caractérisé par l'utilisation d'un algorithme d'apprentissage basé sur la descente du gradient ou sur les réseaux de neurones, pour déterminer les paramètres du système d'inférence flou à travers les données d'apprentissage en entrée et en sortie.

### 1. 3.1. Illustration de l'algorithme ANFIS

C'est un système d'inférence adaptatif Neuro Floue qui consiste à utiliser un réseau de neurone de type Multi Layer Perceptron à cinq couches, pour lequel chaque couche correspond à la réalisation d'une étape d'un système d'inférence floue de type Takagi Sugeno. Pour la simplicité, nous supposons que le système d'inférence flou à deux entrées  $x$  et  $y$ , et a comme une sortie  $f$ . Supposer que la base de règle contient deux règles floues de type Takagi-Sugeno.

Règle1 : SI  $x$  est  $A_1$  et  $y$  est  $B_1$  ALORS  $f_1 = p_1 x + q_1 y + r_1$

Règle2 : SI  $x$  est  $A_2$  et  $y$  est  $B_2$  ALORS  $f_2 = p_2 x + q_2 y + r_2$

L'ANFIS à une architecture posée par cinq couches comme représenté sur la figure III.4.

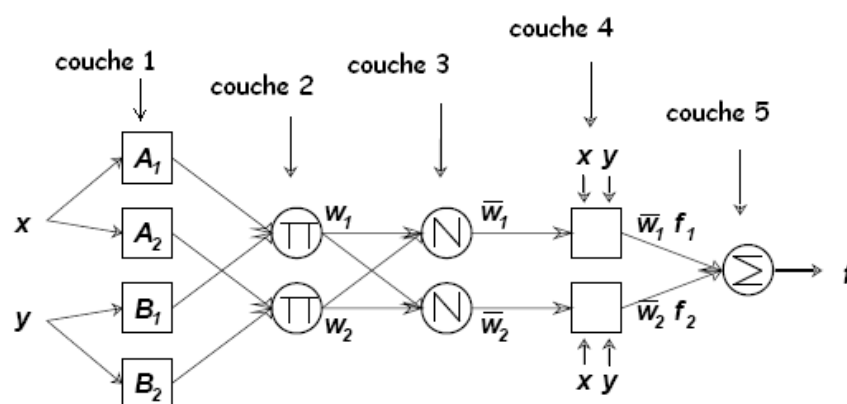


Figure III.4 : Architecture d'une commande ANFIS

Une architecture classique peut être décrite par une première couche comportant autant de neurones qu'il y'a de sous ensembles flous dans le système d'inférence représenté. Chaque neurone calcule le degré de vérité d'un sous ensemble flou particulier par sa fonction de transfert. La seule restriction sur le choix de cette fonction concerne sa dérivabilité. En retrouve dans la littérature, l'utilisation, de fonctions gaussiennes et les paramètres modifiables sont le centre et la pente de la gaussienne (variance). La fonction d'activation des neurones  $i$  de la première couche est donnée par:  $f_i^1 = \mu_{A_i}(x)$ , tel que  $x$  est l'entrée au neurone  $i$ , et  $A_i$  est un sous ensemble flou correspondant au variable  $x$ . En d'autres termes,  $f_i^1$  est la fonction d'appartenance du  $A_i$  et il indique le degré auquel donné  $x$  satisfait le quantifier  $A_i$ . Nous choisissons  $\mu_{A_i}(x)$  pour être en forme de (Gaussien, triangle, trapézoïdal) avec le maximum égal à 1 et le minimum égal à 0, tel que les fonctions généralisée de ces formes est :

$$\text{Triangle :} \quad \mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

$$\text{Trapézoïdale :} \quad \mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$$

$$\text{Gaussienne :} \quad \mu(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2}\right)$$

Où  $\{a, b, c, \sigma\}$  est l'ensemble des paramètres. Pendant que les valeurs de ces paramètres changent, les fonctions en forme précédente changent en conséquence, de ce fait présenter de diverses formes de fonction d'appartenance sur la variable linguistique  $A_i$ . Les paramètres dans cette couche sont désignés sous le nom des paramètres de fonction d'appartenance.

La deuxième couche cachée sert à calculer le degré d'activation des prémisses. Les neurones de la cette couche représentent chacun la prémisse d'une règle. Ils reçoivent en entrée le degré de vérité des différents sous-ensembles flous composant cette prémisse et ont en charge le calcul de son propre degré de vérité. Les fonctions

d'activation utilisées pour ces neurones dépendant des opérateurs présents dans les règles (ET ou OU).

La fonction d'activation des neurones  $i$  de la première couche :

$$W_k = \mu_{A_i}(x) \times \mu_{B_j}(y)$$

Où  $k$  : représente le nombre de règle,  $i$  : représente le nombre de partition de  $x$ , et  $j$  : le nombre de partition de  $y$ .

La troisième couche cachée normalise de degré d'activation des règles. Chaque neurone dans cette couche est un neurone de cercle noté  $N$ . le  $i^{\text{ème}}$  neurone calcule le rapport entre  $i^{\text{ème}}$  poids de règles et la somme de toutes les poids des règles. Cette opération est appelée la normalisation des poids.

$$\overline{W}_k = \frac{W_k}{\sum W_i}$$

L'ensemble des sorties de cette couche seront appelées les poids normalisé.

La quatrième couche cachée sert à déterminer les paramètres la partie conséquence des règles ( $p, q, r$ ). La fonction de chaque neurone dans cette couche est la suivante

$$f_k^4 = \overline{W}_k \times f_k = \overline{W}_k \times (p_k x + q_k y + r_k)$$

Où  $\overline{W}_k$  est la sortie de la troisième couche, et  $\{p_i, q_i, r_i\}$  est l'ensemble des paramètres. Ces paramètres désigné sous le nom les paramètres conséquents.

La couche de sortie contient un seul neurone dans cette couche, est un neurone de cercle noté  $S$  qui calcule la sortie globale comme addition de tous les signaux entrants, c'est-à-dire :



$$f^5 = \sum_k \overline{W}_k \times f_k^4$$

La figure III.5 représente un système ANFIS, à 2 entrées chaque entrée répartie en trois sous ensemble flou et 9 règles.

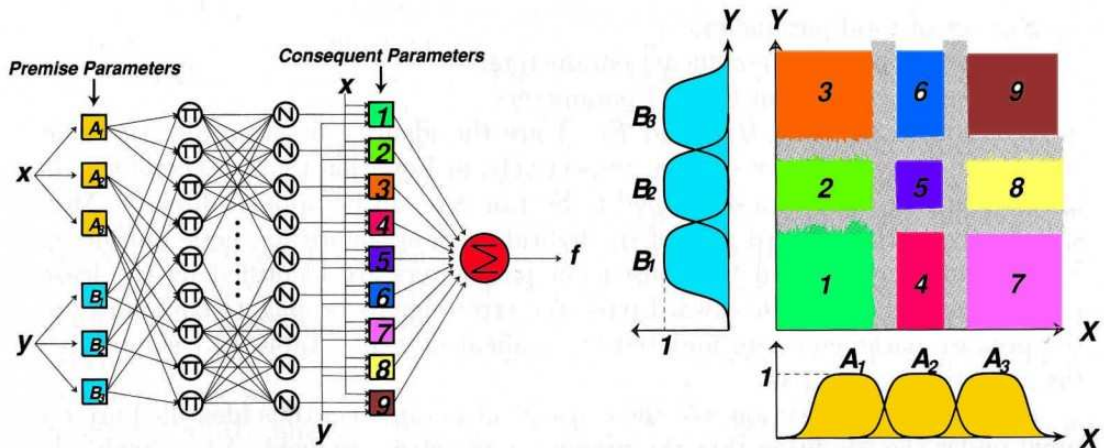


Figure III.5: Exemple ANFIS à 2 entrées avec 9 règles

**Tableau III.2:** les différentes couches d'un système ANFIS tel que n est le nombre des entrées, p est le nombre des sous ensembles flous d'entrée (partition flou)

Les différentes couches	Type des couches	Le nombre de neurone dans la couche
Couche 0	Les entrées	n
Couche 1	Les valeurs	(p.n)
Couche 2	Les règles	$p^n$
Couche 3	La normalisation	$p^n$
Couche 4	Linéarisation des fonctions	$p^n$
Couche 5	Somme	1

Noter que les neurones dans l'architecture ANFIS ont différentes structures:

- Valeurs : fonction d'appartenance définie par différentes formes.
- Règles : habituellement produit.
- Normalisation : division de somme et d'arithmétique.

- Fonctions : régressions linéaires et multiplication avec  $\bar{w}$ , tel que  $\bar{w}$  est la normalisation du poids  $w$ .
- la sortie : Somme Algébrique.

Le système ANFIS applique le mécanisme d'apprentissage des réseaux de neurones sur des techniques d'inférence floues. En d'autres termes l'ANFIS est un système d'inférence flou (FIS) dont les paramètres des fonctions d'appartenances sont ajustés en utilisant l'algorithme d'apprentissage de rétro propagation, ou en combinaison avec un autre type d'algorithmes comme le moindre carré.

Dans l'architecture ANFIS proposée dans la figure III.6, la sortie globale peuvent être exprimée en tant que des combinaisons linéaires des paramètres conséquents. Avec plus de précision, la sortie sur cette figure peut être réécrite comme suit :

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2$$

La sortie est une fonction linéaire des paramètres conséquences (p, q, r). L'architecture ANFIS est une représentation paramétrique des deux ensembles de paramètres: S1 et S2 tel que :

S1 représente les paramètres des ensembles flous utilisés pour la fuzzyfication dans la première couche de système ANFIS

$$S1 = \{\{a_{11}, b_{11}, c_{11}\}, \{a_{12}, b_{12}, c_{12}\}, \dots, \{a_{1p}, b_{1p}, c_{1p}\}, \dots, \{a_{np}, b_{np}, c_{np}\}\}$$

Ou p est le nombre de partition floue de chacune des variables d'entrées et n est le nombre de variables d'entrées.

S2 représente les coefficients des fonctions linéaires (les paramètres conséquents)

$$S2 = \{p_1, p_2, p_3, \dots, q_1, q_2, q_3, \dots, r_1, r_2, r_3, \dots\}$$

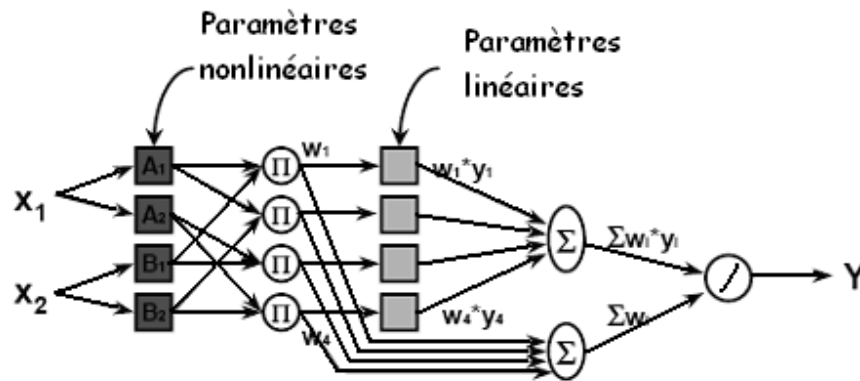


Figure III.6: Méthode d'apprentissage Hybride

Tableau III.3 : Les paramètres à ajuster d'un système ANFIS

	Passage Vers l'avant	Passage En arrière
Paramètre de fonction d'appartenance (non linéaire $a_i, b_i, c_i$ )	Fixe	Rétro propagation
Paramètre de coefficient (Linéaire $p, q, r$ )	moindres carrés	fixe

ANFIS utilise un cycle d'apprentissage de deux passages :

- le passage en avant: S1 est fixe et S2 est calculé en utilisant l'algorithme de moindre carré de l'erreur (LSE). (Le LSE est appliqué seulement une fois lorsque commencer à obtenir les valeurs initiales des paramètres conséquents).
- Passage en arrière: S2 est fixe et S1 est calculé en utilisant l'algorithme de Rétro Propagation.

### 1.3.2. Rétro Propagation de l'ANFIS

Mesure d'erreur  $E_k$  pour l'entrée  $k^{ième}$  donné d'apprentissage: 
$$E_k = \sum_{i=1}^{N(L)} (d_i - x_{L,i})^2$$

$N(L)$  : Nombre de neurones dans la couche L

$d_i$  : lemme composant de vecteur de sortie désiré

$x_{L_i}$ : lemme composant de vecteur de sortie de réel (de l'ANFIS)

Mesure d'erreur globale E: 
$$E = \sum_{k=1}^K E_k$$

Pour chaque paramètre  $\alpha_i$  ( $\alpha_i \in S_1$ ) sont modifiés en suivant l'équation :

$$\Delta\alpha_i = -\eta \frac{dE}{d\alpha_i}$$

Où  $\eta$  est une constante positive appelée pas d'apprentissage.

### 1.3.3. L'outil 'anfisedit'

Pour lancer l'outil il suffit de taper >>anfisedit sur la ligne de commande, le GUI de la figure III.11 s'affiche:

A partir de ce GUI on pourrait:

- Charger les données d'apprentissage, de vérification ou de test on sélectionnant les boutons radio appropriée dans la partie load data, les données seront traces sur le plot région.
- Charger ou générer un premier réseau d'inférence a partir de la portion générâtes FIS.
- Visualiser le FIS générer ou charger a partir du bouton structure de la portion ANFIS info.
- Choisir la méthode d'optimisation des paramètres, dans la portion train FIS, on a le choix entre la retro propagation et la rétropropagation mélanger au moindres carrées (hybride).
- Choisir le nombre d'itérations (epochs) et l'erreur tolérer.
- Lancez l'entraînement en choisissant le bouton train now.
- Comparer la sorti du model après entrainement aux données d'apprentissage, à partir du bouton test now.

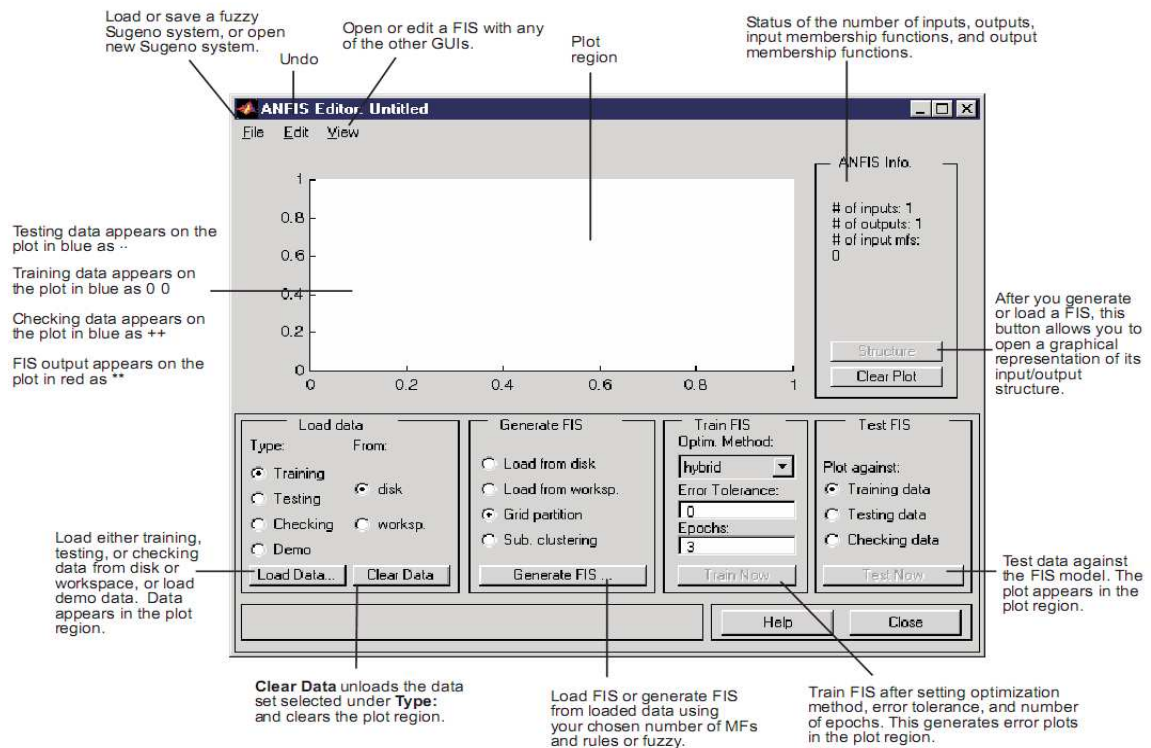


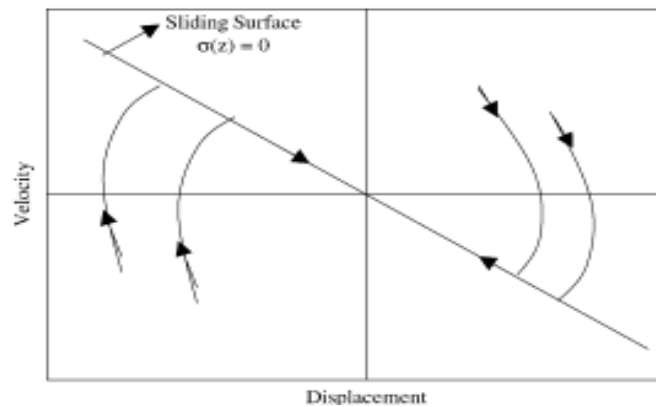
Figure III.7: GUI anfisedit

## 2. La commande par Modes de Glissement et Optimisation par Logique Floue

### 2.1. La commande par MDG [7], [17], [21]

Le régime glissant intervient de manière prépondérante dans la définition et les propriétés d'une classe de systèmes de commandes très importante qui sont les systèmes de commande à structure variable. Les premiers travaux concernant les systèmes de commande à structure variable en modes de glissement ont été proposés et élaborés au début des années 1950 en Union Soviétique par EMELYANOV.

Le principe de modes glissant est illustré dans la figure III.8

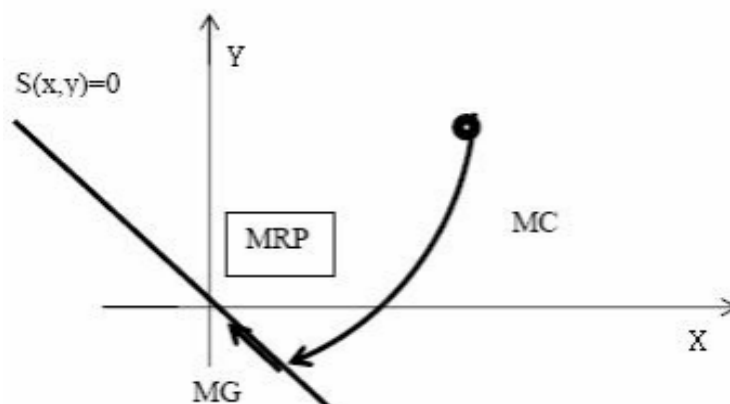


**Figure III.8 :** Principe des modes glissants

La région I est définie par  $s(x_1, x_2) > 0$  alors que la région II est définie par  $s(x_1, x_2) < 0$ . Les allures des trajectoires de phase dans la région I et II sont donc très différentes. La droite  $x_1 = 0$  correspond à une frontière où les trajectoires d'allure différentes se rejoignent alors que la droite  $0.5x_1 + x_2 = 0.5x_1 + \dot{x}_1 = 0$  décrit une trajectoire de phase particulière, représentant la dynamique d'ordre inférieur donnée par  $0.5x_1 + \dot{x}_1 = 0$ . Quand la dynamique du système est décrite de cette manière, on dit qu'il est en mode glissant.

Ceci montre que ce type de systèmes a en général deux modes de fonctionnement (figure III.9):

- Le mode non glissant, (reaching mode) ou encore mode d'accès.
- Le mode glissant, (sliding mode).



**Figure III.9 :** Les étapes de glissement

La commande par modes glissants est une des techniques de commandes les plus connues efficaces et robustes en automatique. Cependant, la plupart des algorithmes de CMG d'ordre arbitraire souffrent de l'absence de conditions suffisantes de convergence constructives.

Le principe de la CMG est, à l'aide d'une commande discontinue, de contraindre le système à atteindre une surface donnée pour ensuite y rester. La synthèse d'une CMG se déroule en deux temps :

- une surface est déterminée en fonction des objectifs de commande et des propriétés statiques et dynamiques désirées pour le système bouclé,
- une commande discontinue est synthétisée de manière à contraindre les trajectoires d'état du système à atteindre et, ensuite, à rester sur cette surface en dépit d'incertitudes, de variations de paramètres.

Soit le système non linéaire, affine en la commande, défini par :

$$\dot{x} = f(x, t) + g(x, t)u + p(x, t)$$

Soit  $S(x, t) : X \times \mathbb{R}^+ \rightarrow \mathbb{R}$  une fonction suffisamment différentiable et considérée comme une sortie fictive du système telle que son annulation permette de satisfaire l'objectif de commande. La fonction  $S(x, t)$  est appelée *variable de glissement*.

L'ensemble :  $S = \{x \in X : s(x, t) = 0\}$

Représente alors une sous-variété de  $X$  de dimension  $(n-1)$  appelée *surface de glissement*.

Une fois la surface de glissement choisie, la seconde étape consiste à construire une commande  $u$  de façon à ce que les trajectoires d'état du système soient amenées vers cette surface et soient ensuite maintenues dans un voisinage de celle-ci malgré la présence d'incertitudes et de perturbations sur le système. Une condition nécessaire et suffisante, appelée condition d'attractivité, pour qu'une variable de glissement  $s(x, t)$  tende vers 0 est :

$$s\dot{s} < 0$$

Cependant, cette inégalité n'est pas suffisante pour assurer une convergence en temps fini vers la surface et donc, un fonctionnement en régime glissant. Elle est généralement remplacée par la condition suivante, appelée *condition de  $\eta$ -attractivité* :

$$s\dot{s} < -\eta|s|, \eta > 0.$$

La méthode dite de la commande équivalente est un moyen de décrire le comportement du système lorsqu'il est restreint à la surface  $\{s = 0\}$ . Elle est obtenue grâce aux conditions d'invariance de la surface :

$$\begin{aligned} s &= 0 \\ \dot{s} &= \frac{\partial s}{\partial x}(f(x,t) + g(x,t)u_{eq}) = 0 \end{aligned}$$

Où  $u_{eq}$ , appelé commande équivalente, est associée au système nominal, i.e. sans incertitude ; elle est déterminée de façon unique par les conditions ci-dessus :

$$u_{eq} = -\left(\frac{\partial s}{\partial x}g(x)\right)^{-1} \frac{\partial s}{\partial x}f(x)$$

Cependant, cette commande ne force pas les trajectoires du système à atteindre la surface de glissement. Ainsi, la commande  $u$  est la somme de la commande équivalente et d'une composante discontinue assurant un régime glissant et l'insensibilité du système vis-à-vis des incertitudes et des perturbations, i.e.

$$u = u_{eq} - G\left(\frac{\partial s}{\partial x}g(x)\right)^{-1} \text{sign}(s)$$

La forme non linéaire de la surface de glissement est une fonction de l'erreur sur la variable à réglé  $x$ , elle est donnée par:

$$S(X) = \left(\lambda + \frac{d}{dt}\right)^{r-1} e(t)$$

$X$  : vecteur d'état.

$e(t)$  : erreur de réglage :  $e(t) = Y(t) - Y_d(t)$

$\lambda$  : Constante positive de glissement.

$r$  : Le degrés relatif de système.



$S(X)=0$  est une équation différentielle linéaire et dont l'unique solution est  $e=0$  et cela pour un choix convenable de  $\lambda$ .

