

UNIVERSITÉ DE MONTRÉAL

SYSTÈMES NON LINÉAIRES : UNE APPROCHE PAR LA LOGIQUE
FLOUE APPLIQUÉE À LA PLATE-FORME DE GOUGH-STEWART

VINCENT RÉMILLARD
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
SEPTEMBRE 2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-19326-6

Our file Notre référence

ISBN: 978-0-494-19326-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

SYSTÈMES NON LINÉAIRES : UNE APPROCHE PAR LA LOGIQUE
FLOUE APPLIQUÉE À LA PLATE-FORME DE GOUGH-STEWART

présenté par: RÉMILLARD Vincent

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de:

M. BARON Luc, Ph.D., président

M. BOUKAS El-Kébir, Ph.D., membre et directeur de recherche

M. BALAZINSKI Marek, Ph.D., membre

à Karine

REMERCIEMENTS

Je remercie mon directeur de recherche, M. El-Kébir Boukas, d'avoir accepté de diriger mes travaux de recherche. Il a su garder ma motivation élevée, par de judicieux conseils et un engagement hors du commun. Malgré les nombreuses responsabilités qu'il lui incombe, il a su être toujours disponible pour la finalisation de mon projet.

Je remercie le président du jury M. Luc Baron, ainsi que M. Marek Balazinski, d'avoir pris le temps nécessaire à l'évaluation et au jugement de mon travail.

Je remercie M. Georges Duclos, directeur de l'École Secondaire Dalbé-Viau, d'avoir accepté de me donner une chance en enseignement, malgré mon inexpérience. Je remercie mes collègues de travail de l'école Dalbé-Viau pour leur patience et leur engagement professionnel.

Je remercie ma famille et mes amis, de leur soutien, leurs encouragements, leur amour.

RÉSUMÉ

Ce mémoire présente une approche d'application de la logique floue pour la commande d'un système fortement non linéaire. La plupart des systèmes dynamiques, qu'ils soient mécanique, électrique, chimique, aérospatiale, robotique, etc., possèdent de nombreuses non linéarités qu'on ne peut négliger lors des applications de commande.

Lors de la commande de tels systèmes, il y a des performances souvent cruciales à atteindre. Ainsi, pour que le système soit précis, pour qu'il soit rapide, pour qu'il soit robuste, etc., il faut le munir d'un contrôleur efficace.

La principale contribution de ce travail est de considérer le système mécanique non linéaire de la plate-forme de Gough-Stewart (PGS), un système reconnu pour être des plus difficiles à commander, et de lui appliquer une méthode basée sur la modélisation floue de Takagi et Sugeno (TS). Le choix de cette approche a été déterminée pour prendre avantage de la modélisation floue TS, tout en contournant le maximum de difficultés.

En effet, par le modèle TS, on réalise l'approximation d'un système non linéaire par un ensemble de systèmes linéaires. Et l'analyse de plusieurs de ces systèmes, par un formalisme basé sur la théorie de la logique floue s'avère beaucoup plus simple que de considérer le système non linéaire dans son ensemble.

À ce point, la formulation du problème sous forme d'inégalités matricielles linéaires convexes (LMI) permet de déterminer si un système est stable ou non, et si on peut le stabiliser par un contrôleur approprié. La résolution de ces LMI permet de déterminer directement les gains du contrôleur par retour d'état qui stabilisent le

système.

Par ailleurs, dans la foulée de l'utilisation de la puissance numérique de la résolution de LMI, plusieurs résultats sont explorés, afin de déterminer les avantages et les inconvénients de chacun et d'en arriver à un contrôleur performant. À cette fin, notons que des outils de placements de pôles sont également appliqués pour des considérations d'amélioration de performances.

La méthode présente des avantages et des inconvénients par rapport à celles déjà effectuées. La comparaison entre diverses méthodes selon des critères qualitatifs et quantitatifs est alors conduite.

On tire ainsi la conclusion que l'approche privilégiée permet la synthèse d'un contrôleur alliant des performances de rapidité et de précision, qui permet de pallier à d'importantes difficultés de modélisation. Et qui plus est, la méthode peut être étendue facilement pour l'élaboration d'un contrôleur plus sophistiqué et pour d'autres applications tout aussi complexes.

ABSTRACT

The following work addresses a the fuzzy logic application to control highly nonlinear systems. Most of these dynamical systems, in some fields like mechanic, electric, aeronautic, etc., show a lot of nonlinearities that cannot be neglected in control applications.

To control such of class of systems, there is often some crucial performances to reach. To obtain fastness, preciseness, robustness, etc., we need an effective controller to perform the prescribed task.

The main contribution of this work appears with the application of the Takagi and Sugeno's fuzzy model (TS) on the Gough-Stewart platform (PGS), which is very well known to be a very difficult dynamical system to control. The chosen method takes advantage of the TS fuzzy modeling and avoid most of difficulties related to nonlinear system analyses.

Indeed, the TS fuzzy model proposes global approximation of the nonlinear model with multiple local linear systems. And the analyses of many linear systems with fuzzy logic formalism appear to be much more simpler than dealing with the complete nonlinear system.

The problem formulation, cast into convex linear matrix inequalities (LMI), allows to characterise properties of the system like stability. If the system is not stable, we can also used the LMI-based formulation to design the suitable controller. Also, the resolution of these LMI bring directly the state feedback controller gains that stabilise the system.

In addition, in the tread of the use of the LMI numerical power resolution, several results are explored, to determine advantages and also disadvantages of each results to design the best controller possible. For this purpose, pole placement tools for performance considerations are also used to obtain the desired improvement of performances.

The method has advantages and disadvantages compared to those already carried out. The comparison between the various methods according to qualitative and quantitative criteria's is then led.

To draw a conclusion, the synthesis approach to find the controller is successful. The controller design makes the system response fast and precise and allows to mitigate important difficulties. Also, the method can be wide easily for making of a controller more sophisticated or for other applications quite as complex.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES FIGURES	xiv
LISTE DES NOTATIONS ET DES SYMBOLES	xvi
LISTE DES TABLEAUX	xviii
LISTE DES ANNEXES	xix
CHAPITRE 1 INTRODUCTION ET REVUE DE LA LITTÉRATURE .	1
1.1 Problématique et choix d'une solution	1
1.1.1 Quelques méthodes	2
1.1.2 Choix de la logique floue	5
1.2 Application à la PGS	6
1.2.1 Revue de la littérature sur la PGS	8
1.2.1.1 Historique	8
1.2.1.2 Cinématique	10
1.2.1.3 Dynamique	12
1.2.1.4 Commande	14
1.2.1.5 Autres difficultés	16