## 2.2. Le phénomène de broutement (*chattering*)

Ce phénomène constitue un désavantage non négligeable car, même s'il est possible de le filtrer en sortie du processus, il est susceptible d'exciter les modes à haute fréquence qui n'ont pas été pris en compte dans le modèle du système. La réticence implique également d'importantes sollicitations mécaniques au niveau des actionneurs, pouvant provoquer leur usure rapide, ainsi que des pertes énergétiques non négligeables au niveau des circuits de puissance électrique. [17]

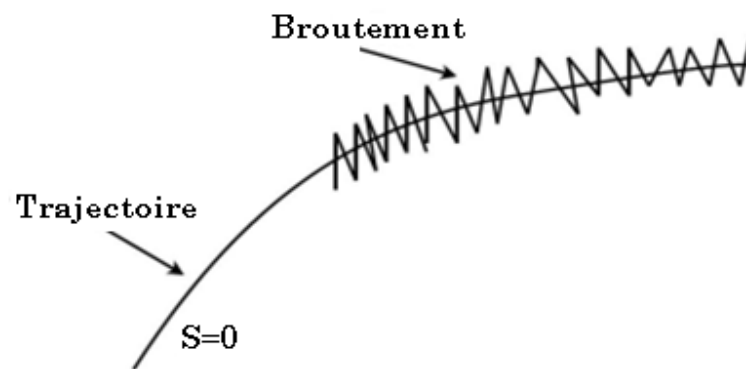
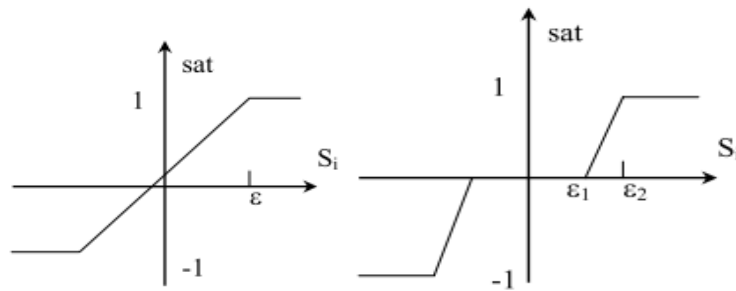


Figure III.10: phénomène de Broutement

## 2.3. Elimination du phénomène du *Chattering* avec contrainte sur la robustesse

De nombreuses études ont été effectuées dans le but de réduire ou d'éliminer ce phénomène. L'une des solutions envisagées consiste à remplacer la fonction signe par une approximation continue, de type grand gain, dans un voisinage de la surface, telle que la fonction *saturation* ou une fonction *sigmoïde* (par exemple

$$\tanh\left(\frac{x}{\varepsilon}\right), \frac{\pi}{2} \arctan\left(\frac{x}{\varepsilon}\right)$$

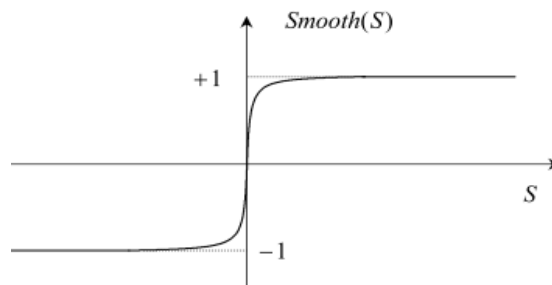


**Figure III.11** : Fonction SAT avec un seuil et deux seuils (zone morte).

$$sat(S) = \begin{cases} -1 & si \quad S < -\epsilon \\ S & si \quad |S| < \epsilon \\ \epsilon & si \quad S > -\epsilon \\ 1 & si \quad S > \epsilon \end{cases}$$

$$sat(S) = \begin{cases} 0 & si \quad |S| < \epsilon_1 \\ \frac{S - \epsilon_1}{\epsilon_2 - \epsilon_1} & si \quad \epsilon_1 < |S| < \epsilon_2 \\ sign(S) & si \quad |S| > \epsilon_2 \end{cases}$$

On peut aussi donner la fonction Smooth:  $Smooth(S) = \frac{S}{|S| + \mu}$



**Figure III.12** : Fonction « Smooth ».

Le régime glissant qui en résulte n'est plus confiné dans  $S$ , mais dans un proche voisinage de celui-ci. Bien que cela permette d'atténuer le phénomène de réticence, la précision par rapport à l'objectif fixé, la robustesse de la commande et le temps de réponse s'en trouvent dépréciés.

## 2.4. Optimisation du MDG par Correction floue de la couche limite de commande [9]

Différentes techniques de commande de FSM peuvent être classifiées dans deux catégories : Les bandes limites floues « *fuzzy boundary layer* » et les fonctions de

Lyapunov flous. Dans la première et la plus répandue approche la couche de frontière est remplacée par un système flou, de ce fait, elle garde la structure globale de commande de MG et réalise des garanties de stabilité et de robustesse similaire à la commande par MG seul. A plus loin l'approche emploie un contrôleur flou pour un réglage en ligne de la largeur de couche de limites.

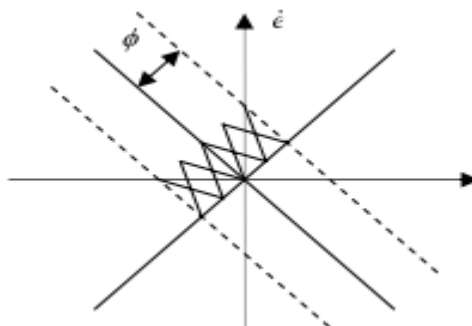
La deuxième approche utilise la loi de commande floue qui substitue complètement les lois de commande de SM, gardant la validité de la *condition de  $\eta$ -attractivité*. Cette méthode est généralement plus intuitif que l'ancienne et s'avère habituellement être moins intense côté calcul numérique.

Dans le but de réduire les oscillations haute fréquence (indésirables sur les réponses), nous allons présenter une solution qui consiste à imposer une variation de la valeur de la commande en fonction du rapport entre l'état de la surface de glissement et une bande couche floue appelée "*Fuzzy Boundary Layer*". Ce qui peut être vu comme une "bande entourant" la surface de commutation, c.à.d. remplacer dans la formule de la commande de maintien la fonction *signe* par la fonction *saturation* donc on aura :

$$u = \varepsilon \text{sat} \left( \frac{s}{\phi} \right)$$

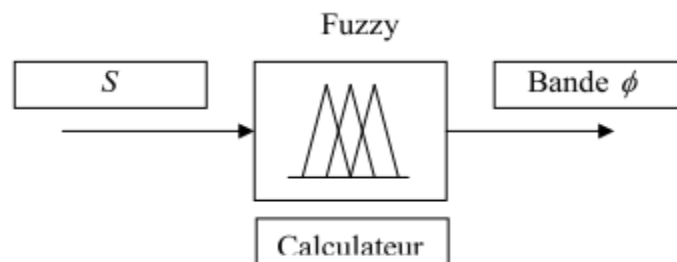
Où  $\phi$  est l'épaisseur de la couche limite (Boundary Layer) et la fonction saturation est

$$\text{définie par : } \text{sat} \left( \frac{s}{\phi} \right) = \begin{cases} \frac{s}{\phi}; & \text{si } \left| \frac{s}{\phi} \right| \leq 1 \\ \text{sign} \left( \frac{s}{\phi} \right); & \text{si } \left| \frac{s}{\phi} \right| > 1 \end{cases}$$



**Figure III.13:** Fuzzy Boundary Layer (Couche limite Floue)

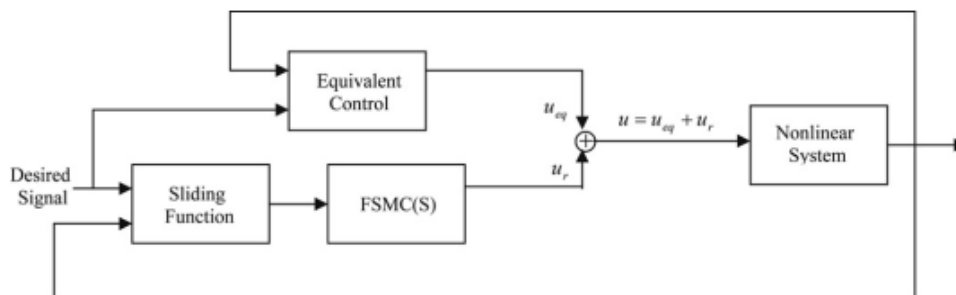
L'introduction de la couche *limite* autour de la surface de commutation réduit les vibrations au coût de l'erreur de poursuite qui s'agrandisse. La couche limite variable est une solution très répandue à ce problème. Dans notre cas, nous avons optés pour une couche variable par logique floue afin d'améliorer l'exécution de la commande par modes glissants. Ceci mène à une stratégie pour ajuster automatiquement l'épaisseur de la couche avec un régulateur flou qui adopte la surface de glissement  $S$  comme une entrée et l'épaisseur de la couche  $\phi$  comme sortie comme le montre la figure suivante :



**Figure III.14 :** calculateur flou de  $\phi$ .

À partir de la base de connaissance on peut fixer les fonctions d'appartenance et la base des règles qui régissent le régulateur flou.

Le principe de la seconde approche est illustré sur la figure III.15.



**Figure III.15:** schéma Simulink du régulateur flou glissant

On défini la commande par modes glissant comme nous l'avons vu précédemment:

$$u = u_{eq} + u_n$$

Telle que la commande équivalente  $u_{eq}$  est calculée de la même manière qu'on modes glissant et la commande de maintien est définie comme suit :

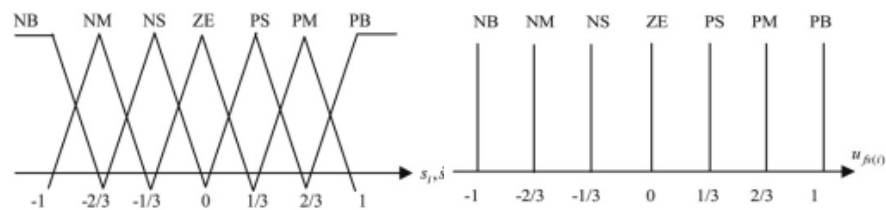
$$u_n = k_{fs} u_{fuzzy}$$

Où  $k_{fs}$  est le facteur de normalisation de sortie de régulateur et  $u_{fuzzy}$  est la sortie de régulateur flou, elle est déterminée par  $S$  et  $\dot{S}$  normalisés.

Les règles de commande floue peuvent être représentées en interférant les variables linguistiques d'entrée  $S$  et  $\dot{S}$  à la sortie linguistique  $u_{fuzzy}$  i.e. :

$$u_{fuzzy} = F S M C (S, \dot{S})$$

Les fonctions d'appartenance pour chaque entrée linguistique  $S$  et  $\dot{S}$  et celle de la sortie linguistique  $k_{fs}$  sont représentées sur les figures suivantes :



**Figure III.16 :** Fonctions d'appartenance des entrées (gauche), la sortie floue (droite)

# Chapitre IV

---

Synthèse des lois de commande

## Introduction

Se basant sur les chapitre précédents, dans ce chapitre, des lois de commande de stabilisation globale ( basculement) et local, hybrides ou standards, seront synthétisées, et analysées sur la base de, la largeur de la plage de conditions initiales pouvant être stabilisées, et aussi en terme de robustesse aux variations paramétriques du modèle, mis au point dans le chapitre précédent, et puis finalement en testera la sensibilité, du système de commande, aux perturbations extérieurs. Enfin on établira un bilan comparatif des différentes techniques utilisées.

### 1. Commande de balancement vers le haut

L'objectifs de ces lois de commandes étant de ramener le bras du pendule le plus proche possible de la position d'équilibre instable, et ainsi le relai sera donner a la commande de stabilisation, pour le maintien de la stabilité. On a choisit d'appliquer deux techniques différentes pour la réalisation de cet objectif, l'une basée sur un modèle, l'autre basée sur le principe de la logique floue. La première est réalisée à partir du principe de stabilisation par méthode directe de Lyapunov, afin de crée un retour d'état non linéaire permettant de conduire la fonction choisit de Lyapunov vers zéro. La deuxième se basant sur un ensemble de règles intuitives et très proches de la logique expérimentale. Un des buts de ce paragraphe est de mettre en évidence les plus et les moins des deux différentes techniques.

#### 1.1. Le balancement du bras par retour d'état non linéaire

Sous ce titre, nous proposons une stratégie de commande pour le balancement vers le haut du pendule inversé sur chariot avec longueur restreinte du longeron. La fonction de Lyapunov sera définie comme la somme du carré de l'énergie du pendule et du carré pesé de la vitesse du chariot. La stratégie est représentée sous une formule compacte et a deux paramètres de modélisation  $\lambda$  et  $\beta$ . En réglant ces paramètres de modélisation, nous pouvons effectuer le balancement vers le haut tel que la restriction à la longueur

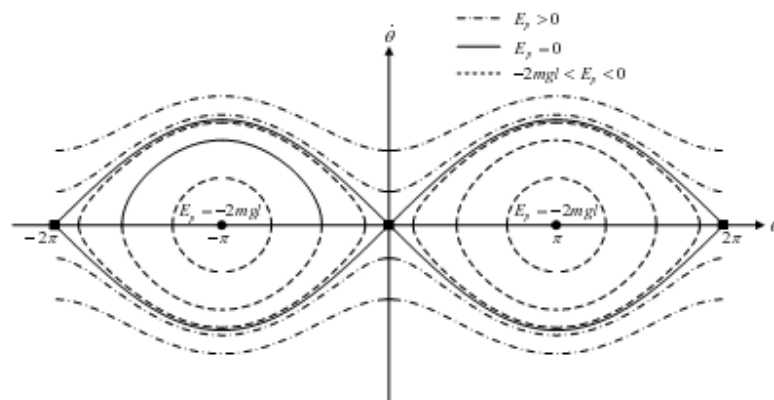
de longeron est satisfaisante. Nous fournissons également une transformation dépendante de l'état pour obtenir l'entrée en tension à un moteur à C.C pour produire la commande en tension à partir de la l'accélération du chariot obtenue à partir de la stratégie proposée. Les équations du système tirées de la modélisation précédente étant:

$$\begin{cases} M\ddot{x} + m_p l \ddot{\theta} \cos(\theta) - m_p l \dot{\theta}^2 \sin(\theta) + \alpha \dot{x} = F_x \\ l \ddot{\theta} + \ddot{x} \cos(\theta) - g \sin(\theta) = 0 \end{cases}$$

On définissant l'énergie du pendule comme:

$$E_p = \frac{1}{2} m_p l^2 \dot{\theta}^2 - m_p l g (1 - \cos(\theta))$$

Si la stratégie du balancement vers le haut peut assurer que  $\dot{x}$  converge vers une constante et que  $E_p$  approche zéro, la trajectoire d'état  $(\theta, \dot{\theta})$  approchera l'orbite représentée en ligne continue sur la figure IV.1, ce qui implique que le pendule s'approche de la position droite avec le temps qui passe.



**Figure IV.1** : Les différents niveaux énergétiques de la trajectoire d'état du pendule

Puisque la longueur de longeron est restreinte, la valeur la plus appropriée où le  $\dot{x}$  doit converger devrait être zéro. Autrement, la stratégie du balancement pourrait violer la restriction de longueur du longeron. Afin d'assurer que le  $E_p$  et le  $\dot{x}$  approcheront zéro, nous proposons une fonction candidate de Lyapunov comme suit :

$$V = E_p^2 + E_c \quad \text{Où :} \quad E_c = \frac{1}{2} m_c \beta \dot{x}^2$$



Avec  $\beta > 0$  est un paramètre de modélisation. Nous calculons d'abord la dérivé de  $V$  en ce qui concerne le temps comme suit :

$$\dot{V} = 2E_p \dot{E}_p + m_c \beta \dot{x}\ddot{x}$$

$$\dot{V} = 2E_p \left[ m_p l^2 \dot{\theta}\ddot{\theta} - m_p l g \dot{\theta} \sin(\theta) \right] + m_c \beta \dot{x}\ddot{x}$$

Puis de (1), on tire:

$$l \ddot{\theta} = -\ddot{x} \cos(\theta) + g \sin(\theta)$$

$$\text{Donc: } \dot{V} = 2E_p \left[ -m_p l \dot{\theta} \ddot{x} \cos(\theta) \right] + m_c \beta \dot{x}\ddot{x}$$

$$\dot{V} = -\dot{x} \left[ 2E_p m_p l \dot{\theta} \cos(\theta) - m_c \beta \dot{x} \right]$$

On considère l'accélération du chariot comme variable de commande,  $u = \ddot{x}$ . Et prenons:

$$u = \lambda \left[ 2E_p m_p l \dot{\theta} \cos(\theta) - m_c \beta \dot{x} \right]$$

Avec  $\lambda$  comme paramètre de réglage, alors:

$$\dot{V} = -\lambda \left[ 2E_p m_p l \dot{\theta} \cos(\theta) - m_c \beta \dot{x} \right]^2$$

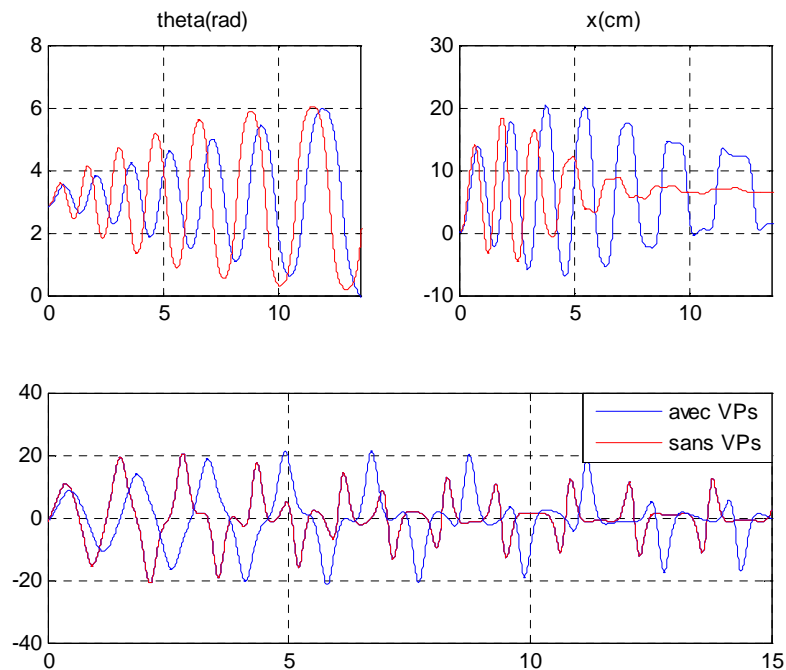
Ce qui assure  $V\dot{V} < 0$  et donc :  $V \rightarrow 0$

La commande étant donné en accélération, une transformation, état dépendent, l'état de l'action s'impose. De la modélisation linéaire du système, on tire la relation:

$$v = \frac{1}{H_m} \cdot (M - m_p \cos^2(\theta)) \cdot \left[ \ddot{x} - \frac{-m_p g \sin(\theta) \cos(\theta) + m_p l x_4^2 \sin(\theta) - (\alpha + RH_b) \cdot \dot{x}}{M - m_p \cos^2(\theta)} \right]$$

Il existe plusieurs techniques de ce genre qui permettent d'envoyer le système vers un point instable en particulier, notamment la méthode exposée dans la référence [23], ou même le déplacement est inclus dans la fonction de Lyapunov ce qui le force au zéro stable à l'état final, avec en tous quatre paramètres de réglage.

Néanmoins dans ce cas un réglage appropriées des gains paramétrait d'atteindre cette objectif aussi bien que d'autre méthodes, ces objectifs, étant d'assurer une arriver du bras en un temps acceptable et avec le chariot au centre du longeron, ou avec un décalage minime, au moment de l'arriver.



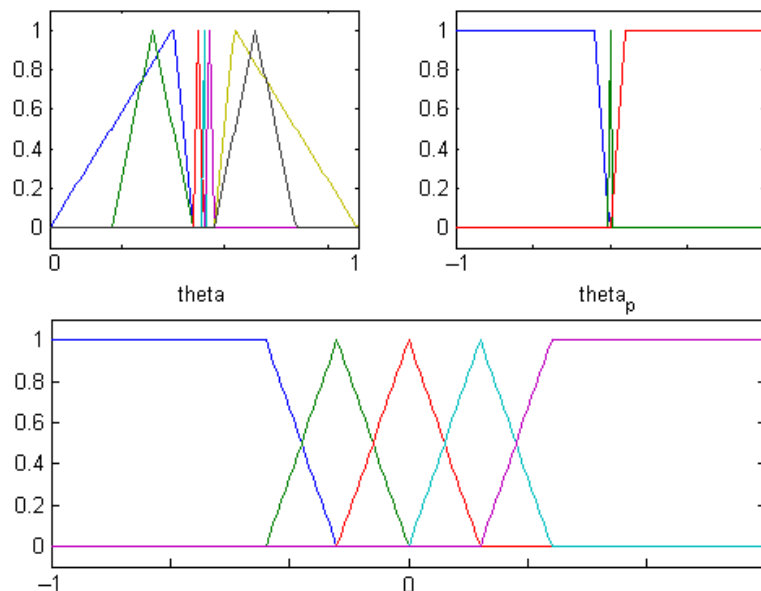
**Figure IV.2:** Simulation du balancement par retour d'état non linéaire avec et sans variations paramétrique (50%) sur la longueur  $l$  du pendule

Pour de bonnes valeurs des deux paramètres de réglage on obtient les graphes de la figure IV.3, où le bras converge en 10 à 15 secondes et avec une erreur statique de 0 à 10% sur la position du chariot (en faisant varier les paramètres de réglages). Pour une variation paramétrique sur la longueur du pendule on aura le graphe en bleu sur la figure IV.3, où des oscillations persisteront jusqu'au bout, de plus la convergence prendra plus de temps. De ce fait un réglage sur place des paramètres du système s'impose.

## 1.2. Le balancement du bras par régulation flou

Dans ces lignes, une autre commande d'érection du bras est proposée. Basée sur la logique floue. Pour un régulateur flou à deux entrées et une sortie, et avec un ensemble de règles intuitives et expérimentales tracée afin d'injecter de l'énergie au pendule à travers les déplacements successifs, dans tous les sens, du chariot. Le rail ayant une longueur limitée, mais suffisante pour effectuer le balancement vers le haut du bras. Le

pendule étant un système symétrique par ses variables d'états, et en devisant les domaines, de la variable inclinaison, ou le bras est sensé osciller, en plusieurs zones comme le montre la table des règles IV.1, ou la variable linguistique PI signifie le voisinage du point d'équilibre stable.



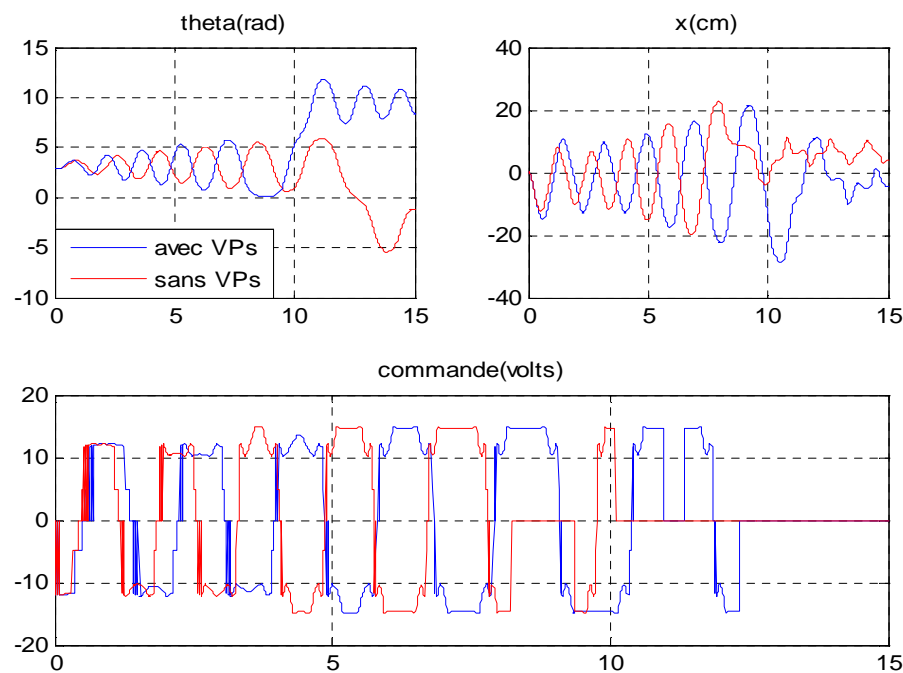
**Figure IV.3 :** base normalisé des fonctions d'appartenance du régulateur floue de soulèvement du bras du pendule

On définit les variables linguistiques décrivant l'état de l'erreur, et de sa dérivée. NP le voisinage négative, NM, et NG font référence à des zones plus éloigné de l'équilibre. Le chariot devrait accomplir un déplacement, plus ou moins, rapide dans le sens de l'inclinaison, est ainsi de suite, on finit par construire le tableau de règles IV.1, qui, basée sur les fonctions d'appartenance de la figure IV.4, donne les résultats de simulations de la figure IV.5.

**Tableau IV.1 :** base des règles de régulateur flou de balancement

	NG	NM	NP	Z	PP	PM	PG
N					P	Z	PG
Z				P			
P	NB	Z	N				

Ce qui, après différents essais par plusieurs gains de normalisation, donne les résultats de la figure IV.4, où les graphes en bleu représentent la bonne modélisation et ceux en rouge sont ceux avec une variation paramétrique de 50% sur la longueur du pendule. Théoriquement le modèle mathématique n'entre pas dans la synthèse des régulateurs flous, on voit clairement que les résultats sont proche les uns des autres, et qu'un bon réglage des paramètres de normalisation finira par donner des graphiques superposés. Les limites du rail sont respectées sans pour autant prendre en compte le déplacement et sa dérivé en entrée du régulateur, en effet il suffit de trouver le bon paramétrage des gains pour ne pas avoir de dépassement sur  $x$ . On voit très bien que les paramètres de gain influence grandement le rendement du régulateur.



**Figure IV.4 :** Simulation du balancement par logique floue avec et sans variation paramétrique de 50% sur la longueur du pendule.

Ce bon réglage, joue un rôle très important, les performances du système dépendent fortement du choix de ces paramètres. Le choix n'est pas déterminé systématiquement, mais après différents essais de simulations. On trouve dans la littérature des méthodes d'optimisation de ces gains et cela en utilisant différents algorithmes comme les

algorithmes génétiques, les algorithmes de colonies de fourmis, optimisation par essaim particulaire...et.

## 2. Commandes de Stabilisation Locale

Dans cette partie, on s'intéresse à la zone de stabilisation se trouvant au voisinage immédiat du point d'équilibre instable. La commande de stabilisation locale de ce système doit réaliser :

- La régulation de l'angle  $\theta$ , en la maintenant dans la position  $\theta(t) = 0$  rad, pour une plage de conditions initiales  $\theta_0 \in [-0.3, 0.3]rad$
- Rejet d'une perturbation d'un newton de force appliquée au pendule.
- La régulation de la position  $x$  de telle sorte qu'elle revient toujours à l'origine du longeron, en partant toujours de conditions initiales quelconques.

On examinera dans deux grands sous chapitres, la commande simple standard, en affichant les inconvénients de chacune. Dans le deuxième sous chapitre on exposera des commandes hybrides.

### 2.1. Commandes Standards

#### 2.1.1. Commande par retour d'état linéaire

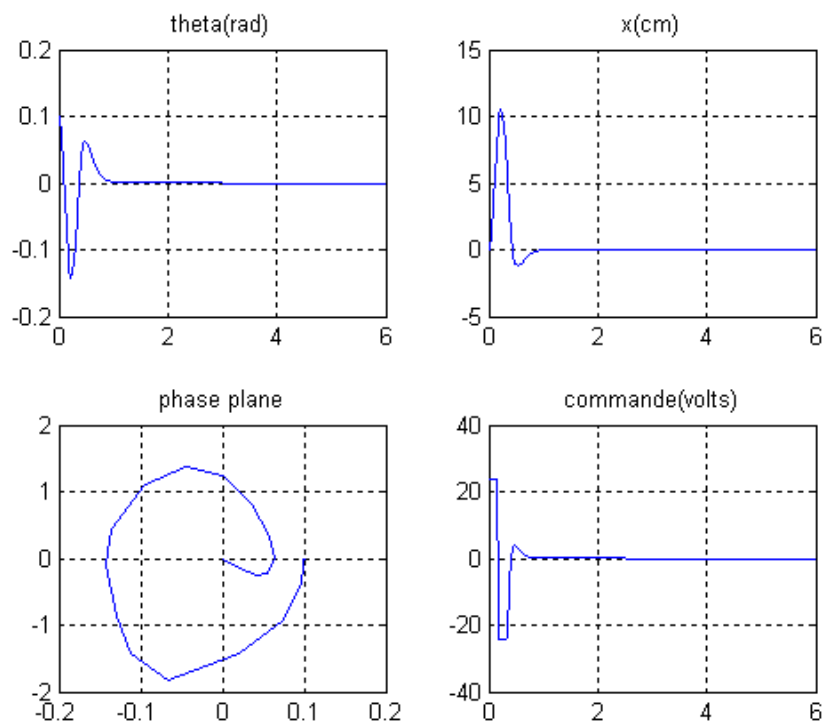
Dans ce sous titre, on abordera la commande par retour d'état linéaire. C'est la commande la plus simple qu'on peut avoir, néanmoins on verra qu'elle ne satisfait pas à certains critères de performance. Le gain de retour d'état est calculé pour les pôles  $[-10 -11 -12 -13]$ , associées au vecteur d'état  $[x \ \theta \ \dot{\theta} \ \dot{x}]$ ,

On Remarque qu'avec une condition de départ  $> 0.1rad$  la réponse sera divergente et le régulateur n'accomplit pas son objectif, et ceux avec les pôles précédents pour toutes les variables d'états du système, on dira alors que la plage des conditions initiales, où le système pouvait être stabilisé par ce genre de régulation, est très restreinte.

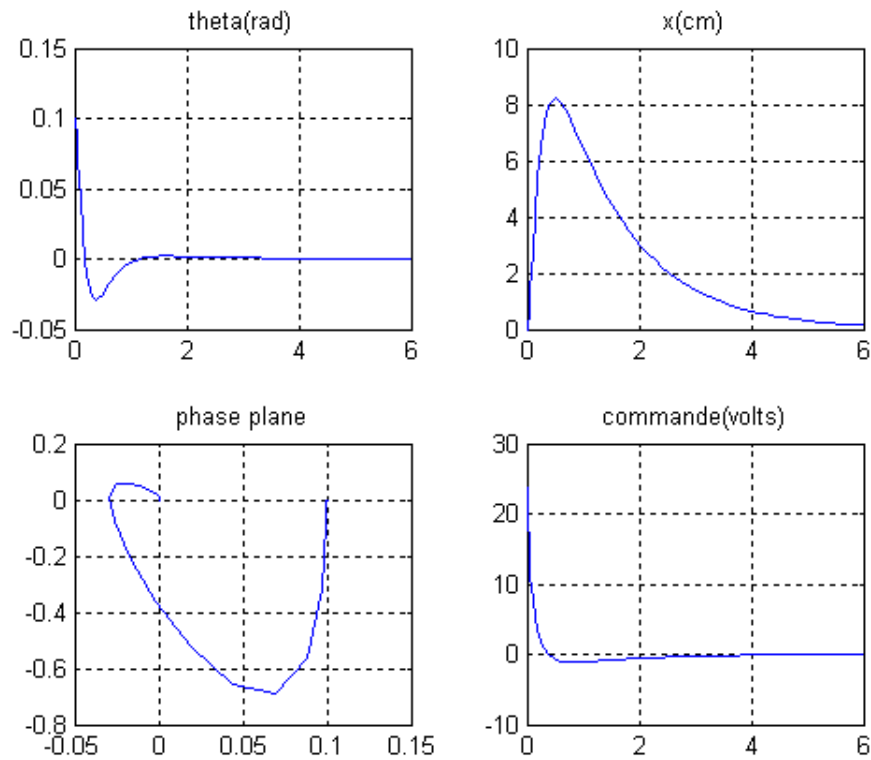
Pour un régulateur optimal type **LQR** avec un réglage du vecteur d'état  $Q = 10.I_4$  et de la commande  $R = 1$ ; et pour les mêmes conditions initiales on aura, comme le montre

la figure IV.6, une commande plus lisse, moins énergétique mais avec un temps de réponse 6 fois plus grand en  $x$  et 2 fois en  $\theta$ , le dépassement (overshooting) en  $x$  est par contre atténué de 30%, par rapport au cas précédent où les pôles sont fixés.

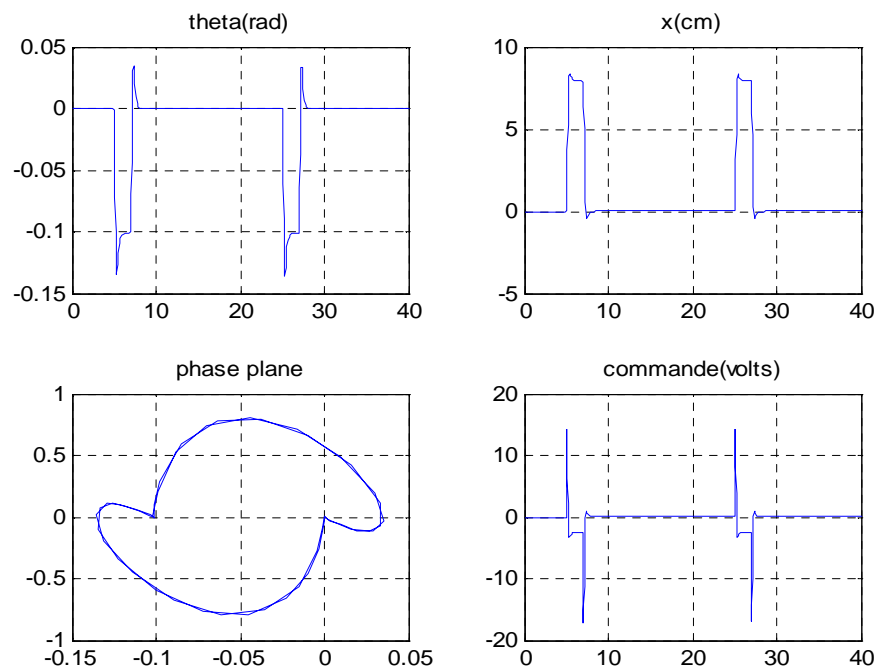
On ajoutant des perturbations impulsionnelle sur l'accélération du pendule, ce qui peut être vu comme une force, on obtient les graphes de la figure IV.9, qui montre la sensibilité de la commande par retour d'état vis à vis des perturbations extérieur. En effet une perturbation assez forte peut faire basculer les pôles des systèmes vers le mauvais cotée. Dans la simulation, de la figure IV.10, et pour des variations paramétrique de 50% sur la longueur du pendule, le système est totalement divergent, et on voit clairement l'importance de l'exactitude du model sur un régulateur par retour d'état linéaire. On note aussi qu'aucune poursuite n'est assurée avec ce genre de système de commande.



**Figure IV.5 :** Simulations pour  $\theta_0 = 0.1\text{rad}$  avec surface linéaire type retour d'état avec placement de pôles



**Figure IV.6 :** Simulations pour  $\theta_0 = 0.1\text{rad}$  avec retour d'état optimal favorisant le signal de commande sur le vecteur d'état



**Figure IV.7 :** Simulations du système perturbé pour  $\theta_0 = 0$  avec retour d'état et perturbation impulsionnelle sur le pendule

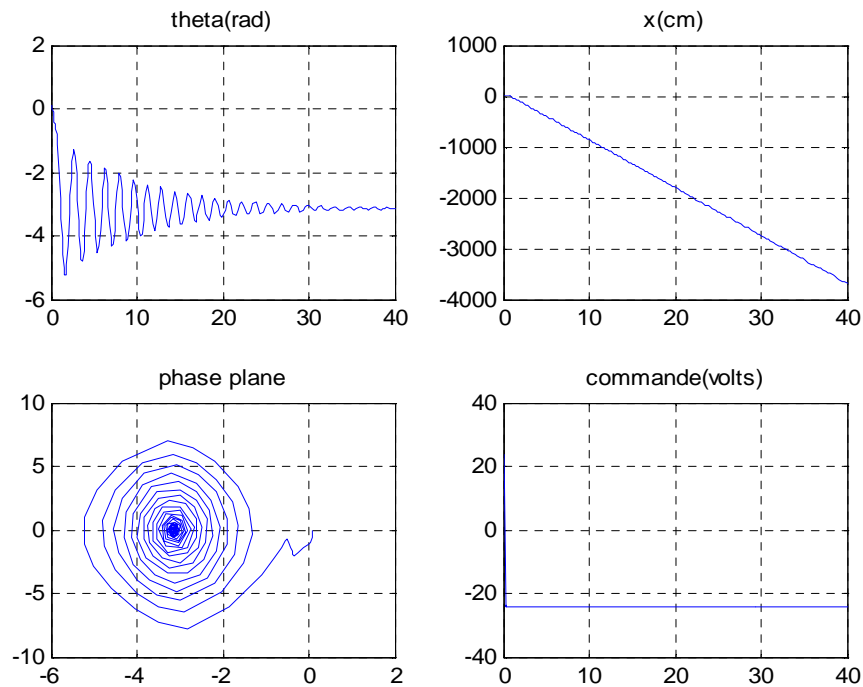


Figure IV.8 : Simulations pour  $\theta_0 = 0.1\text{rad}$  avec variations paramétriques ( $l=60\text{cm}$ )

### 2.1.2. Commande par Régulation Floue a deux Entrées

En exploitant le caractère symétrique du système à commander, en l'occurrence le pendule inversé, on déduit la symétrie de la table de décision associée au contrôleur flou à concevoir ainsi que la symétrie des fonctions d'appartenance des ensembles flous de chaque variable utilisée, par rapport à l'axe des ordonnées. La propriété de symétrie du pendule inverse est justifiée par le fait que les mouvements s'effectuent de la même manière dans les deux sens possibles de déplacement.

Les variables du contrôleur sont l'erreur et la variation de l'erreur en  $\theta$  à l'entrée et la commande du moteur en sortie. Le nombre d'ensembles flous pour chaque variable peut prendre une des valeurs possibles suivantes  $\{3, 5, 7, 9\}$  seulement. On prendra 3 ensembles pour l'erreur 3 pour sa variation, et 5 pour la commande.

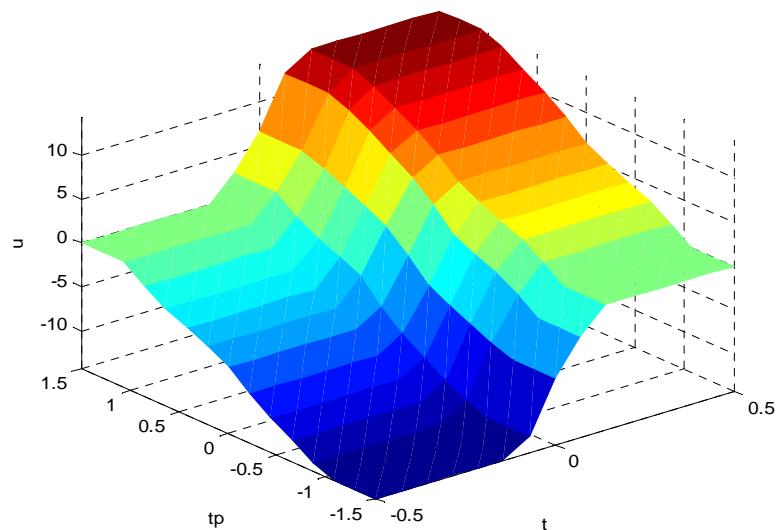
Les ensembles flous seront nommés par des désignations standard et indexés par ordre croissant, si par exemple le nombre d'ensembles flous d'une variable linguistique est égal à 5 : les ensembles flous correspondant seront : NG, NP, Z, PP, PG indexés de 1 à 5.



De plus les ensembles flous comme NG et NP seront considérés opposés à PG et PP respectivement. Tout les univers de discours seront normalisés entre -1 et +1.

**Tableau IV.2:** Règle d'inférence floue d'un régulateur 2 entrées

thêta thêta_p	N	Z	P
N	NM	NP	Z
Z	NP	Z	PP
P	Z	PP	PM

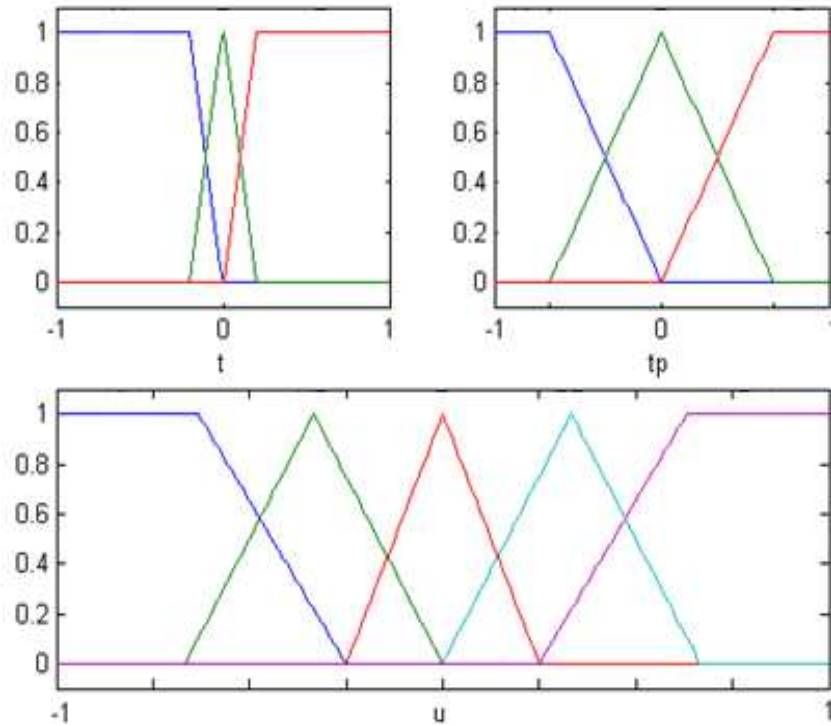


**Figure IV.9:** surface de commande.

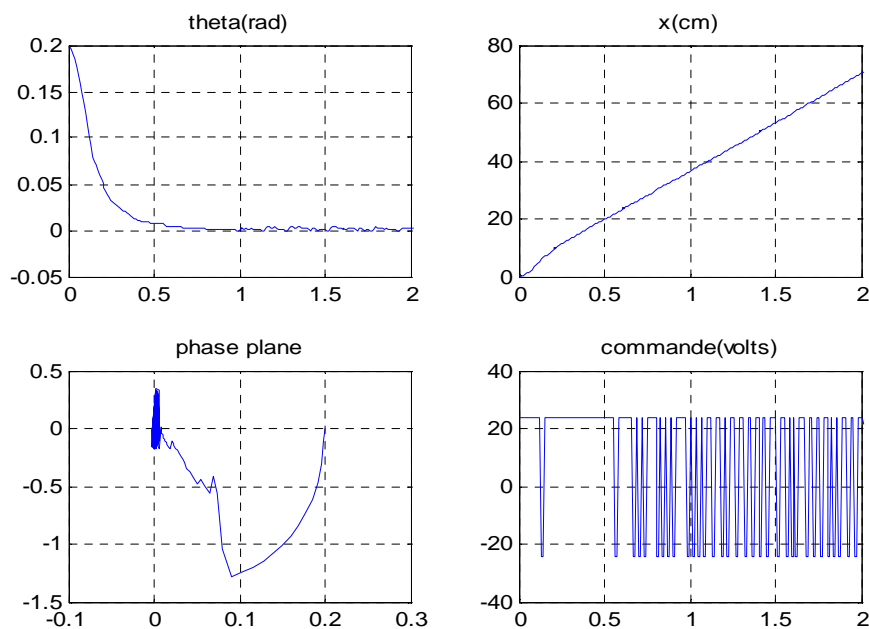
A cette base de connaissance on associe la base de données de la figure IV.10, concernant les fonctions d'appartenance en entrée et en sortie du régulateur. Une première simulation donne les résultats sur la figure IV.11, où on aura une convergence sur  $\theta$  dans un champ d'attraction des conditions initiales de  $\pm 0.2rad$ , néanmoins on aura une commande très énergétique avec déplacement incontrôlable.

Dans la figure IV.12, on voit le rejet de perturbations sur le pendule, et qu'une perturbation d'un newton décalera le pendule par  $0.1rad$ , ceci dit, si une force perturbatrice fait sortir le pendule de la limite de stabilité, on le verra déstabiliser. La commande par régulation flou est indépendante du modèle, la simulation de la figure

IV.13, montre cette propriété, en imposant une variation paramétrique sur la longueur du pendule, la convergence est comme même assurée.



**Figure IV.10:** fonction d'appartenance décrivant un régulateur flou Mamdani



**Figure IV.11 :** Simulation d'une régulation floue à deux entrées avec  $\theta_0 = 0.2rad$

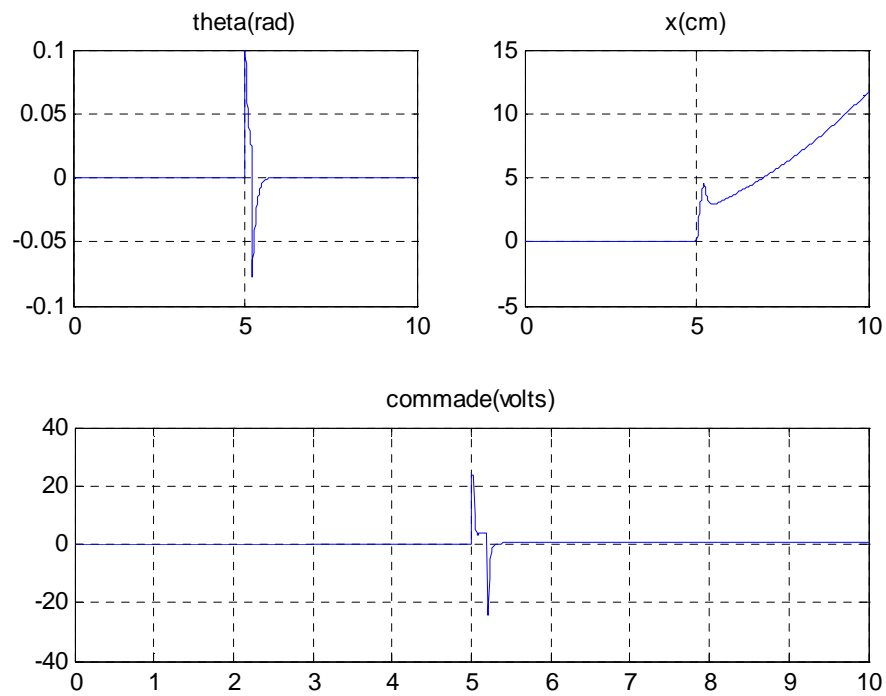


Figure IV.12: Perturbation impulsionnelle sur le pendule d'amplitude maximum 1N

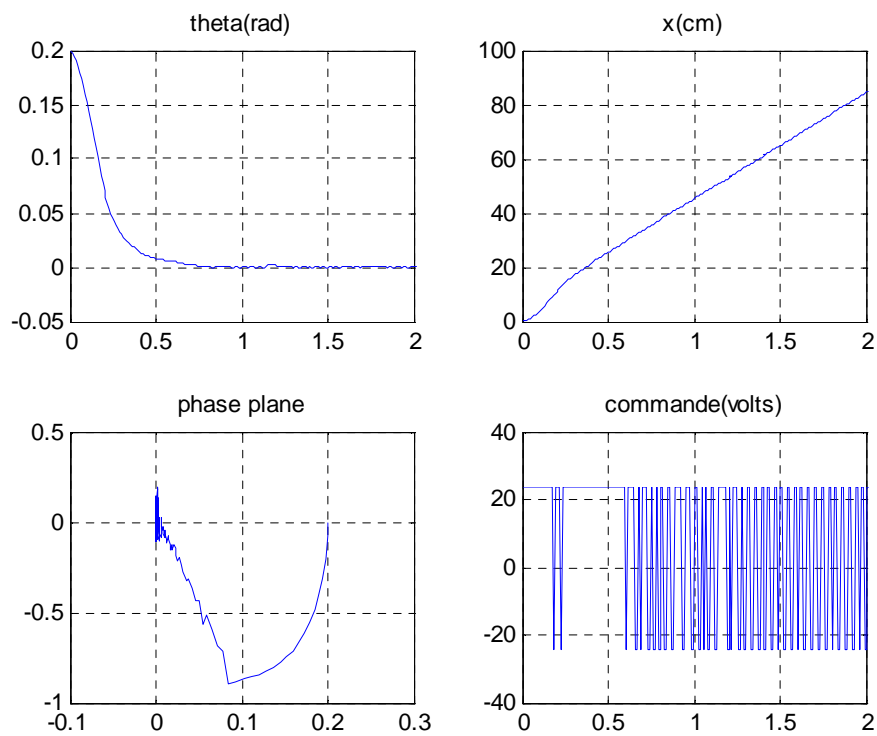


Figure IV.13 : Pour une variation paramétrique sur la longueur du pendule

### 2.1.3. Stabilisation par régulation floue à quartes entrées

Dans l'expérience précédente, on a notamment repéré le défaut concernant le dépassement très important en  $x$ , on peut y remédier en rajoutant, au régulateur précédent, un nouveau régulateur flou ayant le dépassement et sa dérivée en entrée, avec la base de données de la figure V.14. Le schéma de la commande par régulateur flou découplé, ou a quatre entrées, est donné par la figure V.15. La simulation de la réponse du système est donnée par la figure IV.16. Cette commande assure une plage de stabilisation de  $\pm 0.2rad$ , avec rejet des perturbations extérieurs, et une bonne robustesse aux changements paramétriques, cette dernière propriété est justifiés par le fait qu'un régulateur flou est structurellement indépendant des paramètres du modèle, ceci dit un rajustement des gains de normalisations doit être refait.