1.3	Structure du mémoire	17
CHAPITRE 2 APPROCHE PAR LA LOGIQUE FLOUE		18
2.1	Avant-propos	18
2.1.1	Bref historique	18
2.1.2	Applications	19
2.2	Approche générale de résolution	20
2.2.1	Quantification floue	21
2.2.2	Établissement des règles	22
2.2.3	Combinaison des règles	23
2.3	Modèle flou TS	24
2.4	Contrôleur PDC	26
2.5	Stabilité	27
2.6	Stabilité en boucle fermée	29
2.7	Augmentation de la rapidité	31
2.8	Conditions relaxées	32
2.8.1	Conditions relaxées 1	33
2.8.2	Conditions relaxées 2	34
2.9	Système affine	36
2.10	Intégrateur	38
2.11	Placement de pôles	40
2.11.1	Région plan gauche	42
2.11.2	Région circulaire	43
2.11.3	Région conique	44
CHAPITRE 3 MODÉLISATION DE LA PLATE-FORME		46
3.1	Avant-propos	46
3.2	Structure de la PGS	47
3.3	Choix des coordonnées	50

3.3.1	Repères des corps	50
3.3.2	Systèmes de coordonnées	51
3.3.2.1	Tenseur de coordonnées dans l'espace de travail . .	52
3.3.2.2	Vecteur des coordonnées dans l'espace articulaire .	52
3.4	Autres considérations	53
3.4.1	Matrice de Rotation	53
3.4.2	Vitesse et accélération angulaire	54
3.4.3	Matrices jacobienes	55
3.5	Cinématique	56
3.5.1	Relation de position et d'orientation	57
3.5.2	Relation de vitesse	59
3.5.3	Relation d'accélération	60
3.6	Dynamique	62
3.6.1	Dynamique dans l'espace articulaire	63
3.7	Représentation interne	64
3.7.1	Modèle non linéaire	64
3.7.2	Modèle linéaire	65
3.7.2.1	Développement en série de Taylor	65
3.7.2.2	Linéarisation par MATLAB	66
3.7.2.3	Linéarisation multi-modèles	67
CHAPITRE 4	EXPÉRIMENTATIONS	68
4.1	Avant-Propos	68
4.2	Modèle de simulation	69
4.2.1	Structures	69
4.2.2	Principaux paramètres	74
4.3	Premier contact avec la PGS	76
4.3.1	Commande dans l'espace de travail	76

4.3.2	Linéarisation	76
4.3.3	Tests simples	80
4.3.3.1	Stabilité en boucle ouverte	80
4.3.3.2	Stabilité en boucle fermée	81
4.4	Approximation par un modèle TS	82
4.4.1	Construction d'un modèle à 5 règles	83
4.4.2	Résultats préliminaires	87
4.4.2.1	Système affine	88
4.4.2.2	Ajout de l'intégrateur	92
4.4.2.3	Placements de pôles	94
4.4.2.4	Autres trajectoires	95
4.5	Modèle à 15 règles	97
4.5.1	Résultats	101
4.5.1.1	Suivi de position	103
4.5.1.2	Suivi d'orientation	104
4.5.1.3	Mouvement général	105
4.6	Comparaison des résultats	106
CHAPITRE 5	CONCLUSION ET EXTENSIONS POSSIBLES	110
RÉFÉRENCES		114
ANNEXES		122

LISTE DES FIGURES

FIG. 1.1	PGS typique (Bonev 2006)	7
FIG. 1.2	Origines de la plate-forme Gough-Stewart	9
FIG. 1.3	Distinction entre manipulateurs sériel et parallèle	10
FIG. 2.1	Structure d'un système flou	20
FIG. 2.2	Représentation de termes et concepts flous	22
FIG. 2.3	Méthodes de défuzzification	24
FIG. 2.4	Exemples de régions- $\mathcal{D}$	42
FIG. 3.1	Plate-forme de Gough-Stewart	48
FIG. 3.2	Éléments de la plate-forme de Gough-Stewart	49
FIG. 3.3	Attribution des repères sur les corps de la PGS	51
FIG. 3.4	Angles d'Euler	54
FIG. 3.5	Vecteurs pour exprimer la cinématique du système	57
FIG. 4.1	Structure de la PGS (Smith et Wendlandt 2002)	70
FIG. 4.2	Structure d'une membrure (Smith et Wendlandt 2002)	70
FIG. 4.3	Paramétrisation des articulations	72
FIG. 4.4	Repères et coordonnées des corps	73
FIG. 4.5	Vecteur de positionnement des membrures	74
FIG. 4.6	Stabilité du point $L_i = 0$	81
FIG. 4.7	Sous-ensembles flous pour le modèle à 5 règles	84
FIG. 4.8	Points de linéarisation possibles pour le modèle à 5 règles	85
FIG. 4.9	Linéarisation aux 5 points schématiques	85
FIG. 4.10	Trajectoires dans le plan XY	88
FIG. 4.11	Suivi de trajectoire : Conditions affines	91
FIG. 4.12	Suivi de trajectoire : Ajout d'un intégrateur	93
FIG. 4.13	Suivi de position : Intégrateur et placement de pôles	94
FIG. 4.14	Trajectoire triangulaire	95

FIG. 4.15	Suivi de position : Intégrateur et placement de pôles	96
FIG. 4.16	Sous-ensembles flous pour le modèle à 15 règles	98
FIG. 4.17	Points de linéarisation possibles pour le modèle à 15 règles .	99
FIG. 4.18	Linéarisation aux 15 points schématiques	99
FIG. 4.19	Suivi de trajectoire simples : 5 vs 15 règles	102
FIG. 4.20	Résultats pour un suivi de position quelconque	103
FIG. 4.21	Résultats pour un suivi d'orientation quelconque	104
FIG. 4.22	Résultats pour un mouvement complet	105
FIG. 4.23	Comparaison avec d'autres contrôleurs	108

LISTE DES NOTATIONS ET DES SYMBOLES

PGS :	Plate-forme de Gough-Stewart
LMI :	Inégalité Matricielle Linéaire
TS :	Takagi-Sugeno
DDL :	Degrés De Liberté
PID :	Proportionnel-Intégral-Dérivée
SI :	unités dans le Système International
h :	Vecteur de position de la plate-forme
e :	Vecteur des angles d'Euler
q :	Vecteur de coordonnées dans l'espace de travail
L :	Vecteur de coordonnées dans l'espace articulaire
S :	Vecteurs d'élongation des membrures
$(b_i)_B$:	Position de l'articulation i de la bases dans le référentiel B
$(p_i)_P$:	Position de l'articulation i de la plate-forme dans le référentiel P
${}^B R_P$:	Matrice de rotation pour le passage du référentiel P à B
$c\theta$:	Cosinus de l'angle θ
$s\theta$:	Sinus de l'angle θ
ω :	Vitesse angulaire
α :	Accélération angulaire
J :	Matrice jacobienne
τ :	Vecteur des forces dans l'espace de travail
F :	Vecteurs des forces dans l'espace opérationnel
$M(q)$:	Matrice de masse et d'inertie du système
$N(q)$:	Matrice contenant les forces et les couples centrifuges et de Coriolis
$G(q)$:	Matrice contenant les forces et les couples dus à la gravité
$x(t)$:	Vecteur d'état à l'instant t