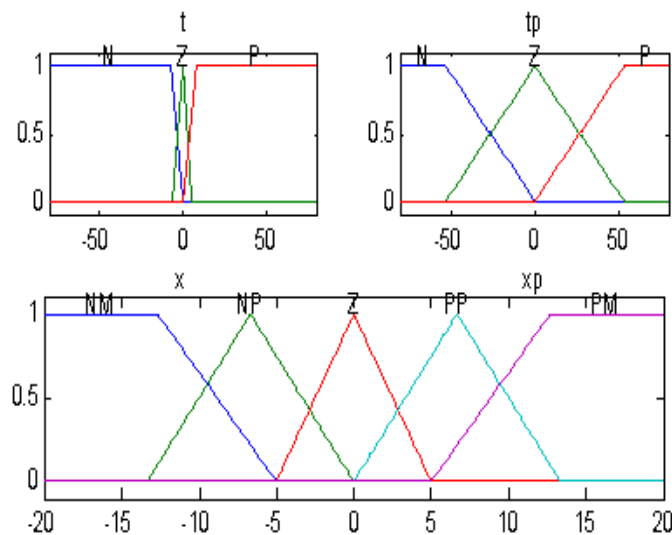


Figure IV.14 : Base de donnée du régulateur quatre entrées

Tableau IV.3 : règles du régulateur flou du déplacement pour 4 entrées 1 sortie

$x \backslash xp$	$N$	$Z$	$P$
$N$	NM	NP	Z
$Z$	NP	Z	PP
$P$	Z	PP	PM

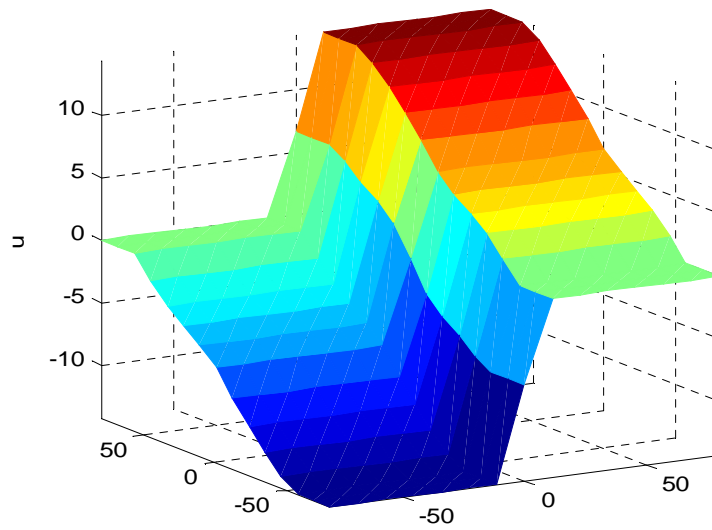


Figure IV.15 : surface de commande rajoutée de

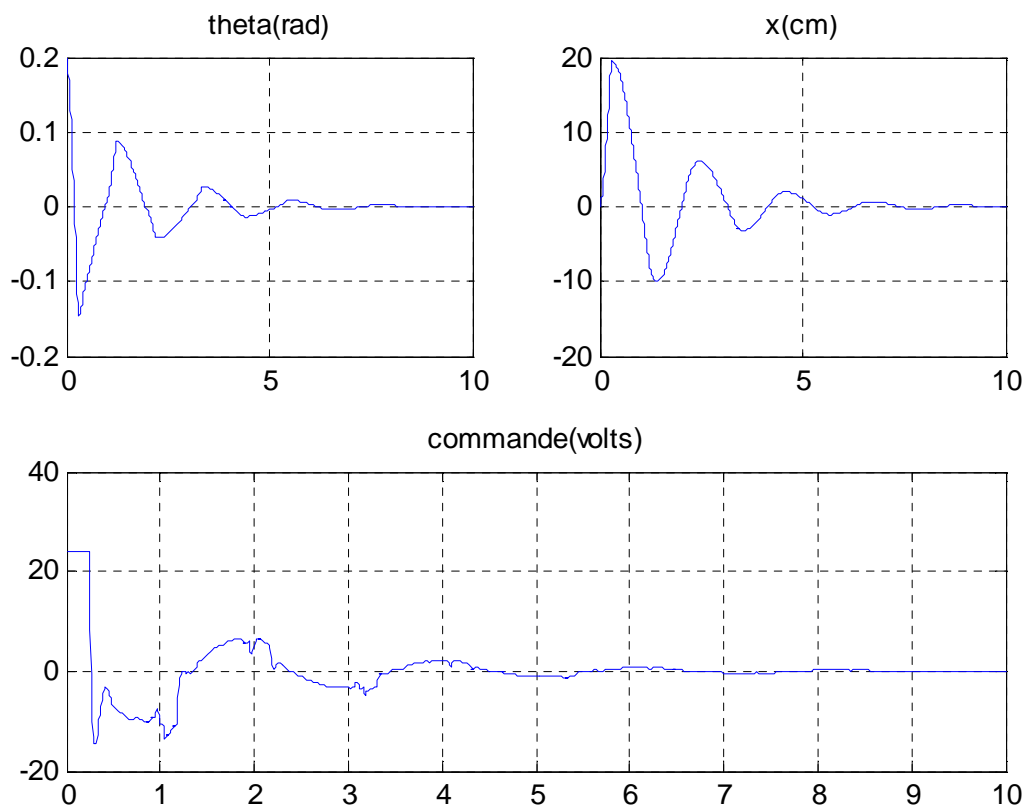
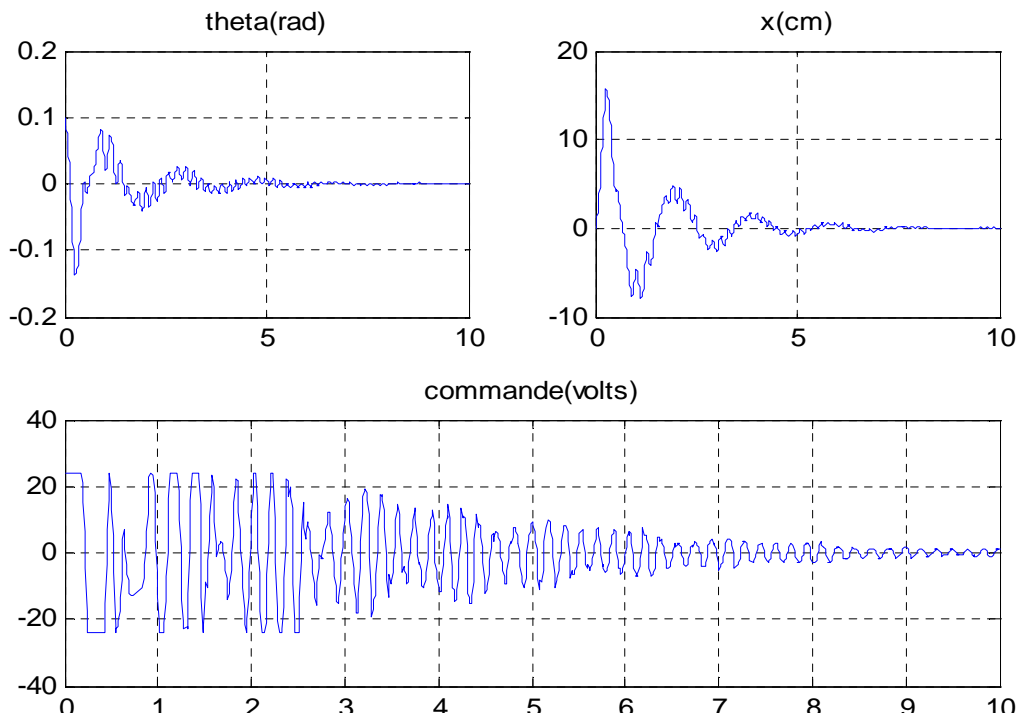
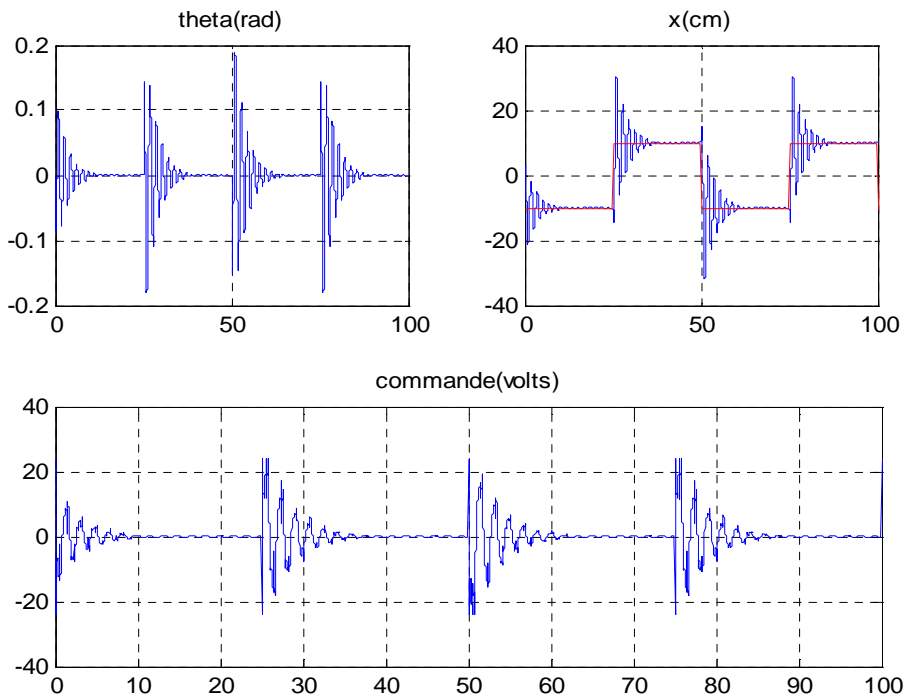


Figure IV. 16: Résultats de simulation pour la Régulation floue 4 entrées  $\theta_0 = 0.2rad$



**Figure IV.17 :** Simulation pour  $\theta_0 = 0.2rad$  avec variation paramétrique de 50% sur la longueur du pendule



**Figure V.18 :** Simulation de poursuite de  $x$  d'un signal carré  $\pm 10cm$

### 2.1.4. Stabilisation par modes de glissement

Dans ces lignes nous introduisons, pour le pendule inverse, un contrôleur stabilisant par modes de glissement, on verra ces propriétés, qualités et défauts, dans le chapitre des commande hybrides on décrira une des variantes de la commande hybride : le floue glissant, qui élimine les défauts du système de commande en gardant ces qualités initiales de robustesse. Le balancement vers le haut du pendule est effectué par retour d'état non linéaire synthétisé dans un chapitre précédent. Les contraintes qui nous sont imposées, par une éventuelle expérimentation, sont la trajectoire du chariot qui ne doit dépasser les limites du rail, et qui doit retourner au point de départ, et de la limite en tension acceptable par le moteur utilisée.

Comme vu dans le chapitre du rappel théorique, la commande par MDG prend la forme suivante :

$$u = u_{eq} + u_s$$

La commande  $u_{eq}$  étant la commande équivalente qui permet de ramener la trajectoire d'état du système le plus proche de la surface choisie, ayant les équations suivantes du système:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(x) + g_1(x)V_c \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(x) + g_2(x)V_c \end{cases}$$

Avec:

$$f_1(x) = \frac{-m_p g \sin(x_3) \cos(x_3) + m_p l x_4^2 \sin(x_3) - x_2(\alpha + H_2)}{M - m_p \cos^2(x_3)}$$

$$g_1(x) = \frac{H_1}{M - m_p \cos^2(x_3)}$$

$$f_2(x) = \frac{Mg \sin(x_3) - m_p l x_4^2 \sin(x_3) \cos(x_3) + x_2(\alpha + H_2) \cos(x_3)}{l.(M - m_p \cos^2(x_3))}$$

$$g_2(x) = \frac{-H_1 \cos(x_3)}{l.(M - m_p \cos^2(x_3))}$$

$$x = [x_1 \quad x_2 \quad x_3 \quad x_4]^T$$

La surface de la forme linéaire vérifie:

$$S(x) = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \lambda_4 x_4$$

$$S(x) = \dot{S}(x) = 0$$

On retrouve la commande:

$$u = \frac{-\lambda_1 x_2 - \lambda_2 f_1(x) - \lambda_3 x_4 - \lambda_4 f_2(x)}{\lambda_2 g_1(x) + \lambda_4 g_2(x)} + K_G \operatorname{sgn}(S)$$

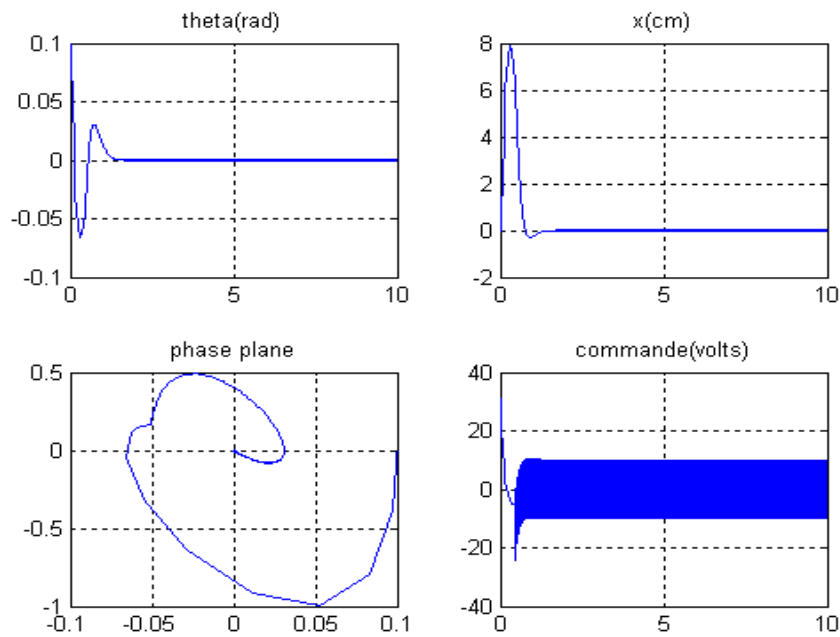
Et en fixant le vecteur  $\lambda$  comme gain de retour d'état avec les pôles [-10 -11 -12 -13], on obtient les résultats suivant de simulation. Le Système peut aller jusqu'à une valeur de  $\theta_0$  entre 0.4 et 0.5 radians, le correcteur stabilisera le système vers le point d'équilibre.

On remarque notamment l'énergétique de la commande du moteur qui lui est très néfaste, une des méthodes pour l'éliminer a été présentée dans le sous-chapitre suivant. A part ceci, la robustesse de la commande par mode de glissement vis-à-vis des variations paramétrique et la sensibilité du système bouclée en terme de rejet de perturbation sont prouvées dans les graphiques des figures V.22 et V.23 respectivement. Ces dernières propriétés étant manquantes dans la commande par retour d'état classique ou optimal. Néanmoins un fort dépassement en  $x$  est a signalé pour les grandes erreurs initiales. Avec une large plage de conditions initiales allons de -0.4rad a +0.4rad.

Une trajectoire d'état en  $\theta$  moins longue, mais au dépend d'un déplacement prolongé avec moins de dépassement. On retrouve les caractéristiques du retour d'états optimal. On fixant les pôles on obtient des résultats plus désirables malgré quelque insuffisance. Ainsi toutes les simulations suivantes relatives aux modes de glissement, et tous autres commande qui lui lié, la surface sera déterminée par cette méthode.

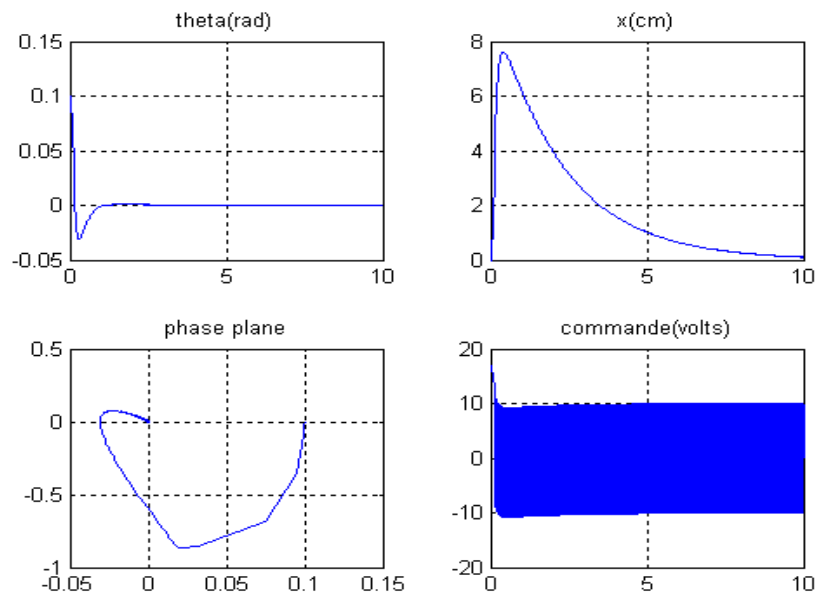
On aura une réponse oscillante, et avec une plage de stabilisation diminuer par 20%. Et on n'a pas de champ d'action pour éliminer cette oscillation. On verra que la commande hybride peut assurer ces même propriétés, et sans chattering.



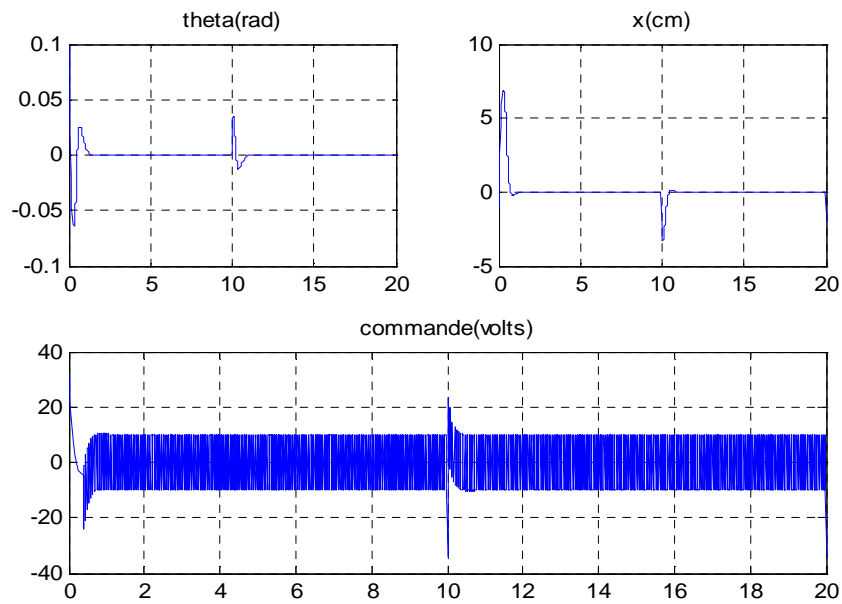


**Figure IV.19 :** Simulation des MDG pour  $\lambda$  par retour d'état avec  $\theta_0 = 0.1rad$

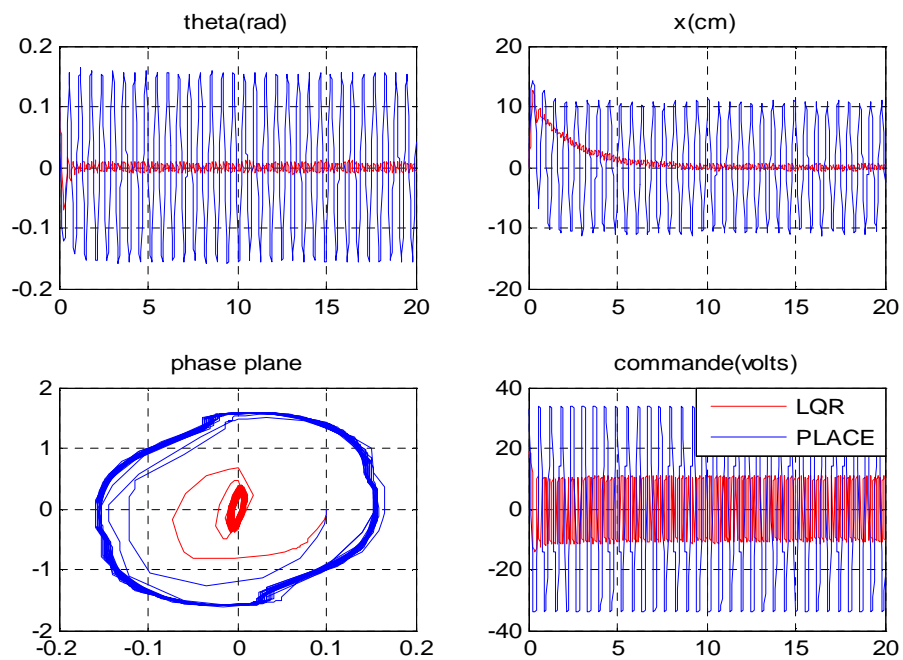
Le régulateur assure aussi une bonne poursuite de référence en  $x$  comme on peut le constater sur la figure IV.24. On peut aussi noter que la commande obtenant est lisse et l'angle  $\theta$  varie faiblement.



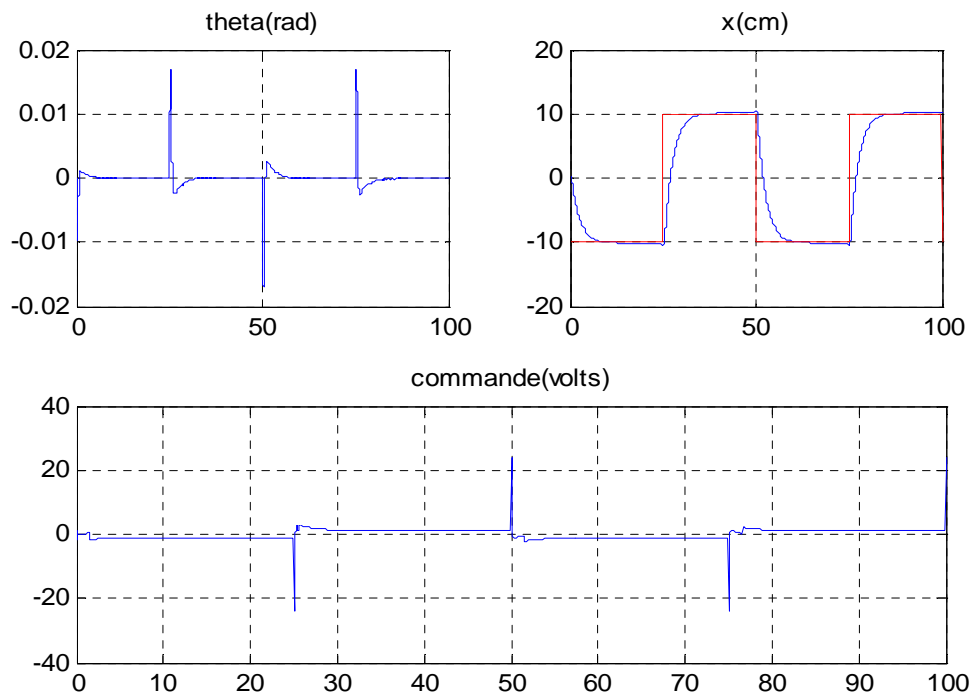
**Figure IV.20:** Simulation pour  $\lambda$  par retour d'état optimal en trajectoire d'état avec  $\theta_0 = 0.1rad$



**Figure IV.21:** Simulation du système perturbé par force-impulsion de 1N avec  $\theta_0 = 0.1rad$



**Figure IV.22 :** Simulations Pour variations paramétriques de 50% sur la longueur  $l$  du bras du pendule inverse, avec une surface par placement de pôles (bleu) et avec placement optimales des pôles (rouge)



**Figure IV.23 :** Simulation de la poursuite par  $x$  d'un signal carré, avec surface linéaire déterminé par calcul d'un retour d'état optimal

## 2.2. Commande hybrides

Dans cette partie du chapitre on verra deux commandes hybrides, à base des commandes standards précédemment présentés. Dans un premier temps, on mettra en œuvre la commande floue glissante, des analyses de robustesse de stabilité de sensibilité et de poursuite de trajectoire pour le déplacement, seront entreprises. Dans un second temps, on verra la commande NeuroFloue du type ANFIS.

### 2.2.1. Stabilisation par mode de glissement flou

Pour la commande floue on remplace la partie  $\text{sgn}(S)$  dans la commande avec MDG par  $\text{sat}\left(\frac{S}{\phi}\right)$  ou  $\phi > 0$  est issue d'un traitement flou de la surface de glissement  $S$  avec les règles suivantes:

- (1) *si : S est NG alors :  $\phi$  est G .*
- (2) *si : S est NP alors :  $\phi$  est P .*
- (3) *si : S est ZE alors :  $\phi$  est Z .*
- (4) *si : S est PP alors :  $\phi$  est P .*
- (5) *si : S est PG alors :  $\phi$  est G .*

Ou les variables linguistiques N, G, Z signifie Négatif, Grand et Zéro et ainsi de suite. On peut alors écrire :

$$u = \frac{-\lambda_1 x_2 - \lambda_2 f_1(x) - \lambda_3 x_4 - \lambda_4 f_2(x)}{\lambda_2 g_1(x) + \lambda_4 g_2(x)} + K_{GF} \text{sat} \left( \frac{S}{\phi} \right)$$

Le vecteur  $\lambda$  sera calculé soit par retour d'états, soit par retour d'état optimal linéaire autour de l'équilibre désiré. Sauf mention contraire  $\lambda$  sera déterminé Par la première méthode. Apres plusieurs essaie, on peut allez jusqu'à stabiliser pour une valeur initiales en  $\theta_0 = 0.3rad$  .Pour  $\theta_0 = 0.3rad$  on aura les résultats de la figure IV.23.

Apres introduction de variables paramétriques sur le système, On aura une plage de stabilisation plus fine, de plus comme le montre la figure IV.25, il va y avoir réapparition du chattring. Deux option alors s'offre a nous, soit on continue comme précédemment de faire le réglage des gains de résolution du régulateur flou (figure IV.26), soit on change les paramètres de la surface de glissement (figure IV.27).

A cause de la limitation sur la tension de commande, le MDGF a comme majeur inconvenant la non stabilisation des grandes inclinaisons de départ (conditions initiales), ces inclinaisons sont stabilisable par MDG standard, néanmoins la grande contribution dans cette technique de floue glissant est la suppression du phénomène de chattring sur la commande qui devient lisse et beaucoup moins énergétique.

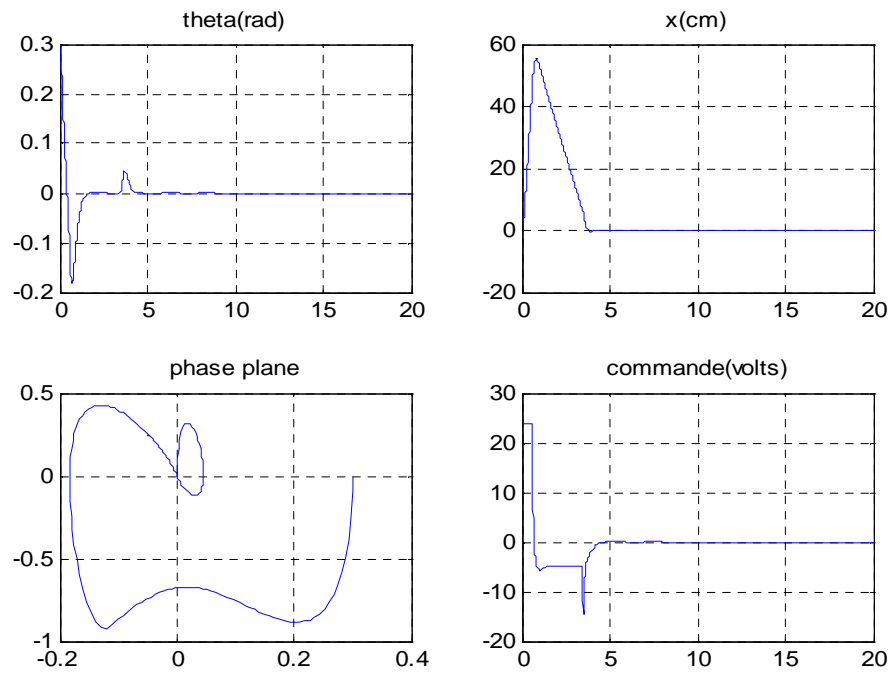


Figure IV.24 : Simulations du mode de glissement flou pour  $\theta_0 = 0.3rad$

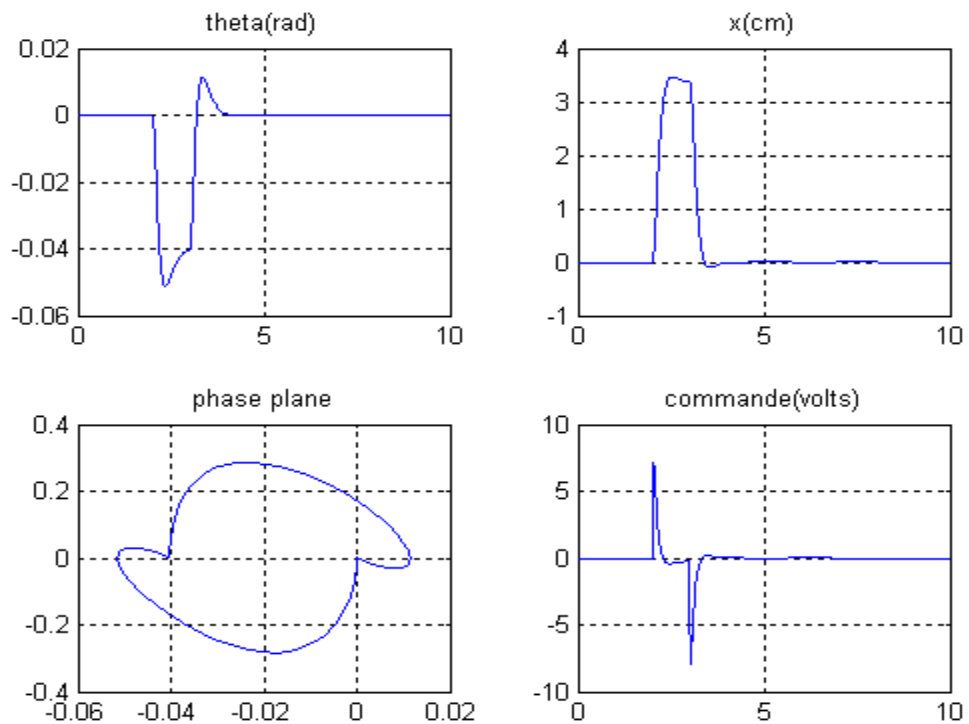
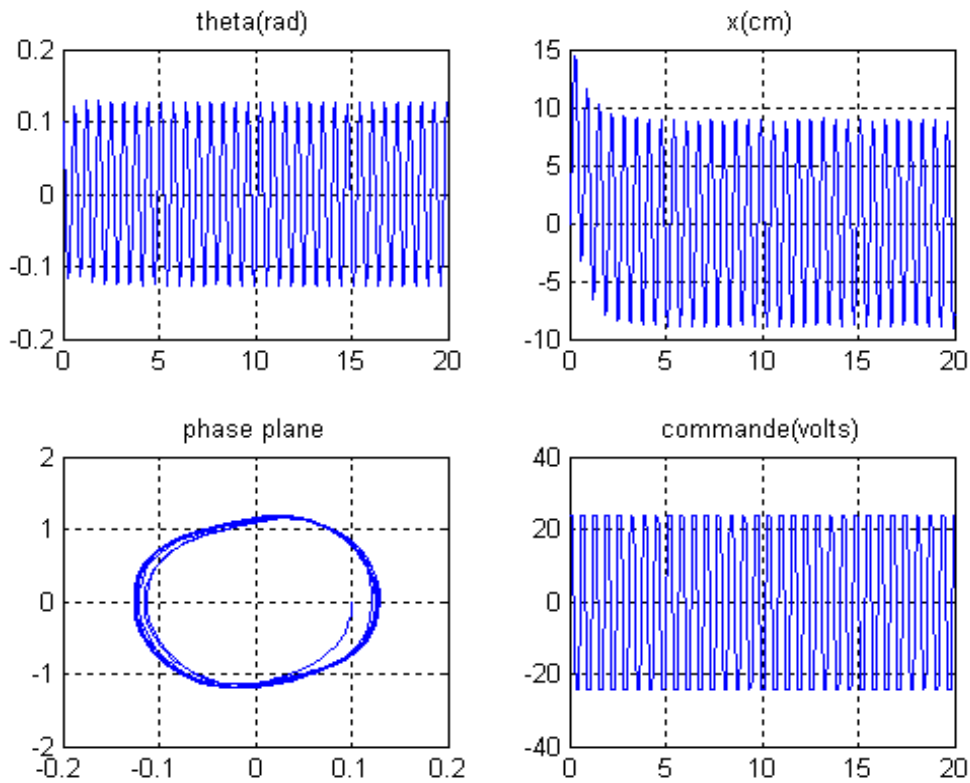
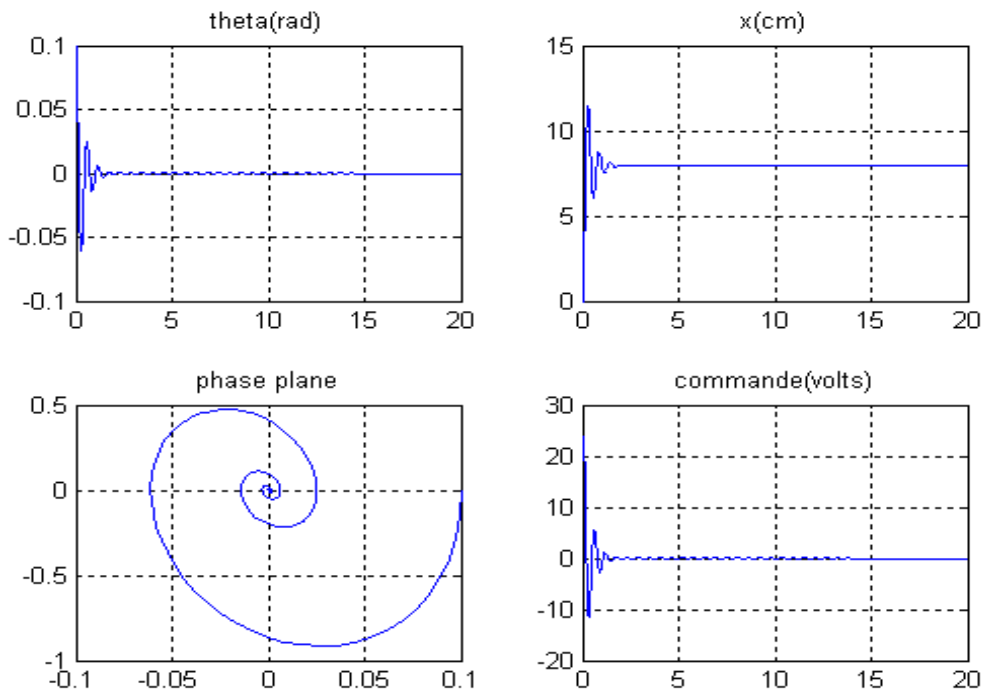


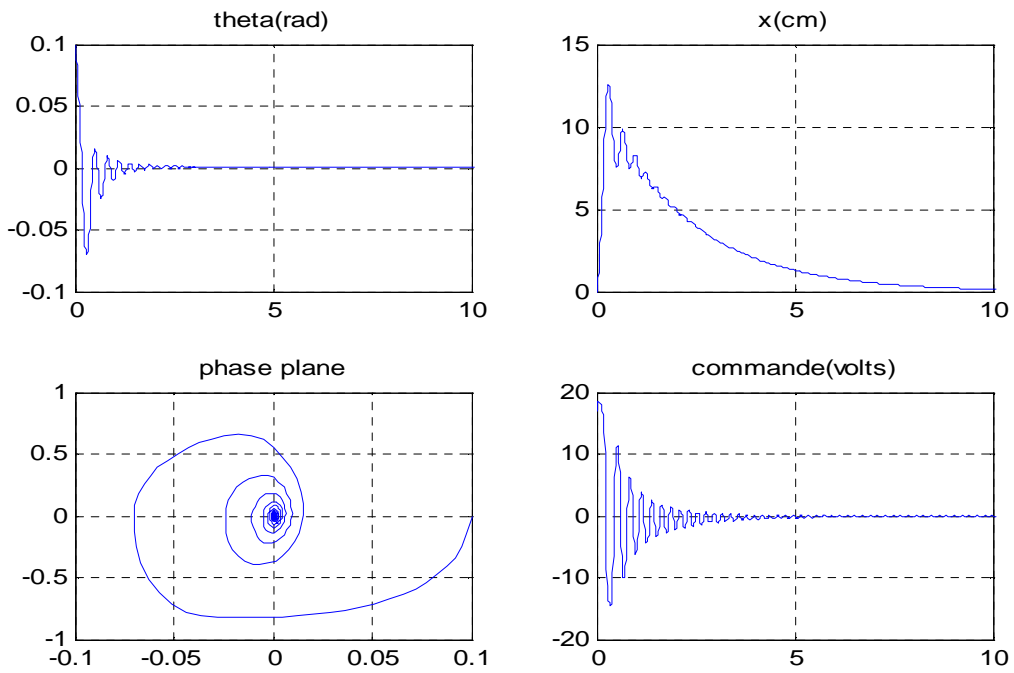
Figure IV.25 : Simulation pour rejet de perturbation impulsionnelle d'amplitude



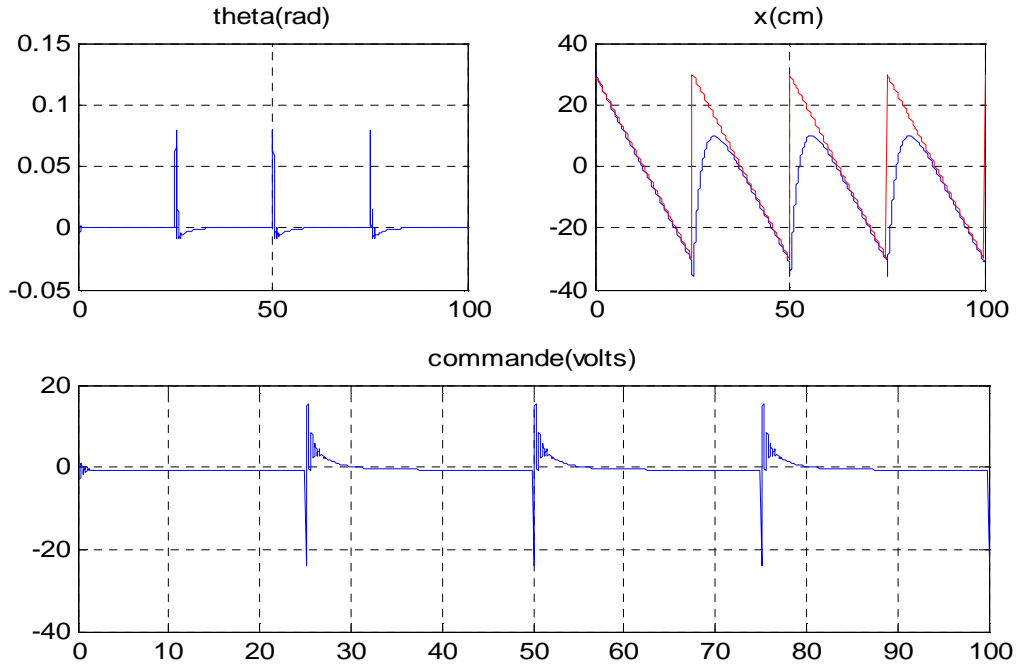
**Figure IV.26 :** Simulations pour variations paramétriques de 50% sur  $I$



**Figure IV.27 :** Simulations de la Variation Paramétrique de  $I$  par 50% pour différent gains de pondération du traitement flou de la surface



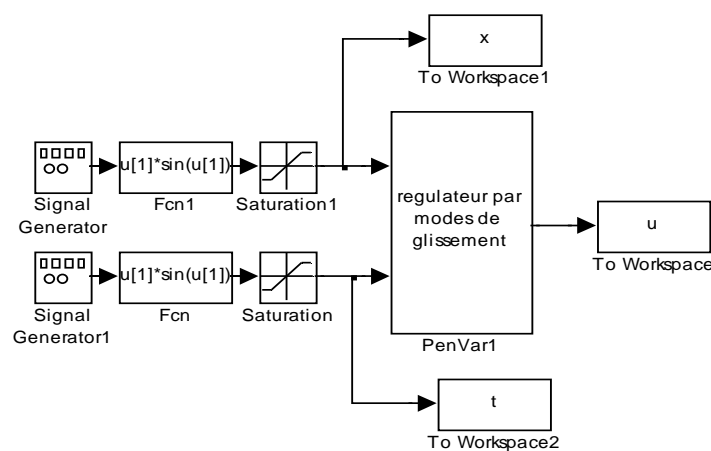
**Figure IV.28 :** Simulation des variations paramétriques pour surface linéaire avec  $\lambda$  déterminé par retour d'état optimal



**Figure IV.29 :** Simulation pour poursuite de  $x$  d'une dent de scie

### 2.2.2. Optimisation NeuroFloue type ANFIS

La commande ANFIS est basée sur l'apprentissage des ensembles flous et des règles de commandes, et ceux soit en optimisant un réseau floue d'inférence, soit en créant un nouveau qui interpolera tous les données d'apprentissage qui lui sont fournies en phase d'apprentissage. Le plus que cette méthode offre, c'est qu'après ces règles et ensembles peuvent être retouchés selon le bon vouloir de l'utilisateur.



**Figure IV.30** : Schéma Simulink montrant la méthode (normalement expérimentale-ici seule la simulation est réalisée) d'acquisition des données d'apprentissage

Il s'agit d'une sorte d'optimisation des algorithmes de commandes déjà utilisés dans ce mémoire, elles permettront comme on verra par la suite, d'éliminer certaines limitations du contrôleur d'origine, comme le chattering, le temps de réponse, le dépassement, ou le temps de montée.

#### a. Optimisation du régulateur par modes de glissements

Il faut dans un premier temps utiliser le régulateur précédemment mis au point dans la partie IV.1.2.d afin de générer un ensemble de données d'apprentissage et un autre ensemble de données servant pour la vérification des données précédentes (checking data), comportant les 4 entrées d'état et la sortie tension de commande, et ceux suivant le diagramme de la figure IV.31. Ceci fait, on utilisera l'outil anfisedit de Matlab pour



mettre en œuvre l'architecture du correcteur flou neuronal optimisant la commande d'origine. Le tableau IV.4 montre pour différents paramètres de réglage (de l'algorithme). Les résultats obtenus de l'erreur en fin de phase d'apprentissage avec le nombre d'époques nécessaire à la convergence de l'algorithme.

**Tableau IV.4 :** données associées a différent essais de simulations

Sphère d'influence	Méthode d'optimisation	Nombre d'époque	Erreur d'entraînement	Erreur de validation
0.5	Hybride	6	4.2038	6.7859
0.3	Hybride	4	3.0427	6.1223
0.29	Hybride	3	2.8414	11.5351
0.28	Hybride	2	3.2393	10.4151

La figure IV.32 montre les résultats de simulation pour une sphère d'influence de 0.3, où, comme le montre le Tableau IV.4, l'erreur sur les données de validations est la plus faible, avec une erreur d'entraînement acceptable. La convergence et la plage d'attractivité en condition initiales, de la variable non actionnée  $\theta$ , sont fidèle à l'algorithme d'origine de la commande par modes de glissement, néanmoins on voit qu'un dépassement de la limite du rail, et une erreur statique assez importante sur la position du chariot, apparaissent. Pour remédier à cette insuffisance d'autre réglage des paramètres s'impose. Quoiqu'un système d'inférence NeuroFloue à 4 entrées est difficile à l'entraînement.

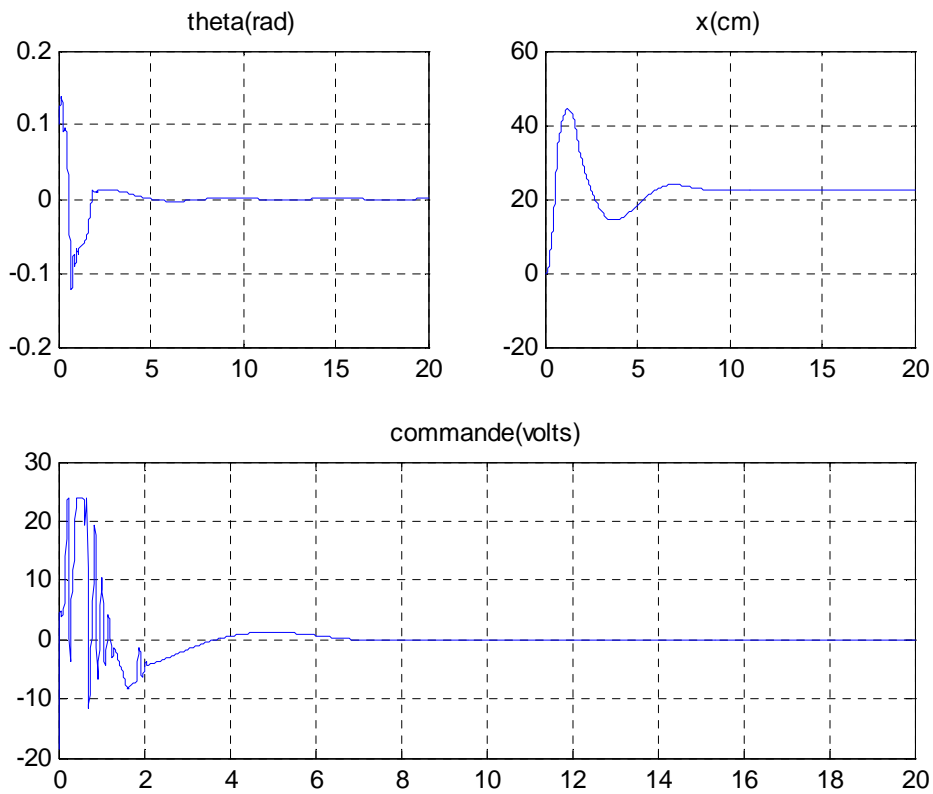


Figure IV.31 : Simulation pour neuro-floue avec  $ROI=0.3$

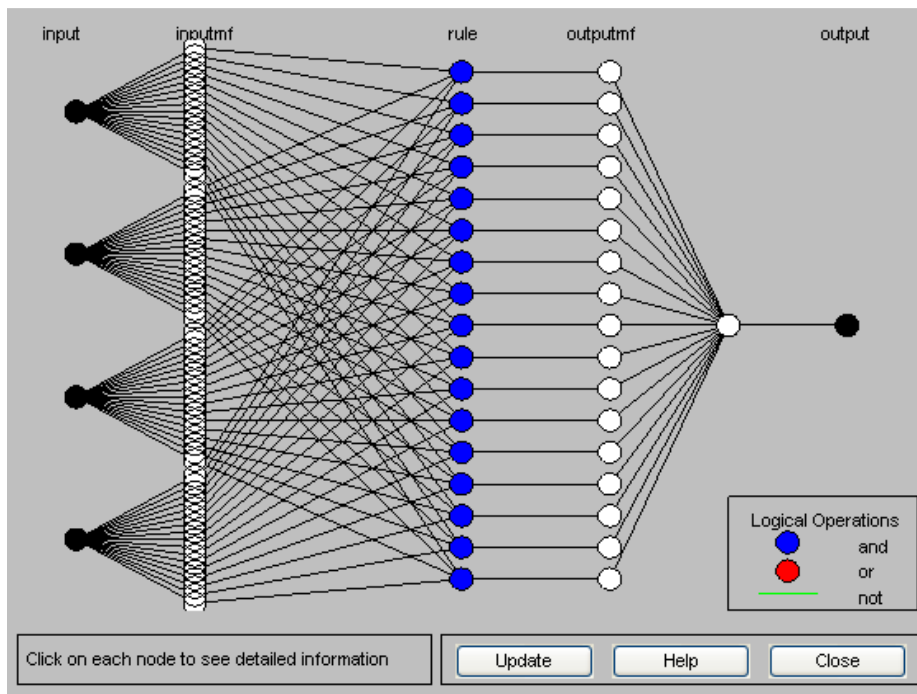


Figure IV.32 : Le réseau d'inférence floue neuronale à 4 entrées pour  $ROI=0.3$

## b. Optimisation du Régulateur Floue Découplé

Les premières simulations du régulateur flou découplé sont assez suffisantes, mais une optimisation des temps de réponse et de l'annulation des vibrations sur le bras et sur le chariot et aussi possible, en modifiant les règles et les fonctions d'appartenance de chacun des deux régulateur floue. Avec l'ANFIS une optimisation de ces règles et fonctions est possibles, en injectant dans l'outil anfisedit les données expérimentales d'entraînement qu'on voit dans les tableaux IV.5 et IV.6. La sphère d'influence ROI=0.3 assure une erreur de validation et d'entraînement assez faible pour le régulateur NeuroFloue de déplacement. De même le régulateur d'inclinaison est optimal pour ROI=0.2. On obtient les graphes de la figure IV.34.

**Tableau IV.5 :** résultats d'apprentissage pour différente sphères d'influence pour le régulateur flou numéro 1 pour la régulation du déplacement.

Sphère d'influence	méthode	Nombre d'époque	Erreur de validation	Erreur d'entraînement
0.5	Hybride	6	12.2468	14.4215
0.4	Hybride	2	10.3245	12.9648
0.3	Hybride	5	5.4587	6.4245
0.2	Hybride	5	4.8973	11.2145
0.1	Hybride	1	1.8524	16.2876

**Tableau IV.6 :** résultats d'apprentissage pour différente sphères d'influence pour le régulateur flou numéro 2 pour la régulation de l'inclinaison.