$y(t)$:	Vecteur de sortie à l'instant t
$u(t)$:	Vecteur d'entrée à l'instant t
$z(t)$:	Vecteur des prémisses de fuzzification à l'instant t
M_{ij} :	Sous-ensembles flous
$w(\cdot)$:	Fonction d'appartenance des prémisses aux sous-ensembles flous
A_i :	Matrice d'état
B_i :	Matrice d'entrée
C_i :	Matrice de sortie
K_i :	Matrice des gains du contrôleur
P :	Matrice de Lyapunov

LISTE DES TABLEAUX

TAB. 2.1	Avantages de l'approche par la logique floue	18
TAB. 4.1	Position des articulations de la PGS	75
TAB. 4.2	Paramètres massiques de la PGS	75
TAB. 4.3	Partie réelle des valeurs propres de la matrice A_1	80
TAB. 4.4	Élongations L_i aux points (1 à 5) de linéarisation	86
TAB. 4.5	Résidus de linéarisation	89
TAB. 4.6	Règles flous pour le modèle à 15 règles	100
TAB. 4.7	Élongations L_i aux points (6 à 15) de linéarisation	100
TAB. 4.8	Comparaison des résultats	107
TAB. II.1	Commande u_{ri} aux points d'équilibre ($\times 10^3$)	124
TAB. II.2	Matrices $(a_{21})_i$ et $(b_{21})_i$	125
TAB. II.3	Matrices $(a_{21})_i$ et $(b_{21})_i$ (...suite)	126
TAB. II.4	Gains K_i du contrôleur	127
TAB. II.5	Gains K_i du contrôleur (...suite)	128
TAB. II.6	Gains K_i du contrôleur (...suite)	129
TAB. II.7	Gains K_{ei} du contrôleur	130
TAB. II.8	Gains K_{ei} du contrôleur (...suite)	131
TAB. II.9	Gains K_{ei} du contrôleur (...suite)	132

LISTE DES ANNEXES

ANNEXE I	OUTILS MATHÉMATIQUES	122
I.1	LMI	122
I.2	Complément de Schur	123
ANNEXE II	VALEURS NUMÉRIQUES	124

CHAPITRE 1

INTRODUCTION ET REVUE DE LA LITTÉRATURE

Dans ce chapitre d'introduction, on fait ressortir la problématique de la commande des systèmes non linéaires. Certaines méthodes, déjà proposées pour commander ce type de systèmes, sont énoncées brièvement. On y indique sommairement des applications dans plusieurs champs d'expertises, avant de présenter une méthode qui s'appuie sur les fondements de la logique floue. Takagi et Sugeno (TS) (1985) proposent un modèle flou permettant de représenter des modèles non linéaires et de synthétiser des contrôleurs flous. Pour montrer l'efficacité des méthodes se basant sur le modèle TS, une application sur la plate-forme de Gough-Stewart (PGS) est envisagée. Pour mieux comprendre ce mécanisme, un survol de nombreux travaux déjà effectués sur la PGS est présenté. Finalement, on expose à la fin de ce chapitre les divisions du mémoire.

1.1 Problématique et choix d'une solution

Les équations et fonctions non linéaires sont d'un grand intérêt en ingénierie. La majorité des systèmes dynamiques, que ce soit dans des domaines d'électronique, de mécanique, de robotique, d'automatique, etc., présentent des comportements non linéaires dont il faut tenir compte lors de leur analyse. Les systèmes dynamiques non linéaires représentent donc une vaste classe. Bien que certains systèmes et équations non linéaires d'intérêt général aient fait l'objet d'études approfondies, ils n'en demeurent pas moins difficiles à étudier et il n'existe pas de théorie unifiée pour l'étude de ces systèmes. Nous limitons cependant l'analyse aux systèmes dif-

férentiables.

En particulier, les systèmes non linéaires ne sont pas sujets au principe de superposition. Ce qui implique que la somme de deux excitations ne correspond pas à la somme des deux réponses correspondantes. La linéarité des systèmes permet l'utilisation de plusieurs hypothèses et approximations simplificatrices qu'on ne peut utiliser pour les systèmes non linéaires. La modélisation est très difficile, parfois même impossible. Et leur comportement par rapport au temps, par exemple, est extrêmement difficile à prédire. Pour en connaître davantage, le lecteur intéressé peut consulter Slotine et Li (1992), Krstic et al. (1995), Khalil (2002) et les références citées dans ces ouvrages.

Un système non linéaire peut être représenté par la forme générale suivante :

$$\dot{x}(t) = f(x(t), u(t)), \quad (1.1)$$

où

x : variables d'état du système ;

$\dot{x}$: dérivée de x par rapport au temps t ;

u : variables d'entrée du système.

1.1.1 Quelques méthodes

Pour l'analyse des systèmes non linéaires, il existe des outils et il est très difficile d'avoir une vision d'ensemble de toutes ces différentes techniques ou méthodes. La théorie de Lyapunov est reconnue comme étant incontournable pour l'analyse

de stabilité de multiples systèmes dynamiques. Certains ouvrages lui sont presque entièrement dédiés (Khalil 2002). Voici quelques méthodes utilisées pour l'étude et la commande de systèmes non linéaires :

- **La linéarisation exacte par retour d'état** est une approche de commande des systèmes non linéaires. Le but de cette méthode est de linéariser le système à l'aide d'une transformation d'état. Le système résultant est équivalent, mais linéarisé. Ainsi, il est possible d'utiliser la rétroaction d'état pour contrôler le système à l'aide d'un contrôleur linéaire. Elle a déjà été efficace pour la commande de plusieurs systèmes tels qu'une transmission hydrostatique (Haichang et al. 2006) et une pile chimique (Na 2005). Cependant, cette méthode n'est pas des plus intuitives. Elle fait appel, pour sa généralisation, à des notions mathématiques complexes, telles que le crochet de Lie et les ensembles involutifs. Elle est souhaitable pour des systèmes simples uniquement, car elle nécessite une connaissance complète de la dynamique du système.

- **La méthode des perturbations singulières** est une opération mathématique utilisable lorsqu'un système présente plusieurs dynamiques au comportement très différent. Par exemple, un manipulateur robotique peut avoir une dynamique très lente en comparaison avec ces actionneurs électriques. D'ailleurs, cette méthode a été utilisée pour un système de robots flexibles coopératifs (Xiao-Tao et Yuan-Chun 2005). Aussi, un procédé industriel tel que le contrôle en température d'un four à convection se prête bien à la méthode (Vasquez et Krstic 2005). La méthode se veut un changement de variables permettant de traiter les sous-systèmes indépendamment.

- **Le backstepping** pour la commande des systèmes non linéaires a été développé autour des années 1990 par Kokotovic (Krstic et al. 1995). C'est une méthode en cascade où le dernier sous-système nous intéresse. En partant de la dernière com-

posante dans la chaîne, il est possible de considérer la variable précédente comme une variable de commande, et ainsi de suite. Le contrôleur est ainsi synthétisé par « retour-arrière » en effectuant des changements de variables appropriées. Cette méthode est bien adaptée pour la commande en trajectoire d'un planeur (Hann-Shing et al. 2005) et peut se prêter à des considérations de commande optimale (Jianyun et Yuanzhang 2005).

- **Les modes de glissement**, par leur action, permettent la synthèse d'un contrôleur à structure variable permettant d'atteindre une valeur de référence par une commutation entre les différents états de sa structure. On appelle mode de glissement cette oscillation très rapide autour de la valeur de référence. Les applications de cette méthode sont nombreuses. Notons que la commande d'une machine à commande numérique est réalisable (Shen et Lin 2005).

Il implique généralement les deux étapes suivantes :

1. Sélection des surfaces de glissement telle que la trajectoire désirée évolue le long de cette surface.
2. Trouver les gains du contrôleur faisant en sorte que le système demeure sur ces surfaces.

Des variantes de la méthode originale sont également utilisées. On parle alors de mode de glissement dynamique et de mode de glissement d'ordre plus élevé, qui présente de meilleures performances que la méthode originale (Koshkouei et al. 2005).

- **La commande adaptative** permet un ajournement des paramètres estimés d'un système. En effet, si les paramètres du système sont estimés, il est possible d'ajourner le contrôleur pour qu'il tienne compte de ces estimations. Par de nombreux calculs, tout au long de la tâche de commande, le contrôleur s'adapte au système réel qu'il contrôle. L'algorithme de Slotine-Li (Slotine et Li 1992) est une

méthode adaptative efficace, mais qui nécessite de nombreux calculs et une bonne connaissance des relations dynamiques du système. Mais elle peut facilement se combiner à d'autres méthodes. De concert avec le backstepping, il est possible de commander un moteur synchrone (Ouassaid et al. 2004) ; combiné avec les modes de glissement, on commande la trajectoire d'un missile (Hwang et al. 2005).

Les quelques techniques énumérées jusqu'ici ne sont pas, bien évidemment, les seules permettant la commande des systèmes dynamiques.

1.1.2 Choix de la logique floue

Pour simplifier l'analyse des systèmes non linéaires, il est très commun de procéder à une linéarisation du modèle, lorsque c'est possible. La littérature sur les systèmes linéaires est abondante et les méthodes sont nombreuses.

La linéarisation en un point de référence permet de concevoir un contrôleur faisant en sorte que le système soit stable pour les conditions d'opération désirées. Plus particulièrement, on peut désirer commander un système par un retour d'état, en linéarisant le système. Dépendamment des méthodes de linéarisation utilisées, il se peut que la continuité de la relation (1.1), ainsi que la continuité de ses dérivées partielles, soient requises. De plus, pour assurer la stabilité du contrôleur, il faut que ce dernier soit en mesure de ramener les pôles instables du système dans le demi-plan complexe de gauche. Et finalement, cette méthode ne fonctionne que localement, autour du point d'équilibre désiré.

Pour analyser les systèmes non linéaires complexes dans le but d'en faire la commande, serait-il possible d'utiliser les outils d'analyses de systèmes linéaires, pour

des conditions d'opérations plus générales qu'en un seul point ?

La réponse à cette question est fournie par Takagi et Sugeno (1985), qui proposent une modélisation basée sur la théorie de la logique floue. En effet, ils présentent un modèle composé d'un mélange de modèles linéaires. En fait, ce modèle flou est décrit par un ensemble de règles SI-ALORS qui représentent localement les relations linéaires entre les entrées et les sorties d'un système non linéaire. Il est démontré (Tanaka et Wang 2001) que le modèle flou TS permet l'approximation de plusieurs des systèmes non linéaires représentés par la relation. S'il sont différentiables, communément appelés « systèmes lisses », alors le modèle TS pour représenter à n'importe quel degré de précision les systèmes de la forme de l'équation (1.1).

Cette approche est donc envisagée car il est possible d'utiliser la même formulation que le modèle flou TS pour représenter un contrôleur par retour d'état. Sur les bases de la théorie de Lyapunov, le problème d'analyse et de synthèse des gains du contrôleur peut se formuler en inégalité matricielle linéaire convexe (LMI), qu'on résout par les outils de calcul numérique appropriés.

1.2 Application à la PGS

Pour montrer la validité des approches utilisant le modèle flou TS pour la synthèse de contrôleurs, une application sur un système fortement non linéaire est effectuée. Le système privilégié est le manipulateur parallèle, soit la plate-forme de Gough-Stewart.

Un manipulateur parallèle du type PGS est un mécanisme à 6 degrés de liberté (DDL) composé d'une plate-forme mobile jointe à une base fixe par 6 membrures extensibles. Une plate-forme typique est illustrée à la figure 1.1.