Sphère d'influence	méthode	Nombre d'époque	Erreur de validation	Erreur de d'entraînement
0.5	Hybride	2	5.4156	3.9472
0.4	Hybride	2	4.2415	2.9648
0.3	Hybride	2	2.7489	1.9893
0.2	Hybride	2	2.0747	1.4842
0.1	Hybride	2	2.8557	0.3542

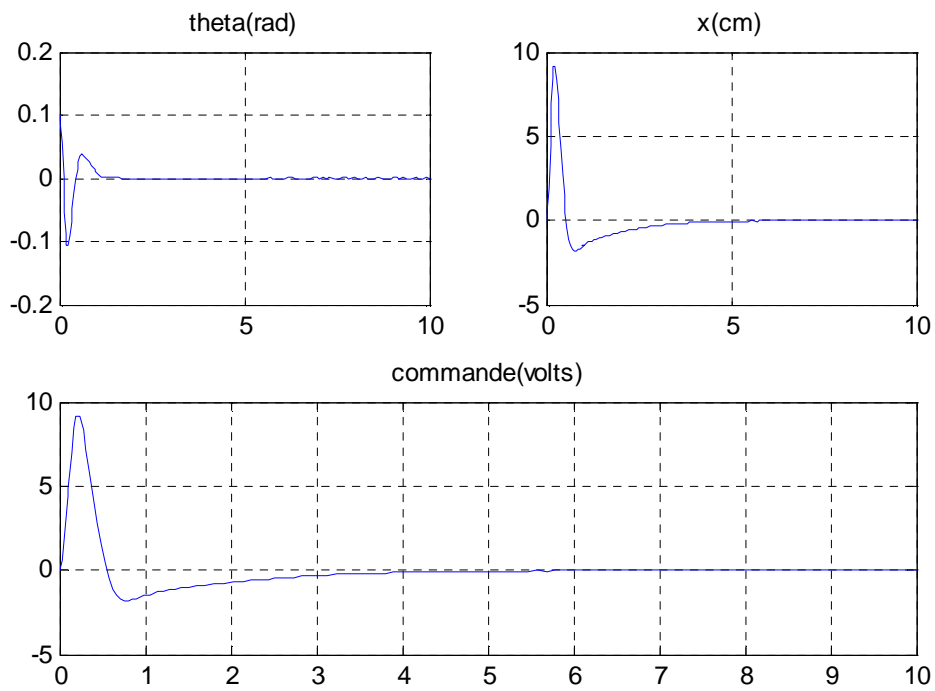


Figure IV.33 : résultats obtenue pour l'optimisation d'un régulateur flou découplé

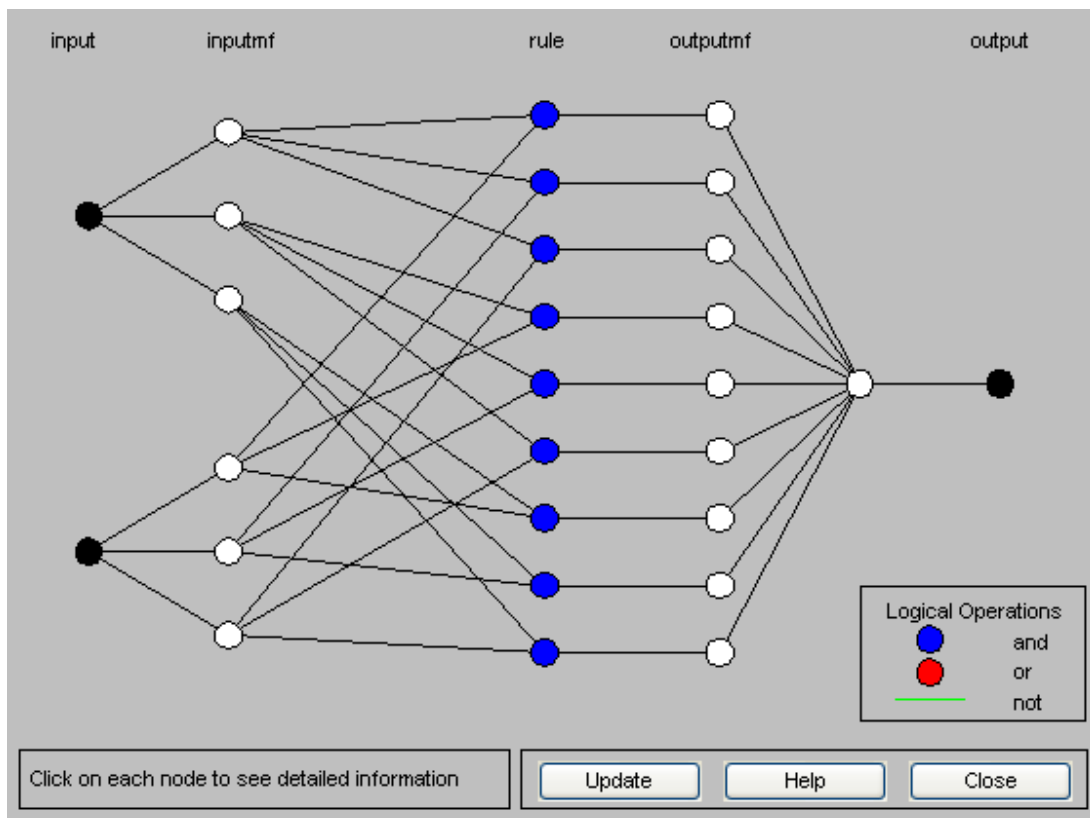


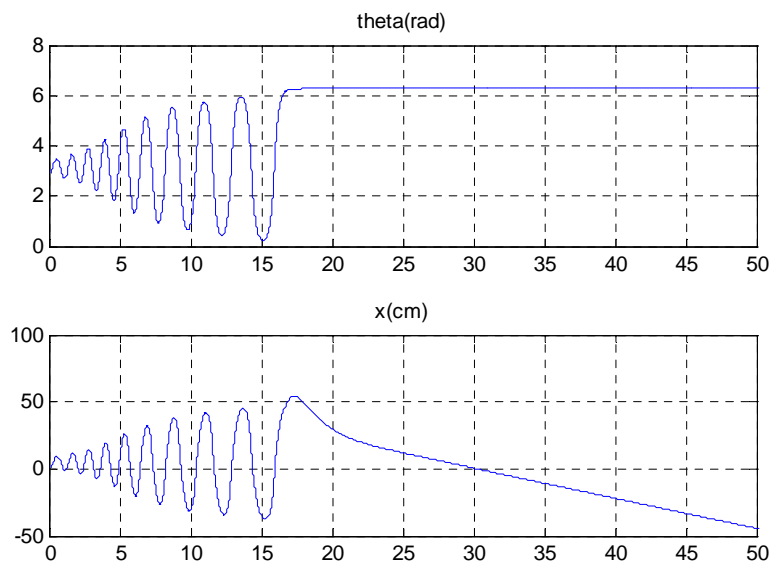
Figure IV.34 : Structure des régulateurs ANFIS utilisé

### 3. Commandes final à expérimenter

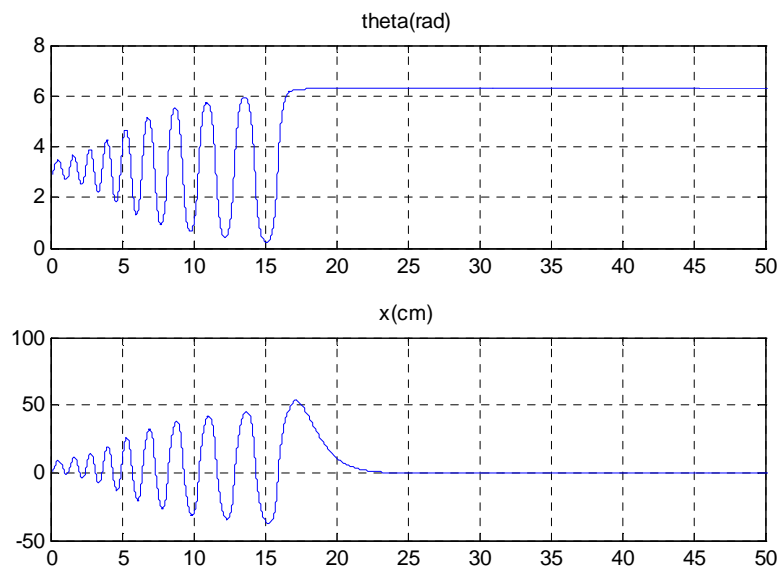
Dans ce paragraphe, on va voir les commandes finalisées (léviation puis stabilisation du pendule et du chariot). On utilisera une technique de léviation beaucoup plus simple et intuitif, il s'agit de la commande en état de résonance, il suffit d'attaquer le système avec une tension proportionnelle à la vitesse angulaire du pendule. Pour une valeur optimale de cette constante, on aura les résultats de la figure IV.35. On note particulièrement que la commande par mode de glissement utilisée ici n'assure pas à elle seul une convergence de la position vers la position initiale. Il faudra donc intégrer une action intégrale sur la position du chariot, la surface deviendra:

$$S(x) = K \cdot x - K_i \cdot \int_0^t x_1(\tau) d\tau$$

Les gains  $[K \quad K_i]$  seront ensuite déterminés par la fonction acker de Matlab, et ceux pour un système linéaire augmenté par l'action intégrale. Les résultats obtenus pour des pôles petits sont donnés dans la figure IV.37.

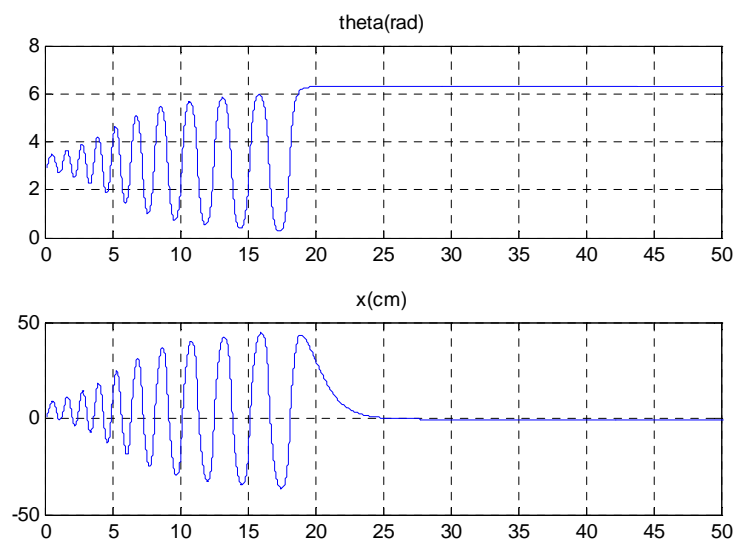


**Figure IV.35 :** Simulation de la léviation par action résonante puis stabilisation par mode de glissement simple



**Figure IV.36 :** Simulation de la léviation par action résonante puis stabilisation par mode de glissement avec action intégrale sur le dépassement

La commande précédente étant non réalisable à cause du dépassement de +10cm sur les limites de positions du chariot, on peut modifier le gain de résonance, et aussi la condition de commutation, de telle façon à ne pas avoir de dépassement. On arrive aux résultats de la figure IV.38.



**Figure IV.37 :** Résultats de simulation de la commande finale

Néanmoins on notera que la commande qui peut donner ces résultats est très grande (40v), et que la saturation de celle-ci va déstabiliser le système.

# *Chapitre V*

---

*Implémentation et résultats expérimentaux*

## Introduction

Ce chapitre est consacré à l'implémentation de lois de commandes synthétisées dans le chapitre précédant. Néanmoins, des modifications sur le système nous y imposées. En effet, nous avons pris l'initiative de changer la carte de commande DSP TMS320C32 qui est initialement utilisée par une dSPACE (DS1104). Nous allons voir les motifs de cette modification dans la suite de ce chapitre.

### 1. Présentation du système



*Figure V.1 : maquette du nouveau système pendule inversé.*

Notre système est initialement commandé à travers une carte DSP TMS320C32, qui est considérée parmi les premières DSP de Texas Instruments (TI) qui sont dédiées à la commande des systèmes, mais malheureusement elle n'est pas en mesure de satisfaire les exigences de nos objectifs de commandes puisqu'elle ne permet pas l'implémentation de lois de commandes modernes qu'on veut justement appliquer sur notre système. En plus, elle est dotée d'un logiciel (CEMtool) qui n'est pas très sophistiqué puisque il ne contient pas des bibliothèques pour les commandes modernes comme le Flou ou les réseaux de neurones...etc. Pour remédier à ce même problème, on était obligé de changer cette carte par une de types dSPACE (DS1104) qui reprend parfaitement à nos attentes. En plus, elle est compatible avec le logiciel Matlab/ Simulink qui est très connu par les ingénieurs.



Le but principal de notre travail étant de changer la plate forme de commande de la maquette, une identification des pins de cette dernière devient nécessaire avant un nouveau câblage avec la PS600. On a résumé les pins utilisée dans le tableau V.1.

La maquette qu'on va présenter se trouve au niveau du Laboratoire de Commande Des Processus (LCP) de l'école nationale polytechnique d'Alger. On peut schématiser cette maquette de pendule inversé par la figure V.2.

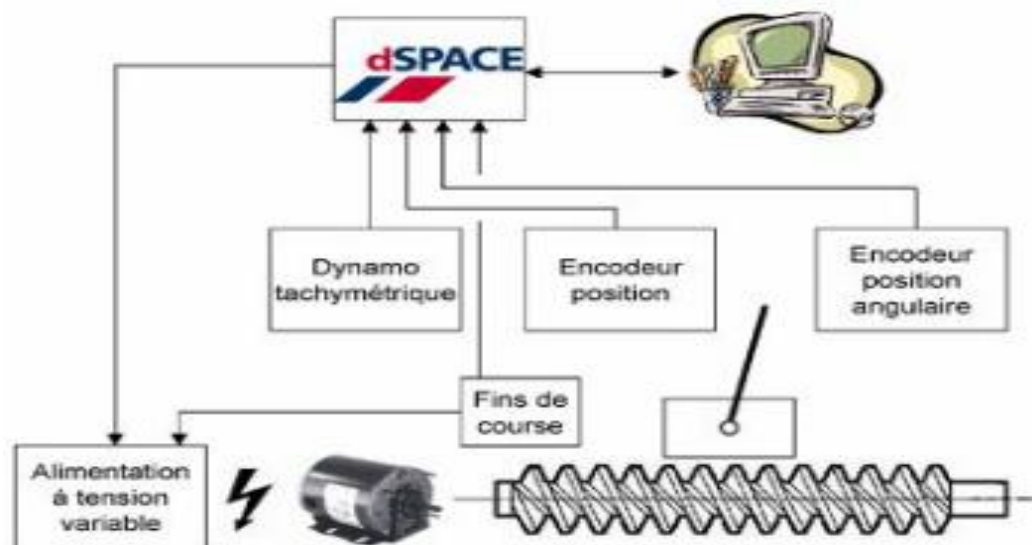


Figure V.2 : schéma de la maquette pendule inversé

### 1.1. Constitution du pendule inversé

Les principales constitutions de pendule inversé dont on dispose au laboratoire sont illustrées sur la figure V.3.

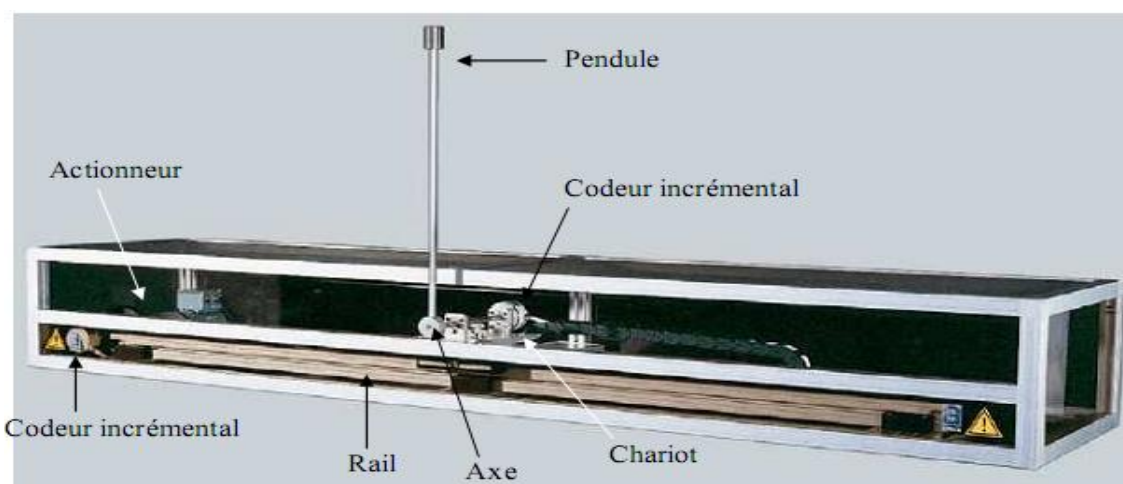


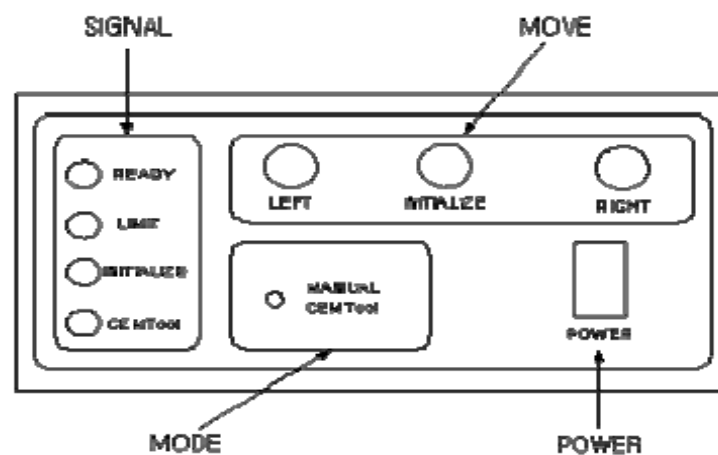
Figure V.3 : Constitution du pendule inversé

La maquette est constituée d'un chariot de masse  $M$  se déplaçant horizontalement sur un longeron de longueur **900 mm** supportant un pendule libre en rotation de longueur  $l$  et de masse  $m$ . Pour réaliser l'asservissement, le système a besoin d'une chaîne d'instrumentation et une chaîne d'actionneurs :

### a. La chaîne d'actionneurs

- **Boitier de control et d'alimentation**

Il sert à alimenter le système (pendule inversé) et lui fournir de l'énergie nécessaire à son bon fonctionnement, il possède un hacheur pour la commande du moteur en MLI, ce boitier joue un rôle très important puisque il protège l'instrumentation de système contre les fortes intensités de courant qui peuvent le détruire et garde le chariot dans les limites de bon fonctionnement en actionnant les fins de course en cas de dépassement. Il offre aussi la possibilité de le commander et d'initialiser manuellement le système.



**Figure V.4** : boîtier de contrôle avec illustration de fonctions

La fonction des différents boutons de ce boîtier est résumée dans le tableau V.1.

**Tableau V.1** : Boutons de la boîte de contrôle et de supervision du pendule

Composants		Fonction
LED	READY	S'allume quand le système est opérationnel
	LIMIT	S'allume quand le chariot atteint la limite de sa course à droite ou à gauche
	INITIALIZE	S'allume pendant l'initialisation du système
	CEMTool	S'allume quand le système est contrôlé par CEMTool
MOVE	LEFT	Se translate vers la gauche avec MODE en MANUAL
	INITIALIZE	Initialisation du système
	RIGHT	Se translate vers la droite avec MODE en MANUAL
MODE	MANUAL/CEMTool	Choix du mode manuel ou CEMTool
CEMTool		Connecteur 25P (à connecter à l'interface RG-IO)
POWER		Alimente le procédé

- **Le moteur**

Le moteur est utilisé pour entrainer la vis sans fin qui supporte le chariot, il est de type courant continu avec circuit d'excitation en aimant permanent, ses caractéristiques sont résumés dans le tableau AI.2

## **b. La chaîne d'instrumentation**

Elle est constituée d'organes de traitement et de la chaîne de réaction qui est constituée elle-même de capteurs de vitesse, de position, d'angle, et des fins de course.

On trouve :

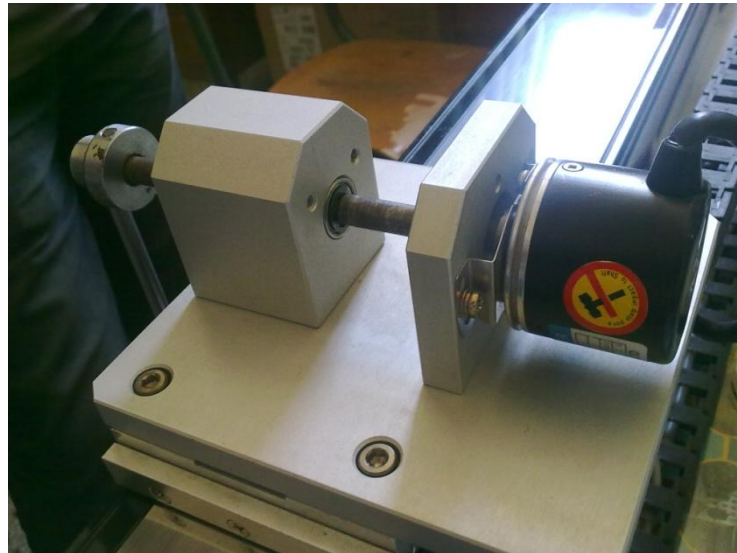
- **L'organe de traitement**

L'organe de traitement est un PC bureau muni du logiciel Matlab 6.1 et son environnement temps réel SIMULINK ainsi que le logiciel Control Desk, il est interfacé avec le système physique par le biais d'une carte d'acquisition dSPACE 1104. Cette carte possède des entrées/ sorties analogiques 0-10V ainsi que des entrées sorties numériques (voir ANNEXEII).

- **Les encodeurs incrémentaux**

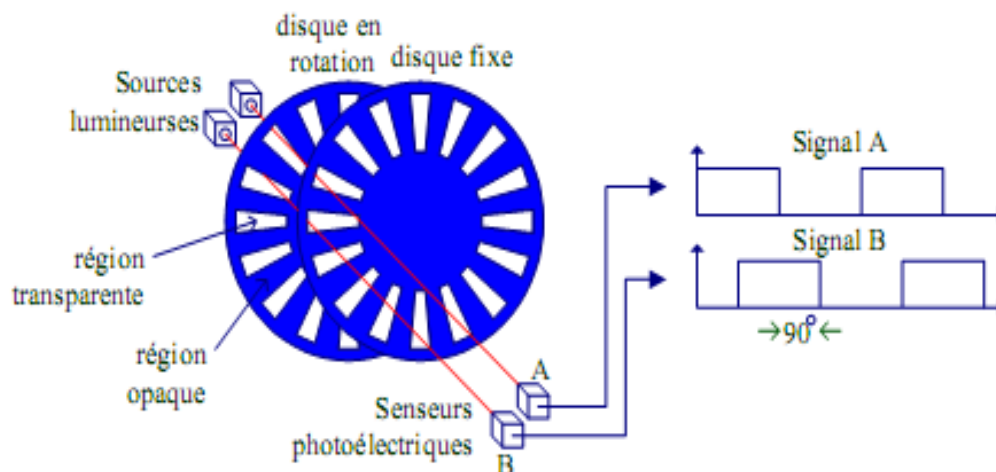
Le capteur de position angulaire du pendule est un encodeur rotatif à impulsions (comme une petite roue codée en binaire à deux bits dans ce cas). Ce type de capteur est généralement très précis et présente une très bonne immunité au bruit (figure AI.4).

La position de chariot est mesurée indirectement par le biais d'une mesure de rotation de l'arbre moteur. Le capteur effectuant cette mesure est tout à fait similaire à celui utilisé pour la mesure de la position angulaire de pendule.



**Figure V.6 :** Encodage incrémental

En effet, l'encodeur fournit deux signaux A et B évoluant en fonction de l'angle  $\theta(t)$  comme indiqué sur la figure suivante :



**Figure V.7:** Principe de fonctionnement d'encodeur incrémental.

Les signaux A et B forment donc un compteur 2 bits. Ceci permet à la carte dSPACE, en réactualisant la mesure de l'angle  $\theta(t)$  lors de chaque franc montant ou descendant de A ou de B, de quadrupler la résolution de l'encodeur.

- **Les interrupteurs de fin de course**

Quatre interrupteurs de fin de course sont câblés au dispositif, deux à chaque extrémité de la vis sans fin. Les deux interrupteurs extérieurs servent à garantir l'intégrité du dispositif. Ils sont reliés à l'alimentation et, lorsque le chariot les actionne, coupent instantanément toute tension appliquée au moteur, parce que la longueur de vis sans fin est limitée et il ne faut pas la dépasser. Les interrupteurs intérieurs permettent quant à eux de réinitialiser le dispositif simplement et en toute sécurité.

Les caractéristiques mécaniques et électriques du système sont détaillées sur le tableau :

**Tableau V.2:** Spécifications techniques du système.

Caractéristique		Valeur
Spécifications Mécaniques	W×L×H (mm)	1300×200×250 (mm)
	Longueur du pendule	400 mm
	Poids du pendule	0.2 Kg
	Poids du chariot	1 Kg
	Déplacement	900 mm
	Pas de la vis à bille	12.7 mm
Spécifications électriques	Tension de sortie du moteur	24 V, 60 W
	Vitesse maximale du moteur	3800 tr/mn
	Impulsions de l'encodeur	4000 impulsions (×4)
	Tension d'entrée du moteur	0 ~ 5V

## 1.2.La carte dSPACE1104



**Figure V.8:** carte dSPACE 1104.

La carte DSP utilisée est la DSPACE DS1104, le processeur principal est un MPC8240, avec un cœur PowerPC 603e, avec horloge interne à 250 MHz. Il a une capacité mémoire de 8 Mo en Flash et de 32 Mo en SDRAM.

Il dispose de 8 convertisseurs analogiques numériques (4 en 16 bits, 4 en 12 bits), de 8 convertisseurs numériques analogiques (CNA) de 16 bits pouvant délivrer une tension +/- 10V, d'une liaison série, de 2 codeurs incrémentaux, de 20 entrées-sorties numériques, d'un DSP esclave (TMS320F240) et de 3 timers (32 bits) peuvent fonctionner de manière indépendante.