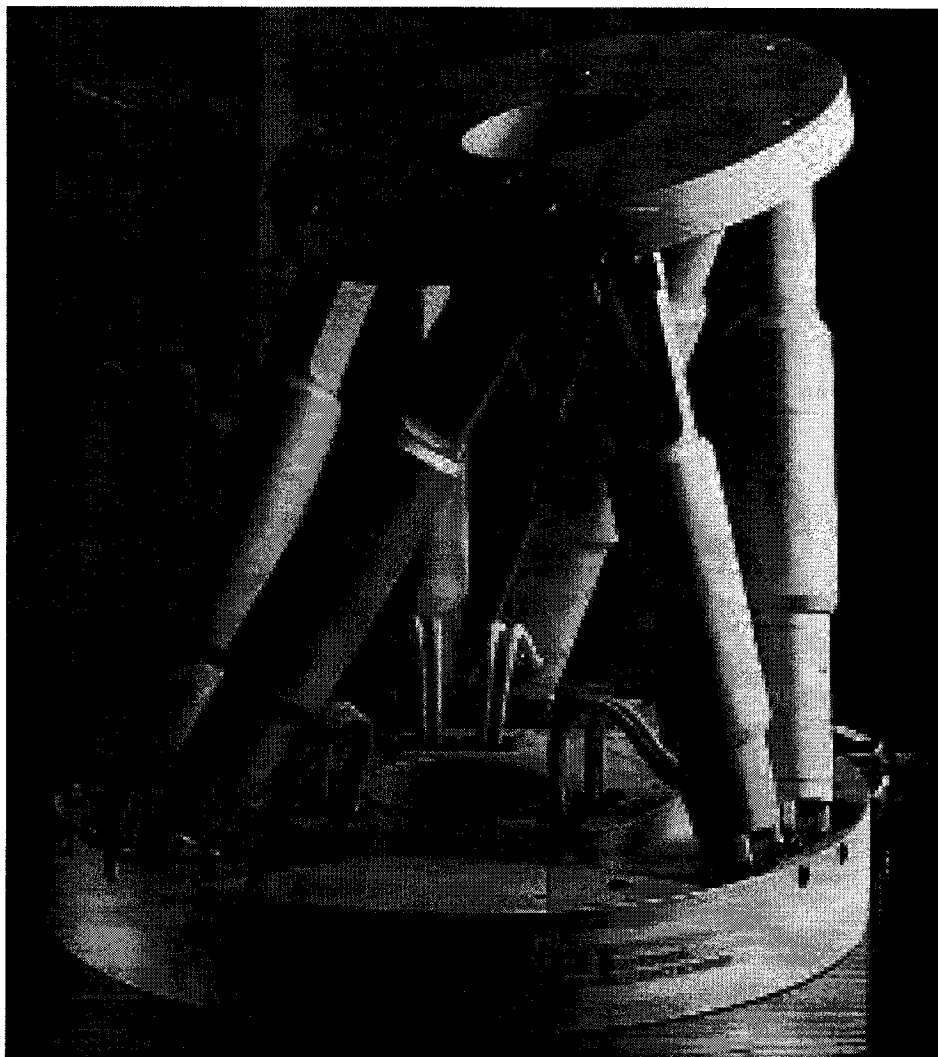


FIG. 1.1 PGS typique (Bonev 2006)

L'intérêt de la PGS, le manipulateur parallèle le plus populaire, a grandement augmenté depuis quelques années (Dasgupta et Mruthyunjava 2000). La commande d'un tel système est très difficile car plusieurs techniques de commande connues

requièrent une bonne connaissance du modèle. On peut alors recourir à certaines approximations, ou encore à des techniques de commande plus complexes. Les détails des différentes techniques seront énoncés dans la revue de la littérature.

1.2.1 Revue de la littérature sur la PGS

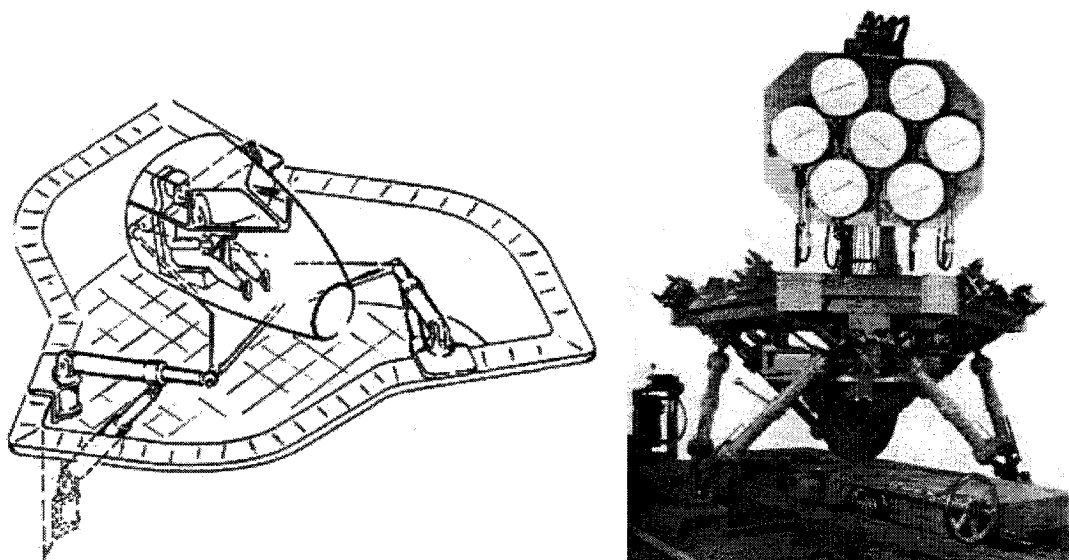
Dans le but de mettre en lumière les travaux liés aux difficultés et la façon d'aborder les différents problèmes de la PGS, un survol de la littérature est effectué.

1.2.1.1 Historique

En 1965, on propose (Stewart 1965) un mécanisme à 6 degrés de liberté illustré à la figure 1.2. Il est composé d'une plate-forme triangulaire supportée par des joints sphériques reliés à une combinaison de deux membrures extensibles jointes à une base en deux axes distincts, permettant d'ajuster localement la position d'un joint sphérique. La contribution de chaque membrure extensible nous donne une plate-forme œuvrant selon les 6 DDL de l'espace. C'est précisément ce mécanisme, voué à l'utilisation comme simulateur de vol, que Stewart introduisit dans son célèbre papier de 1965, qui eut un grand impact dans la communauté scientifique ; Tellement que d'emblée, et encore aujourd'hui, plusieurs voient Stewart comme le père des manipulateurs parallèles.

Cependant, d'autres articles traitant de mécanismes parallèles à 6 DDL, comme celui du testeur de pneu (Gough et Whitehall 1962) illustré à la figure 1.2, ont vu le jour avant celui de Stewart. Pour rendre hommage à ce mécanisme utilisant 6 actionneurs linéaires en parallèle, beaucoup plus près d'un point de vue géométrique de ce que l'on dénomme aujourd'hui plate-forme de Stewart, l'appellation plate-

forme de Gough-Stewart est fréquemment utilisée et elle est privilégiée dans ce mémoire.



(a) Plate-forme de Stewart (Proc. IMechE, 1965-66) (b) Plate-forme de Gough (Proc. IMechE, 1965-66)

FIG. 1.2 Origines de la plate-forme Gough-Stewart

Il faut attendre jusqu'aux années 80 pour voir l'intérêt des structures parallèles se transposer réellement au niveau de la robotique. Dans les années 90, cet intérêt s'est confirmé pour les manipulateurs parallèles en général et surtout pour la PGS. Du fait de sa configuration lui conférant 6 DDL, le rapprochement entre ce manipulateur parallèle et les manipulateurs sériels connus est étudié par Waldron et Hunt (1991). Géométriquement, les distinctions sont évidentes entre les deux types de manipulateur, tels qu'illustrés à la figure 1.3.