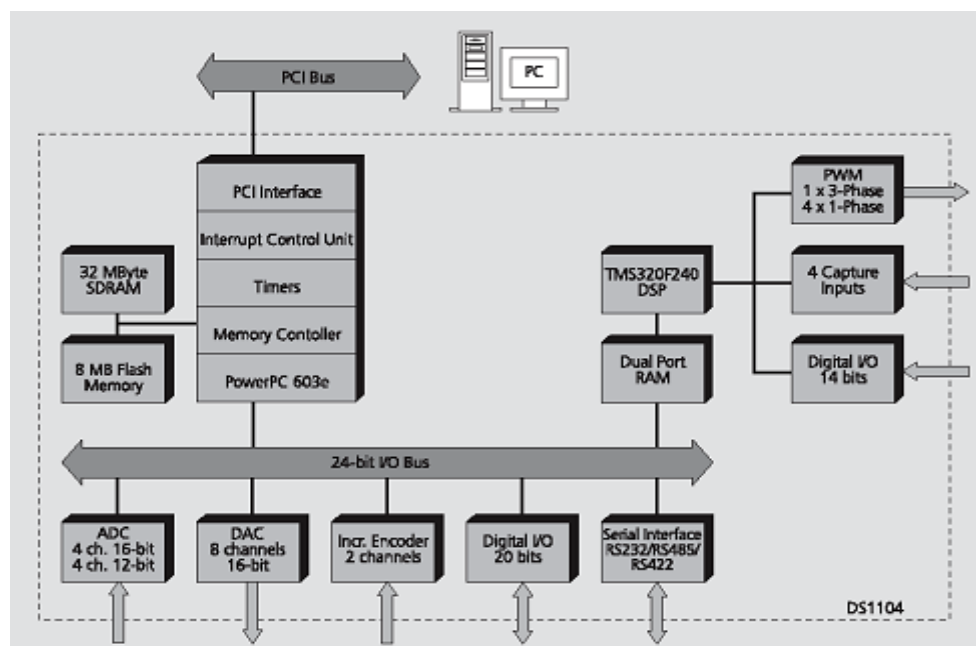
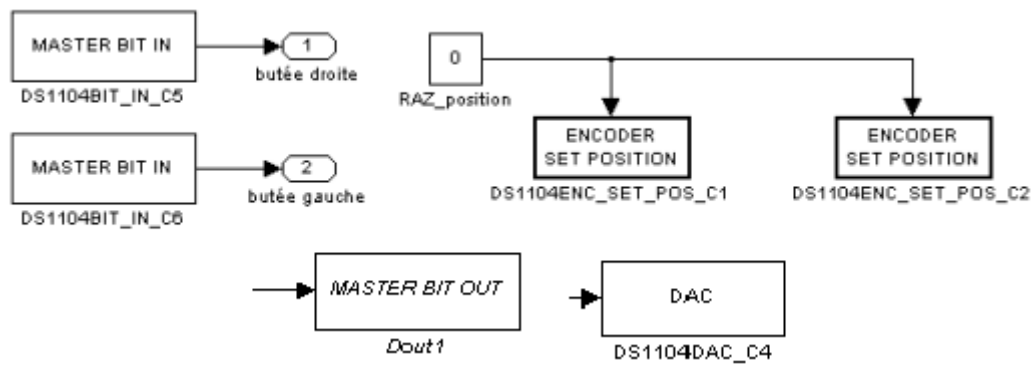


Figure V.9: Architecture du DS 1104

Pour programmer le DSP, il faut réaliser son schéma sur la boîte à outils Simulink de Matlab. Une bibliothèque spéciale « **DS1104 Real Time Library** » regroupe les différentes fonctions utiles pour gérer le DSP. A l'aide de la commande Build dans Simulation Parameters, on charge le code directement dans le DSP.

Il est préférable de se placer dans son répertoire de travail pour construire le fichier sur Simulink car la construction génère de nombreux fichiers (voir annexe). Le logiciel Control Desk offre une interface qui permet de visualiser et de modifier en temps réel différentes variables du schéma Simulink.

Un exemple de blocs utilisés est présenté figure ci-dessous. En faisant un double click sur chaque bloc, on peut choisir le champ dans lequel on souhaite écrire ou lire les données.



**Figure V.10:** Exemple de blocs utilisés pour la conception du schéma Simulink

Une fois le schéma Simulink terminé et sauvegardé, l'étape suivante consiste à générer le code associé au schéma Simulink et à le transférer dans le DSP. Si la compilation a réussi, le chargement du programme et son exécution dans le DSP se font automatiquement. Pour compiler un schéma Simulink :

- Aller dans le menu simulation, puis simulation paramètres.
- Dans l'onglet **Solver**, fixer Type à **fixed-step** puis **fixed step** à Ts (période d'échantillonnage choisie). Mettre stop time à **inf**.
- Dans l'onglet **real-time workshop**, choisir la cible puis **Build**.

La compilation génère de nombreux fichiers dont deux sont particulièrement importants :

- Schéma.sdf listant tous les paramètres du schéma bloc, utilisés sous Control desk pour faire le lien entre le firmware et l'IHM.
- Schéma.ppc qui est le firmware chargé dans la carte DSP.

**NB :** Lorsque le programme est compilé, il utilise par défaut les valeurs des blocs du schéma Simulink car celui-ci est automatiquement exécuté. Attention à bien choisir les valeurs initiales de ces blocs.

Le logiciel Control Desk permet la réalisation d'une interface graphique permettant de visualiser et de modifier en temps réel les différentes variables du schéma Simulink. Les extensions utilisées sous Control Desk sont :

**Schema.cdx :** contient le projet ou expérimente

**Schema.csv :** fichier d'acquisition de mesures au format ASCII

**Schema.idf :** fichier résultant de l'acquisition des mesures en mode Stream to Disk

**Schema.lay :** contient le panneau graphique développé (c'est la layout)

**Schema.mat** : fichier d'acquisition de mesures pouvant être chargées sur Matlab

**Schema.ppc** : contient l'application (donc l'exécutable) pouvant être chargée dans le DSP.

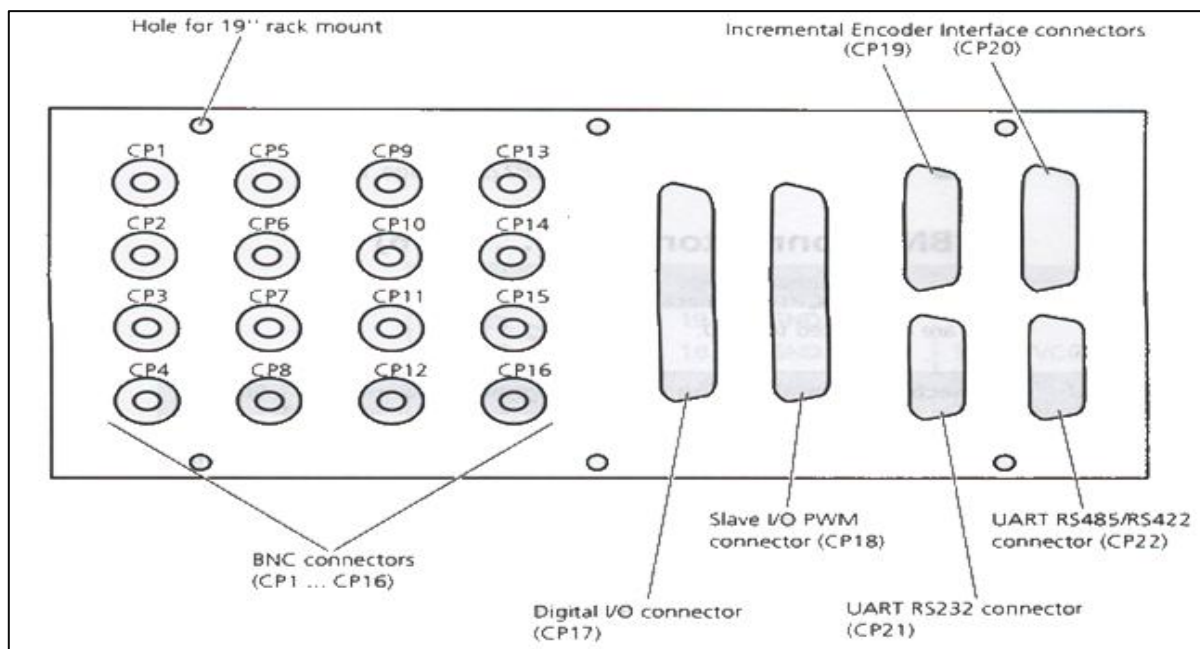
**Schema.sdf** : contient l'ensemble des paramètres du fichier Simulink

### 1.3. Le boîtier de connexion PS 600

Il assure la liaison entre le système et la carte dSPACE, il comporte plusieurs ports d'entrées/sorties analogiques et numériques:

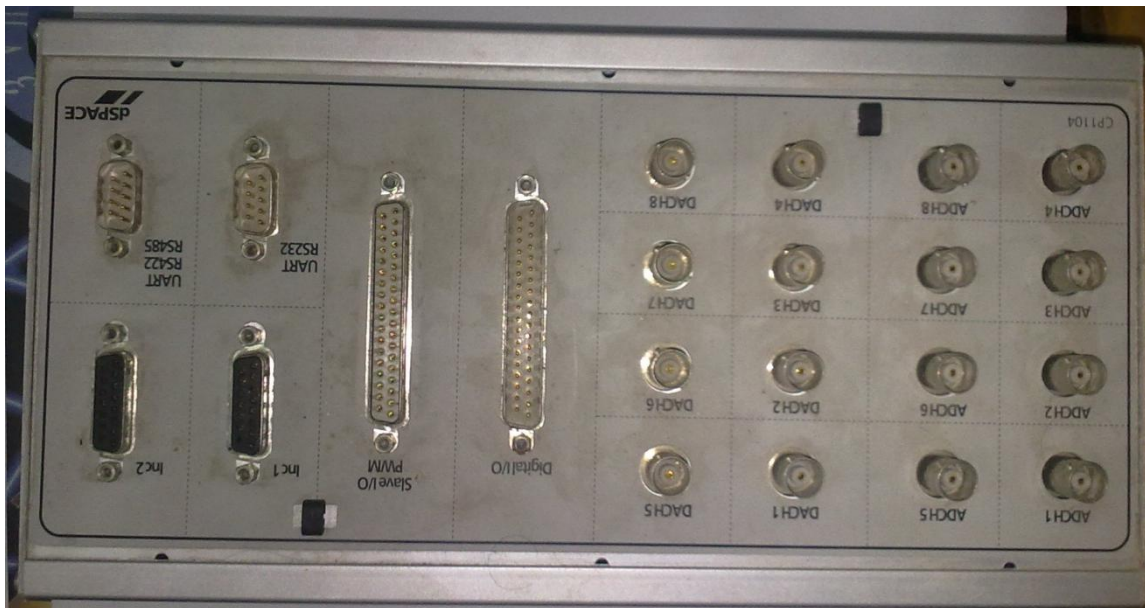
- PWM I/O pour commander les moteurs
- 2 ports pour des encodeurs incrémentaux (CP19-20)
- 16 ports type BNC dont 8 d'entrée et 8 de sortie (CP1-16)
- Un port d'entrées/sorties numériques (CP17)
- un port RS232 (CP21)
- Un port RS485/422 (CP22)

Dans notre cas, on n'aura pas besoin de la sortie PWM, vu que le boîtier de commande de la figure V.4 se charge de générer la PWM à partir du signal à son entrée, et ceci avec son hacheur intégré.



**Figure V.11.a** : Schéma de l'interface PS600





**Figure V.11.b :** boîtier PS600.

## 1.4. Le logiciel ControlDesk [26]

Une fois le programme de commande est fait sous Simulink, le logiciel contrôle Desk le charge dans la dSPACE.

Control Desk est une interface qui permet de visualiser en temps réel différentes variables du fichier développé sous Simulink et de modifier également des paramètres définissant le mode de fonctionnement des blocs constituant le schéma Simulink. La visualisation de variables ou de signaux et la modification de paramètres sont possibles par l'intermédiaire d'instruments graphiques que l'on sélectionne. Un exemple de construction d'un écran graphique est présenté ci-après (ou IHM : Interface Homme Machine).

Les différentes fenêtres rencontrées sur Control Desk sont :

- 1 – La fenêtre de navigation (Navigator)
- 2 – L'aire de travail contenant la layout
- 3 – Le choix d'instrument (instrument selector).
- 4 – La fenêtre d'outils (tool window).

Si l'une de ces fenêtres n'est plus visible à l'écran, on peut les ouvrir dans **View→Controlbar**.

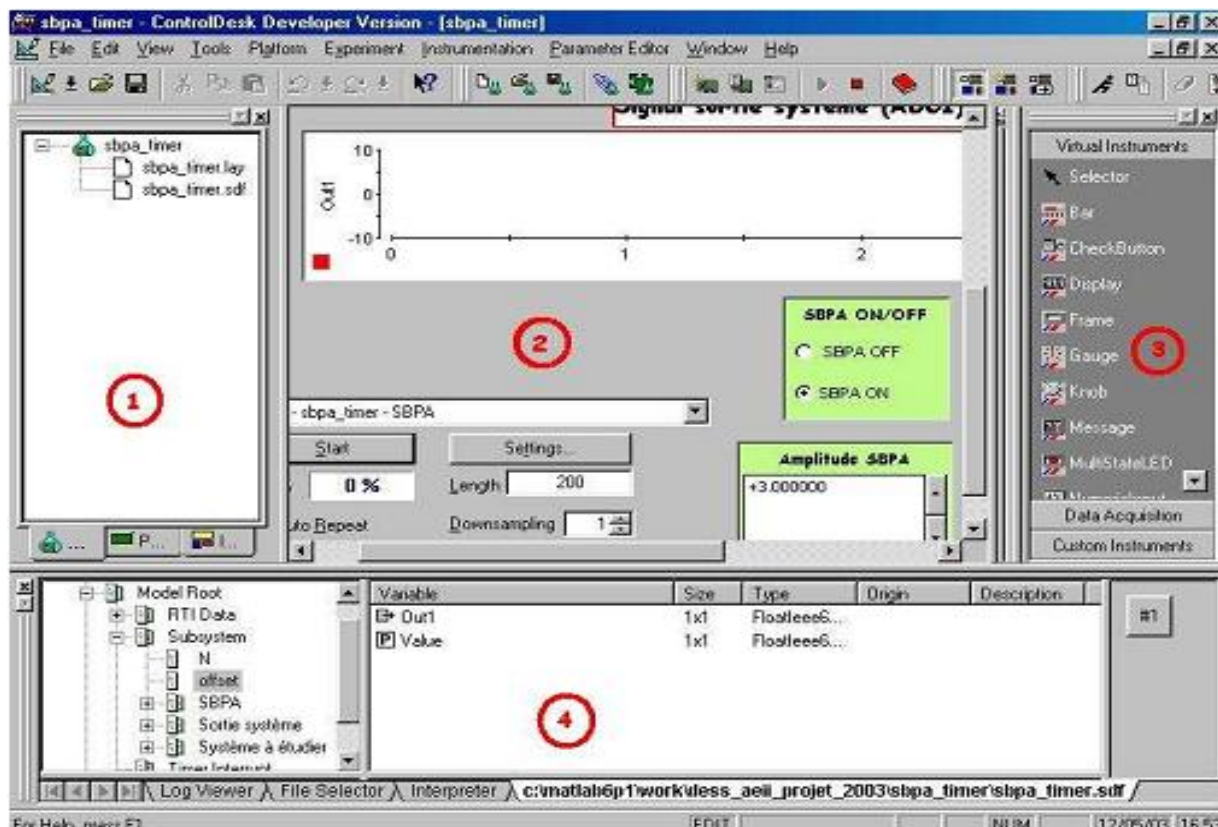


Figure V.12 : Exemple d'IHM développée sous Control Desk.

### i. Démarrer un nouveau projet sur Control Desk


Sous Control Desk, un projet s'appelle un "experiment".

Dans l'onglet 'File', cliquer sur 'New experiment' puis saisir le nom et le répertoire désiré pour le projet. Dans l'exemple, le fichier s'appelle essai.cdx.

**Remarque :** un "layout" est une interface graphique à laquelle on peut ajouter divers instruments dans le but de visualiser ou de modifier en temps réel les différentes variables du projet.

Dans 'File', cliquer sur 'New' → 'Layout'

↳ Se référer à la partie créer un layout.

Pour lancer la simulation, cliquer sur l'icône.  On est alors en mode simulation (ou animation).

### ii. Ouvrir un projet existant

Pour ouvrir un projet existant, aller dans 'File', cliquer sur 'Open experiment'. Charger l'application dans la carte DSP.

Par exemple, l'application "essai" peut être chargée dans le DSP en ouvrant l'expérience `essai.cdx` puis en déplaçant `essai.ppc` jusqu'à la fenêtre Navigator comme le montre la figure A.III.2. Dans ce cas, il n'est pas nécessaire de compiler le schéma Simulink.

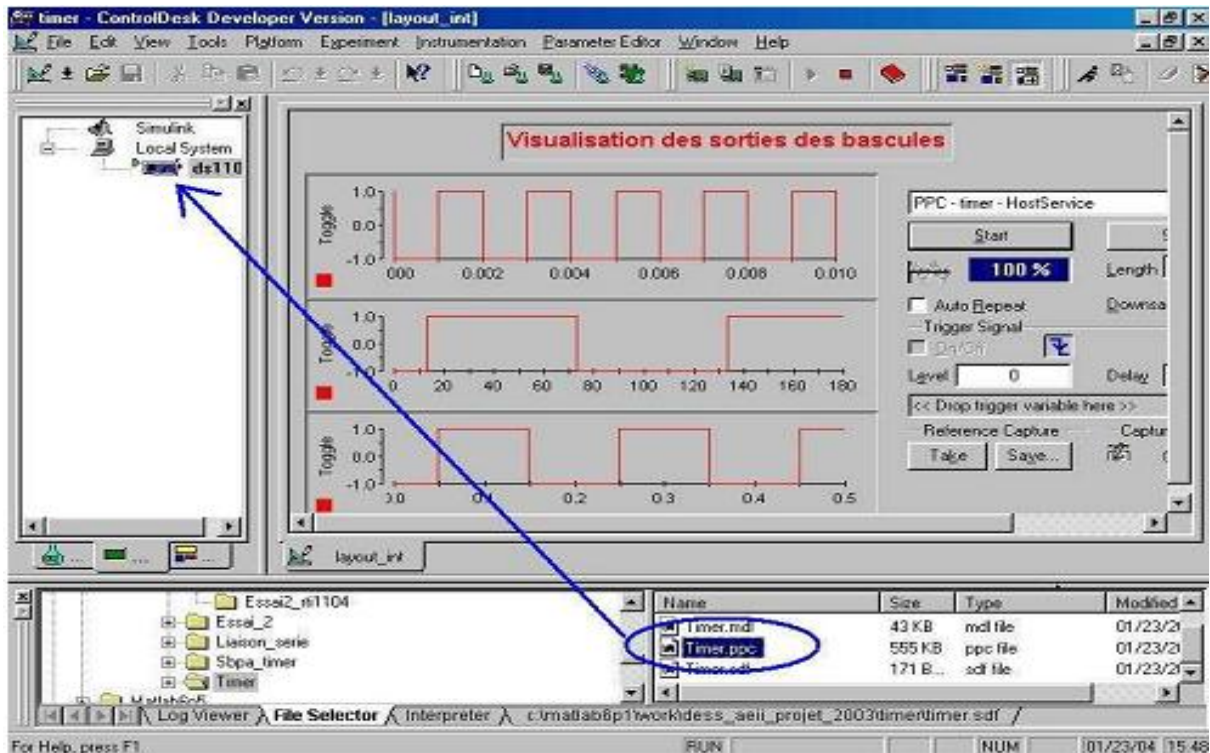


Figure V.13: Exemple de chargement d'une application dans la carte dSPACE

### iii. Travailler dans l'espace de travail


#### ➤ Créer un layout

L'espace de travail est l'endroit dans lequel on va créer un layout. Il existe deux catégories d'instruments :

- Virtual instruments : ces instruments permettent la visualisation et/ou la modification de variables.



- Data acquisition : ces instruments permettent l'acquisition de variables (à l'écran ou dans un fichier).

Le logiciel offre également la possibilité grâce à l'onglet 'Custom instruments' d'éditer de nouveaux instruments et de les sauvegarder dans une librairie.

Au départ, le layout est une fenêtre grise. Pour développer son propre layout, il faut se mettre en mode Edit (icône .

**Remarque :** on peut réaliser plusieurs layouts dans un même projet. On passe de l'un à l'autre en cliquant sur l'onglet 'Layout' de la fenêtre de gauche et en choisissant le layout à ouvrir.

Il faut ensuite ajouter tous les instruments de visualisation et de modification de variables. Enfin, il faut affecter à chaque bloc sa variable et modifier les propriétés de l'instrument si besoin, en double cliquant dessus.

Dans la fenêtre d'outils, on récupère les différentes variables ou paramètres du schéma Simulink. L'initialisation d'un objet graphique est réalisée en effectuant un clic gauche sur la variable et en la faisant glisser jusqu'à l'instrument désiré. Le sigle  Out1 signifie que l'on va pouvoir visualiser la variable. Le sigle  Value signifie que l'on va accéder à la valeur de la variable et pouvoir la modifier.

L'instrument 'Capture Settings' dans 'data acquisition' est un outil essentiel, il permet de choisir la durée de l'acquisition, de l'arrêter momentanément, de sauvegarder une acquisition de données.

Le bloc 'Plotter Array' dans 'data acquisition' permet la visualisation sous forme graphique de la variable au cours du temps.

Pour modifier les propriétés d'un instrument, double-cliquer dessus puis sélectionner 'Properties'. On peut ainsi modifier le nom d'une variable, régler l'échelle des axes, etc.

#### ➤ Sauvegarder des données sur fichier

Control Desk offre la possibilité d'effectuer une acquisition de données. L'outil 'Capture Settings' dans 'data acquisition' dispose d'un onglet 'capture variable'. On peut effectuer des acquisitions simples, avec sauvegarde automatique, acquisitions continues, etc.

On peut effectuer des sauvegardes au format **.csv** ou **.mat**. Le premier permet une sauvegarde dans un fichier ASCII éditable directement depuis ControlDesk. Le deuxième présente l'avantage d'être directement récupérable sur Matlab sous forme d'une structure. Dans l'outil '**Capture Settings**', cliquer sur 'Settings' puis sur l'onglet 'Acquisition' afin de sélectionner le mode de capture désiré :

**Simple** : fait l'acquisition du signal pendant la durée établie (length) dans 'Settings' (de 0 à 100% du temps capturé).

**Autosave** : fait une capture de données dans le fichier spécifié.

**Autoname** : effectue une capture dans le fichier spécifié et incrémente le nom du fichier automatiquement. Exemple : defaut001.mat, defaut002.mat ...

**Continuous** : permet de capturer les données de manière continue jusqu'à ce que l'on stoppe l'acquisition. La dernière période visualisée est alors sauvegardée. Vérifier que la vitesse du processeur temps réel soit compatible avec la vitesse d'acquisition du PC.

**Stream to disk** : enregistre les données en continu et les écrits sur le disque dur. Cette acquisition crée des fichiers portant l'extension '**.idf**'. A la fin de l'acquisition, il faut convertir le fichier '**.idf**' en un fichier '**.mat**'. Pour cela, aller dans Tools, cliquer sur Convert IDF File. Lorsque l'acquisition est arrêtée, on peut visualiser les données depuis le début.

La conversion peut se faire de deux manières différentes :

- Conversion vers un fichier .mat contenant toutes les données.
- Conversion vers n+1 fichiers .mat contenant chacun les données temporelles d'une variable plus un fichier temporel.

Sous Matlab, on charge le fichier **.mat** (load <nom\_du\_fichier>.mat). Ensuite, on trace les différentes variables en récupérant le temps par la commande <nomdufichier>.X.Data et la (les) variable(s) observée(s) par <nomdufichier>.Y(n).Data. Pour faciliter ce travail un fichier prog.m a été réalisé ci-après.

```
load essai.mat;
t = essai.X.Data;
var1 = essai.Y(3).Data;
var2 = essai.Y(2).Data;
var3 = essai.Y(1).Data;

%tracé graphique
figure;
plot(t, var1, t, var2, t, var3);
grid;
title('Mettre un titre');
XLABEL('Temps en secondes');
YLABEL('Variables visualisée');
```

**Programme Matlab de récupération des données de Control Desk (prog.m)**

### ➤ La fenêtre d'outils

La fenêtre d'outils comprend 'Log viewer', 'Interpreter', 'File selector' et le fichier de variable portant l'extension '.sdf'.

**Log viewer**: affiche les messages générés par Control Desk ou par la carte temps réel.

**File selector**: permet de sélectionner une application et de pouvoir la télécharger dans la carte DSP. Ceci évite d'avoir à effectuer la commande Build à partir du schéma Simulink lorsqu'une version compilée existe déjà.

**Fichier de variable**: (extension .sdf) permet d'accéder aux variables de l'application.

Le fichier de variables est chargé automatiquement dans le projet lors de l'exécution de la commande Build de Simulink.

Si toutefois le fichier de variables n'est pas chargé, aller dans File → Open variable file et charger le fichier portant l'extension .sdf qui est dans le répertoire de travail.

A partir du fichier de variables, on peut sélectionner une variable ou la valeur d'une variable afin de la faire glisser dans un instrument du Layout.

## 2. Mise au point du nouveau système de commande

Dans cette section, on présente les grandes étapes suivies afin d'établir le changement du système de commande précédent à base de TMS320C32 et CEMtool, par la dSAPCE1104 et Matlab.

### 2.1. Identification des PINS

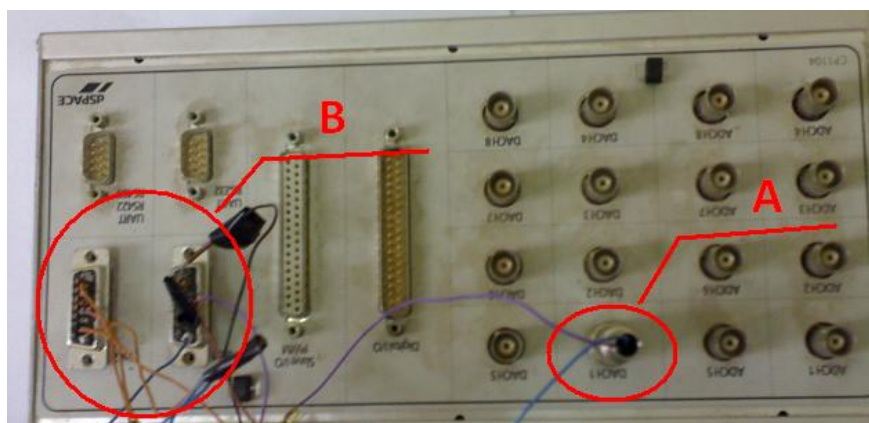
Le câblage des pins est réalisé avec des câbles simples de connexion. La PS600 dispose de deux entrées encodeur et de 8 entrées analogiques, et aussi de 8 sorties analogiques. Pour notre application, on utilisera les deux entrées encodeurs, pour les coordonnées du système, et une sortie analogique, pour sa commande. Et ceci comme le montre la figure V.3 avec (A) branchement de la commande et (B) les mesures (encodeurs).

**Tableau V.3 : L'identification des pins [20]**

Pendulum No.	Signal Types of Pendulum System	Signal Types of Rapid Design Controller	RG-DSPIO01Pin No.	COULEUR DES CABLES
1	Motor Encoder A	Encoder A (ch0)	9	JAUNE
2	Motor Encoder $\bar{A}$	Encoder $\bar{A}$ (ch0)	27	JAUNE/NOIR
3	Motor Encoder B	Encoder B (ch0)	8	JAUNE/ROUGE
4	Motor Encoder $\bar{B}$	Encoder $\bar{B}$ (ch0)	26	JAUNE/BLEU
5	Pendulum Encoder A	Encoder A (ch1)	11	JAUNE/VERT
6	Pendulum Encoder $\bar{A}$	Encoder $\bar{A}$ (ch1)	29	JAUNE/GRIS
7	Pendulum Encoder B	Encoder B (ch1)	10	BLANC
8	Pendulum Encoder $\bar{B}$	Encoder $\bar{B}$ (ch1)	28	BLANC/NOIR
9	Encoder 전원 (+5V)	+5V	36	BLANC/ROUGE
10	GND	GND	18	BLANC/BLEU
11	Motor Operating Signal (+)	Analogue Output 0 Signal	1	BLANC/VERT
12	Motor Operating Signal (GND)	Analogue Output 0 Grounding	3	BLANC/GRIS

## 2.2. branchements des PINs

C'est l'étape la plus importante, après avoir fait l'identification des signaux portés par chacun des câbles de la figure V.2 on les dessoude, et on les ressoude avec la nouvelle interface de commande c'est-à-dire l'interface DS600 de la DS1104, comme le montre la figure V.4.



**Figure V.15 : branchement de la commande (A) et les mesures (B)**

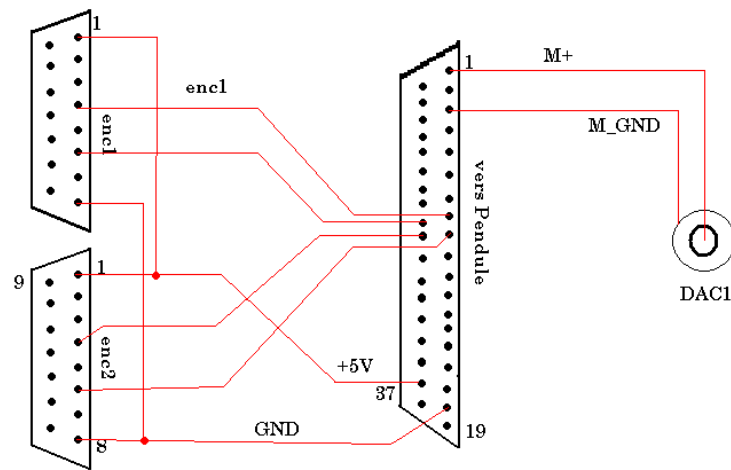


Figure V.16 : Schéma de câblage entre la dSPACE et le système

En suite, nous avons configurés ses entrées/ sorties sur Simulink qui permet l'échange des données entre le système et la dSPACE.

Les programmes de Simulink qui assurent ces échanges sont donnés par la figure V.5.

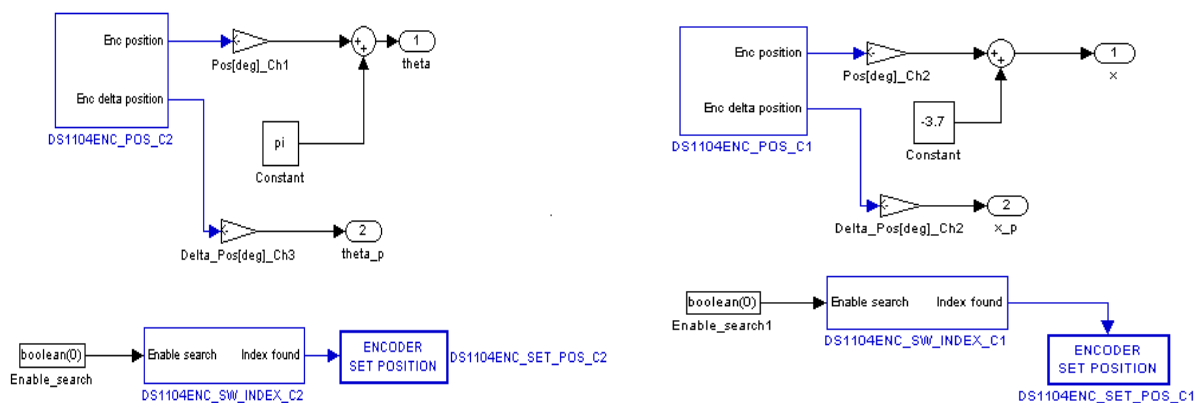


Figure V.17 : Blocks Simulink d'acquisition de données

Comme les capteurs dont on dispose sur la maquette sont des encodeurs incrémentaux ; il est nécessaire pour obtenir les valeurs de l'angle et de la position de rajouter des blocs de conversion qui permettent de traduire le nombre d'impulsion calculé en valeur exploitable ayant pour unité le radian et le centimètre respectivement.

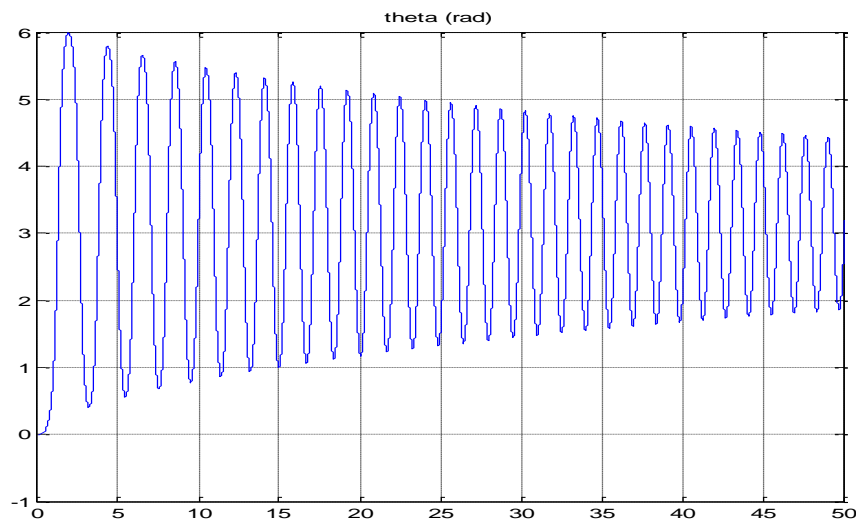
### 3. Résultats expérimentaux

Après avoir fait la mise en marche de système et faire les essais préliminaire pour initialiser les différents capteurs, on passe à l'implémentation de commandes mais avant, il faut d'abord vérifier la validation du modèle.



### 3.1. Expérimentation du système réel en BO et validation du modèle

Le but de cette expérience est de faire une comparaison entre les mesures prises sur le système réel avec Control Desk et les résultats de simulation obtenus dans le paragraphe précédent, et cela afin de valider le modèle mathématique élaboré dans le chapitre II et sur lequel on a synthétisé les commandes. En utilisant la carte dSPACE comme périphérique d'acquisition de données, on tire le fichier \*.mat de control desk avec l'outil de capture, de sa on obtient les graphiques comparatives de la figure II.6.



**Figure V.18:** Simulation du système réel en BO

On constate tout d'abord que la pulsation naturelle du système réel est la même que celle du système modélisé (4 périodes en 6 secondes). On note néanmoins une différence de taille entre le modèle et le système réel, au niveau de l'atténuation, en effet le système réel aussi plus longtemps que le simulé. Ceci est dû aux hypothèses simplificatrices lors de l'élaboration du modèle, on suggère alors d'améliorer le modèle avant d'entamer toute autre expérimentation.

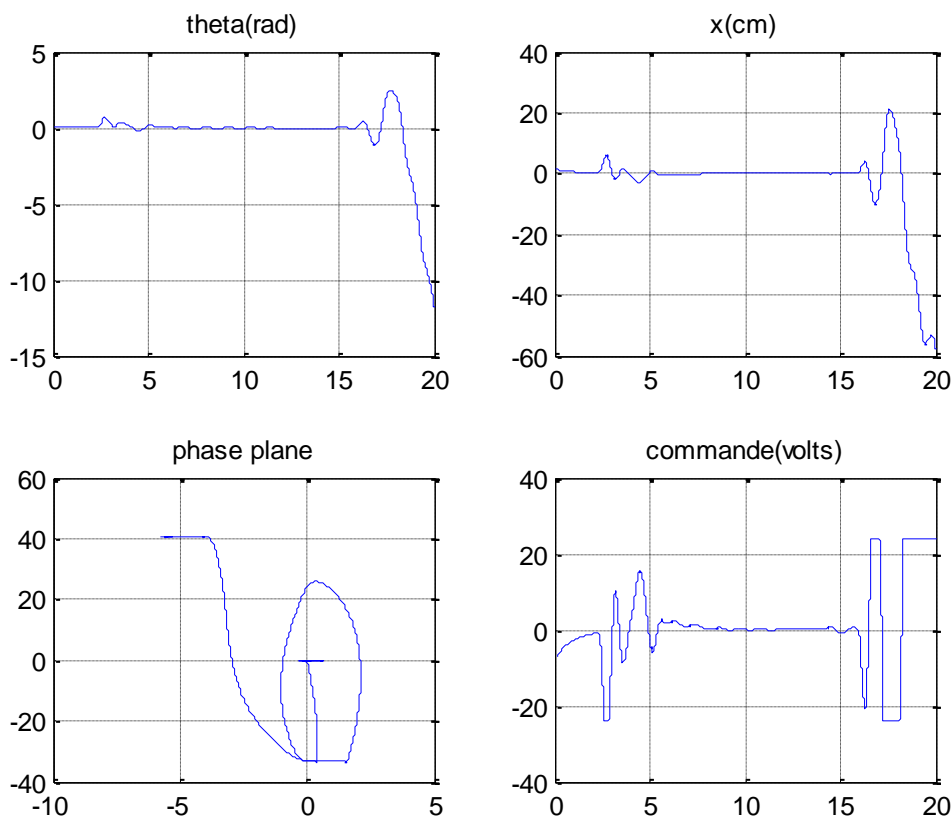
### 2.2. Commande par modes de glissement et léviation par résonance

Dans cette commande, les gains de la surface de glissement sont obtenus par placement de pôles  $P = [-2, -2, -2, -2]$ , en utilisant le modèle linéaire obtenu avec la commande Matlab linmod. Les premières simulations donnent des résultats similaires à ceux prévus par la simulation comme on le voit dans la figure IV.36. En effet le système de commande par modes de glissement aura du mal à stabiliser une grande condition initiale sur la vitesse du

pendule comme on le voit sur la figure V.8. Pour remédier à ce problème, un réglage approprié des gains du régulateur de léviation par résonance s'impose. Après tâtonnement, on fini par trouver les graphique de la figure V.9.

### 2.2.1. La commande de léviation par action résonante

Dans cette partie, on verra les résultats expérimentaux relatifs à la commande de léviation par action résonante, ou comme vu dans le chapitre 4, le moteur sera attaquer par une tension proportionnel à la vitesse du pendule. Ceci dit l'ordre de proportionnalité va changer. On aura la constante légèrement plus petite que celle utilisé pour la réalisation de la simulation, comme le montre les figure V.19 suivante.

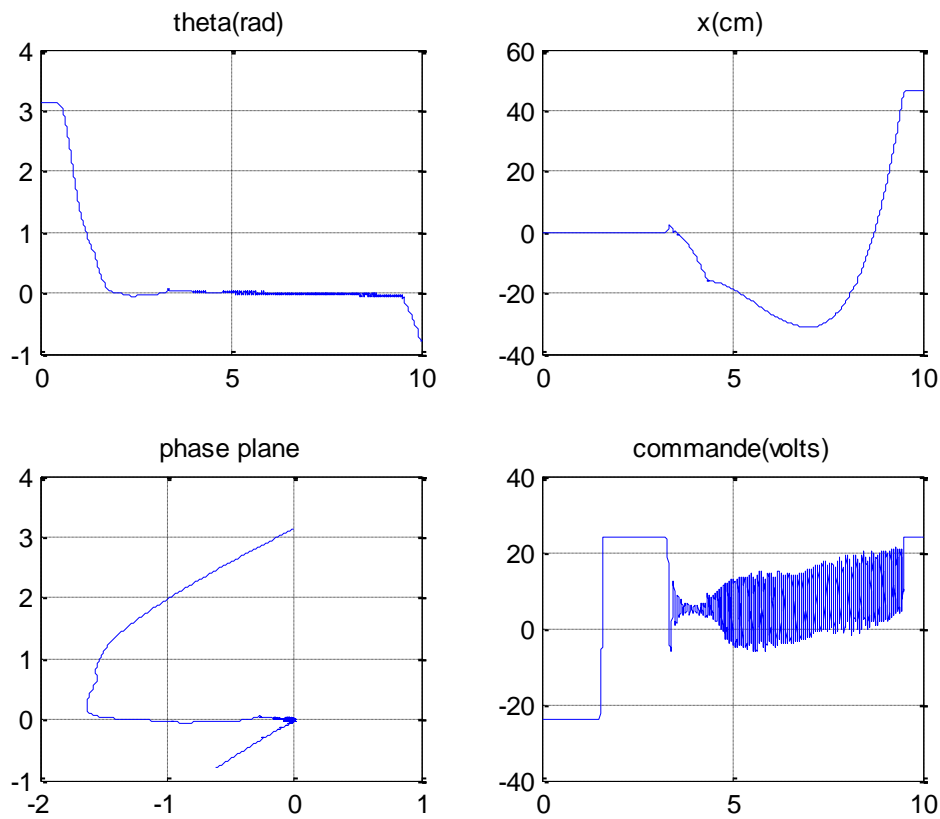


**Figure V.19 :** Expérimentation du balancement par action résonante

### 2.2.2. Commande de stabilisation par modes de glissements

Il est bien connu que l'implémentation d'une loi de commande entraîne une dégradation de cette dernière, en modifiant ces caractéristiques et ses performances. Le premier test de notre commande donne le résultat prévu par la simulation (figure IV.35). Malgré l'ajout d'une action intégral sur le déplacement, le résultats reste imprévisible et le chariot a

tendance à s'éloigner de l'origine, ceci peut être réglée avec une action intégrale sur le déplacement. Le résultat est présenté dans la figure V.20.



**Figure V.20 :** Résultats d'implémentation du mode de glissement simple

On peut dire que entre simulation et réalisation, il existe plusieurs étape d'adaptation des signaux en entrée et en sortie du système, en effet un oscilloscope nous permettra de repérer la forme des signaux présent sur la maquette, et peut révéler des perturbations ou des bruits sur un signal, ce qui peut être réglée avec un filtrage ou tous autre technique de traitement de signal.

---

# Conclusion générale

---

Le travail réalisé dans le cadre de ce mémoire, concerne la mise en œuvre d'un nouveau système de commande pour la maquette pendule inversé qui a été mise à notre disposition au sein de Laboratoire des Commandes des Processus (LCP) de l'ENP, ensuite la synthèse et l'implémentation de lois de commande.

Le pendule inversé présente plusieurs avantages ; et comme nous l'avons vus dans le premier chapitre ; son principe est applicable dans pas mal de domaines comme la robotique ; l'aérospatiale ; la médecine, ...et dans la recherche puisqu'il est considéré comme un très bon banc d'essai et de test de commandes vus ces caractéristique très intéressantes à savoir qu'il est non linéaire et sous actionné.

Pour atteindre nos objectifs fixés nous nous sommes tout d'abord menés à connaître les différentes constitutions de notre système et cela en identifiant ses chaînes d'instrumentation et d'action. Par la suite ; et dans le but de synthétiser les commandes ; une modélisation s'est imposée. En effet, en utilisant la méthode de Lagrange Euler ; et en optant pour des hypothèses simplificatrices ; nous avons aboutis à un modèle non linéaire qui est très approché du système réel comme nous avons pu le constater après sa validation.

Dans le chapitre de lois de commande, nous avons commencé par des commandes simples (standards). En effet, on a synthétisé deux commandes de basculement (swing up) à savoir la commande par retour d'état non linéaire et la commande floue puis nous avons relevé le pour et le contre de chacune. Nous avons constaté que le flou donne de bons résultats puisque le temps de basculement est plus petit que par la première commande. En suite, nous avons synthétisé différentes commandes de stabilisation comme le flou, le glissant et le retour d'état simple et optimal. Pour la commande par logique floue, au début on a dimensionné un régulateur à deux entrées (on n'a pas pris en compte le déplacement  $x$  et sa dérivé), et on a remarqué que le pendule se stabilise, mais  $x$  non et sort de la limite tolérée de longeron (non implémentable). Et pour remédier à cet inconvénient, nous avons dimensionné un régulateur flou découplé qui assure la stabilisation de tous les états de système et a donné de résultats intéressants de point de vue robustesse aux variations

paramétrique mais cela en réajustant les gains de normalisation ce qui n'est pas facile en générale. L'avantage majeur de la commande par logique floue est qu'elle est structurellement indépendante du modèle ce qui le classe parmi les commandes robustes. L'autre commande synthétisée pour la stabilisation est la commande par les modes glissants. En effet, on a choisie une surface comportant tous les états du système et ses paramètres on les a calculé au premier lieu par placement de pôles simple puis par placement de pôles optimal ce qui nous a donné un résultat meilleur puisque la commande est moins énergétique. Néanmoins, malgré que cette commande a donné de très bons résultats puisqu'elle est robuste vis-à-vis des variations paramétriques et assure la stabilisation du système en temps fini et rejette les perturbations, elle présente un inconvénient majeur qui est le phénomène de réticence (chattering) ce qui peut nuire le moteur lors de l'implémentation. Nous avons aussi synthétisés la commande par retour d'état simple et optimale et qui ont donné de bons résultats mais qui sont uniquement valables dans une petite plage autour du point d'équilibre et elles ne sont pas robuste puisque une variation paramétrique sur la longueur du pendule fait déstabiliser le système ce qui rend les chances de les implémenter est vraiment réduites.

Pour remédier certaines limites que comportent certaines commandes que nous avons fait jusqu'à maintenant, et pour profiter des avantages des unes et des autres ; nous avons combiné entre les commandes et synthétisé ce qu'on appelle les commandes hybrides. En effet, pour régler le problème de la commande de glissement, on a pensé à l'optimiser par un régulateur flou en substituant la fonction signe par la fonction saturation dans la pente est calculée par régulateur flou. Les résultats obtenus sont très satisfaisants mais après plusieurs essais pour choisir des bons paramètres pour le régulateur flou. La seconde commande hybride que nous avons synthétisée est la commande neuro-floue de type **ANFIS** qui donne des résultats optimaux.

La dernière parties de ce mémoire est consacrée à la mise au point du nouveau système de commande ou nous avons illustré les principales étapes suivies pour l'aboutissement à ce but :

-La première étape consiste à l'identification du système et ses différentes entrées/ sorties qui assure la communication entre le système et la carte dSPACE et cela en sélectionnant les

PIN correspondants, c'est une étape très importante et les erreurs ne sont pas tolérées et justement elle nous a pris beaucoup de temps.

-La deuxième partie consiste à faire les branchements nécessaires et en soudant les différents fils.

-Dans l'étape suivante, nous avons fait l'adaptation des signaux et les convertir en signaux exploitables par la dSPACE puisque les encodeurs (capteurs) retournent des valeurs sous formes d'impulsions et donc nous sommes obligés de les convertir en degrés ou en radians pour les visualiser. En suite, nous avons vérifié tout cela à travers des petits programmes de tests.

-La dernière étape consiste à se familiariser avec le logiciel Control Desk qui a le rôle de chargement des programmes générés par le Real Time Workshop de Simulink, il nous permet aussi de visualiser n'importe quel signale parce qu'il a une interface Homme/Machine bien interactives.

Une fois nous avons terminé tout ça, nous avons entamé la dernière étape de notre travail qui consiste à l'implémentation de lois de commandes synthétisées auparavant. Dans cette phase ; et faute de moyens ; on a contenté seulement par l'implémentation de lois de commandes simples et on n'a pas pu aller jusqu'aux bout de notre objectif qui est l'implémentation des commandes hybrides car la version de notre logiciel de simulation est très ancienne et ne permet pas l'implémentation de régulateur comme le flou et sur qui se base nous commandes hybrides.

Enfin, nous pouvons affirmer que ce projet nous a poussé à faire appel à plusieurs connaissances, puisque il nous a permet de touché et a la théorie et a la pratique en même temps, et ce justement qui nous a permis de voir la différence flagrante entre les deux car contrairement à la simulation, l'implémentation des commandes est très difficile à faire et il demande du temps et surtout la patience.

Finalement, étant donné que le système fonctionne bien maintenant, nous proposons comme perspective de changer la version du logiciel de simulation par une qui permet l'implémentation de commandes hybrides modernes comme le flou et le neuronal en plus changer les PC.

## Bibliographie

- [1] S. RIACHY. *Contribution à l'estimation et à la commande de systèmes mécaniques sous-actionnés*, thèse doctorat de l'Ecole Centrale de Lille 2008.
- [2] H.KHAMMAR, S.TLIBA, *commandes hybrides d'un système non linéaire sous actionné, Simulation et expérimentation*. PFE ENP, 2009.
- [4] A. DJOUDI, T. BENABDELLAH, *synthèse et application de lois de commandes non linéaire sur un pendule inversé*, PFE ENP, 2008.
- [5] J. J. Slotine, W. Li, *Applied Non Linear Control*, Prentice-hall international eds, 1991.
- [6] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, London, 1995.
- [7] BÜHLER : « Réglage par mode de glissement ». Presse polytechniques romandes. Lausanne 1986.
- [8] BÜHLER: « Réglage par logique floue ». Presse polytechniques romandes. Lausanne 1994.
- [9] Mustafa Resa Becan "*Fuzzy boundary layer Solution to Nonlinear Hydraulic Position Control Problem*", World Academy of science, Engineering and Technology 2005.
- [10] International Journal of Control, Automation, and Systems: Swing-up Control for an Inverted Pendulum with Restricted Cart Rail Length, (2009).
- [11] Astrom, K.J., Furuta, K.: Swinging up a pendulum by energy control. Automatica (2000).
- [12] Nonlinear Dynamics (2005) Lyapunov-Based Controller for the Inverted Pendulum Cart System ,CARLOS AGUILAR IBANEZ, GUTIERREZ FRIAS.
- [13] R. Lozano, I. Fantoni, and D. J. Block, "Stabilization of the inverted pendulum around its orbit," System & Control Letters
- [14] D. GEEST "*techniques neuronale floue pour les problèmes de régulation*", thèse Doctorat INSA Toulouse, 1995.
- [15] "DS1104 PPC Controller Board", Germany: dSPACE, July 2008.
- [16] F. BOUDJEMA "*Recherche Opérationnelle et commande optimale*", cours 5<sup>ième</sup> année Automatique, ENP.
- [17] M. TADJINE. "*Automatique Avancée*". Cours 5<sup>ième</sup> année Automatique, ENP.

- [18] A. BOUBAKIR, F. BOUDJEMA, C. BOUBZAKIR, N. IKHLEF, Loi de Commande par Mode de Glissement avec Une Surface de Glissement Non Linéaire Appliquée au3 Système Hydraulique à Réservoirs Couplés, 4th International Conference on Computer Integrated Manufacturing CIP'2007.
- [19] Hung T. Nguyen. Nadipuram R. Prasad Carol L. Walker. Elbert A. Walker. "A first course in fuzzy and neural control". Chapman & Hall/CRC.2003.
- [20] Manual of the Inverted Pendulum Designer, RealGain Co., Ltd.
- [21] V.I. Utkin, "Variable structure systems with sliding modes", IEEE Transactions on Automatic Control, Vol, 22, No 2, pp 212-222, 1977.
- [22] ALOUANI Fouad,"Commande par Logique Floue Appliquée aux pendules inversés et au Simulateur de Vol d'hélicoptère : Simulation et Expérimentation ", 2006.
- [23] Ji-Hyuk Yang, Su-Yong Shim, Jung-Hun Seo, and Young-Sam Lee « Swing-up Control for an Inverted Pendulum with Restricted Cart Rail Length International Journal of Control, Automation, and Systems (2009)
- [24] Alan Bradshaw and Jindi Shao « *Swing-up control of inverted pendulum systems* » *Robotica volume 14. pp 397-405. 1996.*
- [25] A. Abrous, *Modélisation, commande et Réalisation d'un Convertisseur Matriciel.(Programmation des S-Fonction et Manuel d'utilisation de la carte DS 1104 de dSPACE) . Thèse de Magister de l'ENP .2005.*
- [26] guide d'implémentation de la dSPACE. 2003