Mais d'un point de vue qualitatif, les travaux de Collins et Long (1995) font ressortir une importante distinction entre les deux types de manipulateur. En effet,

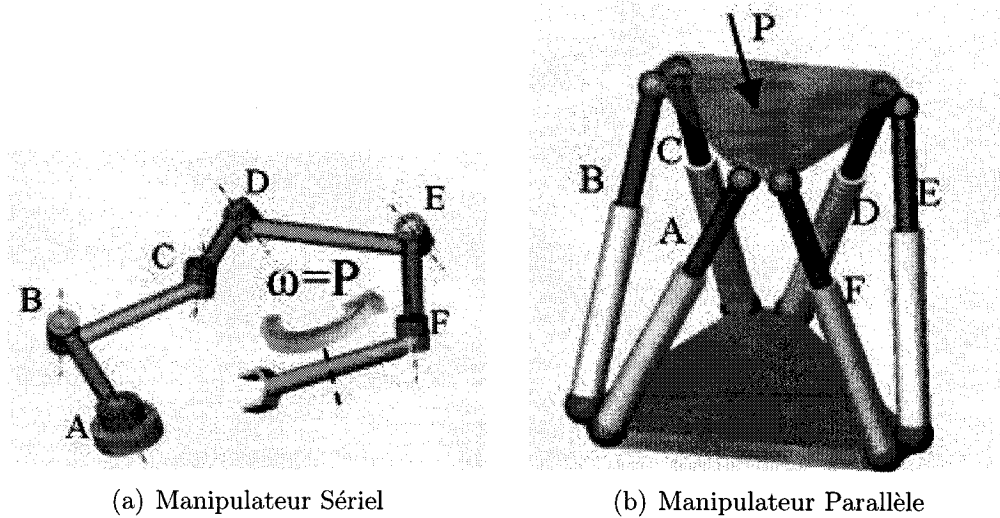


FIG. 1.3 Distinction entre manipulateurs sériel et parallèle

pendant que la cinématique directe est assez simple pour un manipulateur sériel, elle est très compliquée dans le cas des manipulateurs parallèles. À l'inverse, la cinématique inverse est complexe pour les manipulateurs sériels et simple pour les manipulateurs parallèles. Il faut se référer à la section 3.1 pour les définitions exactes de cinématique et de dynamique pour les manipulateurs.

Tel que convenu par la théorie, la situation est opposée en statique (Collins et Long 1995). Pendant que la transmission de force directe se trouve plus ou moins simplifiée en comparaison avec la transmission de force inverse, encore une fois, c'est le contraire pour les manipulateurs sériels.

1.2.1.2 Cinématique

Durant les années 90, la grande partie des efforts est consacrée au problème de cinématique directe, consistant à trouver la position et l'orientation de la plate-forme mobile pour 6 longueurs de membrures déterminées. L'architecture fermée d'une

PGS rend ce problème très complexe. Cette complexité a fait que l'évolution des différentes méthodes d'analyse dans ce domaine précis a été progressive et qu'on voit apparaître les manipulateurs parallèles comme sujet de recherche important.

Avant d'effectuer l'analyse de la cinématique sur la PGS complète, plusieurs analyses ont été effectuées sur des cas spéciaux de structures parallèles. En effet, plusieurs travaux étudient des mécanismes parallèles dont certains points de connexion sont jumelés par deux ou par trois sur l'une ou l'autre des plate-formes.

Encore très embryonnaires jusqu'ici, les travaux sur la PGS atteignent un autre stade grâce à la contribution de Fichter (1986) et de Merlet (1987). Fichter présente les équations de la cinématique pour une PGS générale, mais dont les membrures sont considérées sans masse et en négligeant la friction dans les joints. Des notions de dynamique font aussi l'objet de cet ouvrage. Merlet ajoute quelques considérations à la modélisation, traite des aspects de design et discute également d'autres applications possibles de la plate-forme. À eux deux, ces travaux englobent pratiquement tous les aspects de bases relatifs à la cinématique et à la dynamique de la PGS.

Les approches d'abord privilégiées pour le problème de cinématique directe ont été numériques. D'un point de vue pratique, ces solutions sont très avantageuses et permettent de trouver une solution réelle. Mais d'un point de vue théorique, la recherche de toutes les solutions réelles est souhaitée. Innocenti et Parenti-castelli (1993) développent un algorithme permettant de les trouver toutes. Également, Dasgupta et Mruthyunjava (1996) élaborent un algorithme pour trouver les solutions réelles basées sur des considérations géométriques seulement. Dans le but

d'obtenir toutes les solutions, Raghavan (1993) utilise une formulation polynomiale et la résolution fait ressortir 40 solutions complexes.

Les approches analytiques sont principalement de deux types. Il y a les méthodes basées sur l'établissement d'un polynôme et les méthodes basées sur la géométrie. Premièrement, Wen et Liang (1994) réduisent les équations de cinématique pour en arriver à un polynôme leur permettant d'établir, par sa résolution, le nombre maximal de solutions à 40 pour une PGS à base planaire. Aussi, Hunt et al. (1993) font sensiblement les mêmes constatations que Wen et Liang, en utilisant des arguments géométriques.

Mais malgré l'avancement engendré par tous ces auteurs sur la cinématique directe, le problème reste non résolu en entier. On veut déterminer efficacement la solution permettant une bonne simulation et la commande en temps réel. Pour ce faire, des techniques de résolution par des capteurs redondants ont été utilisées par Baron et Angeles (1994 et 2000) pour trouver une solution particulière à la cinématique directe. Aussi, la redondance permet la linéarisation et le découplage des membrures conduisant à des performances en temps réel améliorées.

1.2.1.3 Dynamique

La documentation sur la dynamique de la PGS est très limitée en comparaison avec celle sur la cinématique. Mais au tournant du siècle, les travaux sur la modélisation complète ont vu le jour en privilégiant différentes méthodes à partir de différentes considérations. Regardons les travaux sur quelques méthodes utilisées pour la synthèse de la dynamique de la PGS :

- **Newton-Euler** : La méthode basée sur les équations de Newton-Euler est la méthode utilisée dans plusieurs ouvrages. Do et Yang (1998) trouvent la solution au problème de dynamique inverse en considérant plusieurs simplifications. La modélisation plus complète de la dynamique inverse par la méthode de Newton-Euler a été réalisée par Gosselin (1996) et Dasgupta et Mruthyunjava (1998). Khalil et Guegan (2002) proposent également une représentation des membrures par éléments de dynamique cartésienne et effectuent un calcul des performances de la méthode (Khalil et Guenan 2004). Harib et Srivasan (2003) incluent la dynamique des actionneurs et la friction aux joints dans une analyse très complète d'une machine-outil.
- **Lagrange** : Le développement simplifié des équations de Lagrange pour décrire le mouvement de la plate-forme a été réalisé par Geng et al. (1992). Ji (1994) renchérit en considérant l'effet de l'inertie des membrures dans son analyse.
- **Travail virtuel** : Wang et Gosselin (1998) proposèrent une méthode basée sur des notions de travail virtuel, plus efficace numériquement que la méthode de Newton-Euler. Aussi, par l'approche de ressort virtuel (Wang et Gosselin 2002), la dynamique a été complètement découplée pour rendre la simulation encore plus performante.
- **Équations de Kane** : Liu et al. (2000) proposent des équations de la dynamique de la PGS basées sur les équations de Kane. En supposant que chacune des membrures est une entité indépendante, ils formulent la dynamique complète des membrures et de la plate-forme selon plusieurs contraintes induites par ces sous-structures. Encore une fois, l'approche se veut plus directe et plus systématique que les méthodes de Newton-Euler et celle de Lagrange. La dynamique résultante est également très concise.

1.2.1.4 Commande

En comparaison avec les analyses de la cinématique et de la dynamique, les travaux traitant de la commande de la PGS sont beaucoup moins nombreux. Toutefois, malgré la difficulté de modéliser complètement la PGS, plusieurs techniques de commande ont été appliquées. Voici quelques travaux d'envergure sur le sujet :

- **Commande robuste** : Une procédure de design de contrôleur robuste permettant de pallier aux incertitudes paramétriques de la modélisation a été développée par Kang et al. (1996). Cette méthode s'appuie sur les fondements de l'approche de Lyapunov. Déjà utilisée pour les manipulateurs sériels, Kang et al. en présentent l'adaptation pour la PGS.
- **Commande adaptative** : La méthode du « backstepping » a été implantée à la PGS par Sirouspour et Salcudean (2001). Cette méthode de commande non linéaire permet une erreur de suivi de position exponentiellement stable, et permet aussi de compenser pour les incertitudes paramétriques par une adaptation continue. Les considérations de stabilité se font une fois de plus par une analyse basée sur la stabilité de Lyapunov.
- **Rétroaction d'état** : En considérant la PGS comme un système multi-entrée multi-sortie, Cherdchoosilpa et al. (2002) présentent un contrôleur par retour d'état avec placement de pôles. D'un système non linéaire, une approximation est utilisée pour construire le contrôleur. Aussi, ce contrôleur permet de réduire les effets des perturbations par un module de commande « Feed-forward ».
- **Contrôleur Proportionnel-Intégral-Dérivée (PID)** : En plus de présenter les différents éléments de la dynamique de la PGS, Lee et al. (2003) commandent la PGS par une technique se basant sur la dynamique inverse et sur le modèle approximatif de la PGS. Ils linéarisent le système par la méthode du couple précalculé,

bien connue pour les manipulateurs sériels. Le suivi de trajectoire est assuré par un contrôleur PID, auquel est ajouté un module de compensation des erreurs de modélisation intentionnellement introduites dans la modélisation. Cette compensation est assurée par un contrôleur H_∞ .

- **Modes de glissement** : Kim et al. (1998) et plus récemment Huang et al. (2004) ont développé des contrôleurs basées sur les modes de glissement. Plus particulièrement, ils ont réalisé le design en permettant certaines incertitudes paramétriques. Se basant sur des propriétés propres à la PGS, le contrôleur développé par analyse de stabilité de Lyapunov assure la convergence de la réponse.

- **Commande prédictive** : Deux méthodes de commande sont appliquées par Nadjimi et al. (2006) basées sur des notions de commandes prédictives. L'une implique un contrôleur LQG ("Linear quadratic gaussian") pour minimiser l'effet du bruit sur la mesure des variables et l'autre est combinée avec un contrôleur par rétroaction de vitesse pour faire varier les pôles du système et d'améliorer les performances.

- **Commande par la logique floue** :

Le contrôle de la PGS par des techniques basées sur la logique floue a également déjà fait l'objet d'études. Et pour assurer l'originalité d'application qui est présentée, voici l'ensemble des travaux recensés traitant de logique floue appliquée à la PGS :

- Chung et al. (1999) utilisent le critère de Popov pour analyser la stabilité du système commandé par un contrôleur flou.

- Mann et Surgenor (2002) modélisent et commandent une PGS pour du micro-positionnement avec un contrôleur flou à deux niveaux. En effet, un premier niveau du contrôleur flou permet de générer le bon voltage pour chacun des actionneurs piézoélectriques, tandis qu'un second niveau du contrôleur assure la robustesse par

rapport aux incertitudes.

- Lin et Tsay (2000) comparent un contrôleur PID linéaire classique à un autre PID, auquel des règles floues viennent se greffer. La comparaison montre l'amélioration de la réponse par le contrôleur flou à trois dimensions.
- Garagic et Srinivasan (2004) utilisent quant à eux la logique floue pour contrer la friction dans l'application de la PGS pour une machine-outil. En utilisant le modèle flou TS, ils estiment les effets inconnues de la friction pour mieux les compenser.

1.2.1.5 Autres difficultés

En plus de la difficulté à obtenir les modèles complets de la cinématique et de la dynamique directes, d'autres problèmes se présentent lors de la commande de la plate-forme. En effet, la présence de singularités particulières et la représentation de l'espace de travail sont deux difficultés supplémentaires liées au fonctionnement de la plate-forme et méritent d'être soulignées.

Les singularités des PGS ne sont pas cinématiques comme celles des manipulateurs sériels. Premièrement, elles proviennent de la configuration propre de la plate-forme et aux limites des articulations, ce qui peut occasionner une rigidité complète de la plate-forme pour certains déplacements. Deuxièmement, certaines configurations de la PGS rendent la transmission de force impossible dans certaines directions. Ce type de singularités se produit lorsque la jacobienne est singulière.

L'espace de travail limitant toutes les positions et les orientations extrêmes de la plate-forme est très difficile à représenter. En effet, cet espace est représenté par

une combinaison de position et d'orientation relatives aux 6 DDL de la PGS.

1.3 Structure du mémoire

La suite de ce mémoire est organisée comme suit :

Le chapitre 2 introduit des notions générales de logique floue. Plus particulièrement, le modèle floue TS (Takagi et Sugeno 1985) est explicité dans le but de l'appliquer à la PGS et d'en faire la commande. Les analyses de stabilité et les procédures de design de différents contrôleurs font l'objet également de ce chapitre. La résolution de ces problèmes, énoncés sous forme de LMI, est possible par la théorie de Lyapunov.

Le chapitre 3 présente les principaux éléments de la modélisation de la PGS. On y traite de cinématique et de dynamique dans le but d'en déduire un modèle dynamique fidèle permettant d'utiliser les outils de commande retenus pour l'application.

Le chapitre 4 présente un module de simulation permettant de valider les procédures décrites au chapitre 3. Les résultats des différents contrôleurs sont exposés, et leur analyse est effectuée.

Le chapitre 5 constitue la conclusion de ce mémoire. On y retrouve le résumé de ce qui a été réalisé et on y synthétise les principaux résultats obtenus. Les limitations de certains aspects de recherche sont également présentées. Finalement, une discussion termine ce mémoire sur les extensions possibles à un tel projet.

CHAPITRE 2

APPROCHE PAR LA LOGIQUE FLOUE

2.1 Avant-propos

Dans ce chapitre, il est question d'introduire la procédure de design de correcteur complète retenue afin de commander les systèmes non linéaires. Effectivement, l'utilisation de la logique floue semble tout indiquée pour pallier aux difficultés de commande de cette classe de système. On note quelques avantages liés à l'utilisation de la logique floue au tableau 2.1.

TAB. 2.1 Avantages de l'approche par la logique floue

- | |
|--|
| <ul style="list-style-type: none">• Facilité d'implantation• Solution de problèmes multivariables• Robustesse vis-à-vis des incertitudes |
|--|

Ces avantages permettent de commander le système complexe de la PGS. Mais avant, regardons quelques bases de la logique floue permettant de mieux effectuer la procédure de design.

2.1.1 Bref historique

Aujourd'hui, on compte plusieurs ouvrages traitant de logique floue. À noter qu'on peut retrouver les principaux concepts utilisés dans ce chapitre dans plusieurs tra-

vaux traitant de logique floue (Bouchon-Meunier et Marsala 2003)(Nguyen et Walker 2006).

L'idée de logique floue ne date pas d'hier. Il y a plus de 60 ans déjà, ces concepts flous trottaient déjà dans la tête de certains philosophes (Black 1937). Cependant, il faut attendre plusieurs années avant qu'on ne les transpose dans un domaine plus concret et plus scientifique.

Zadeh (1965) a été le premier à formaliser ces concepts flous par sa théorie d'ensembles flous. Connaissant un succès très mitigé, il fallut son exportation vers l'Europe et l'Asie afin qu'elle se développe de façon plus substantielle.

Dans le domaine de la commande, le premier à utiliser la logique floue pour une application précise est Mamdani (1974). Il met en pratique la théorie de Zadeh pour une application industrielle, soit la commande d'une chaudière à vapeur.

2.1.2 Applications

Aujourd'hui, la logique floue a atteint sa pleine maturité. Le champ des applications, déjà très vaste, ne cesse d'augmenter. On y dénote des applications dans les appareils électroménagers, dans les systèmes audio-visuels, dans les automobiles, dans les engins autonomes mobiles, de transports, de conditionnement d'ambiance, pour les décisions, les diagnostics et la reconnaissance, et bien sûr, pour la commande dans à peu près tous les domaines industriels.

En fait, principalement, les travaux récemment énoncés dans la littérature tendent

vers des applications liées aux deux domaines généraux suivants (Ciocan, 2005) :

- La commande de systèmes difficilement modélisables ;
- La commande non linéaire pour des systèmes modélisables.

2.2 Approche générale de résolution

Pour la résolution avec succès d'un problème de commande par la logique floue, il faut respecter une procédure rigoureuse. Un système de commande flou peut être représenté de différentes façons. Mais le plus souvent, on y retrouve les trois étapes suivantes, schématisées à la figure 2.1 :

1. Quantification floue ;
2. Établissement des règles ;
3. Combinaison des règles.

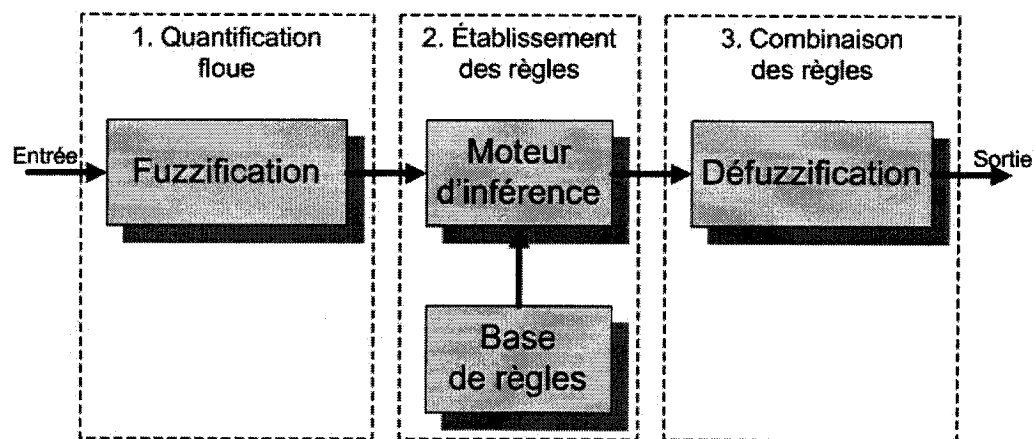


FIG. 2.1 Structure d'un système flou

Pour y aller formellement avec l'explication de chacune des étapes de la figure 2.1, il est nécessaire d'énoncer certaines définitions importantes pour la compréhension de

l'approche. De la même manière que la logique booléenne est associée à la théorie des ensembles, la logique floue est associée à la théorie des sous-ensembles flous (Zadeh 1965).

2.2.1 Quantification floue

Communément appelée fuzzyfication, cette étape permet de transformer les variables déterministes du système en variables floues. Les variables du système sont alors projetées sur des sous-ensembles flous M_i selon un degré d'appartenance $w(\cdot)$ dans l'intervalle $[0, 1]$. Les variables d'entrée du système sont donc caractérisées par des variables linguistiques. Le tout est explicité formellement dans les définitions suivantes (Figure 2.2) :

Définition 2.2.1 *Un sous-ensemble flou, M_i , d'un ensemble de référence, U , est défini par une fonction d'appartenance, w , qui associe à tout élément, z de U , une valeur réelle, $w(z)$, dans l'intervalle $[0,1]$.*

Définition 2.2.2 *Une variable linguistique est représentée par un triplet (z,U,T) dans lequel :*

z : nom de variable (âge, taille, position, vitesse,...),

U : l'univers des valeurs prises par z ($\mathbb{R}, \dots$)

$T = \{M_1, M_2, \dots\}$: un ensemble de sous-ensembles flous de U .

Une variable linguistique peut alors être représentée comme à la figure 2.2, où on y voit aussi les termes et les concepts définis.

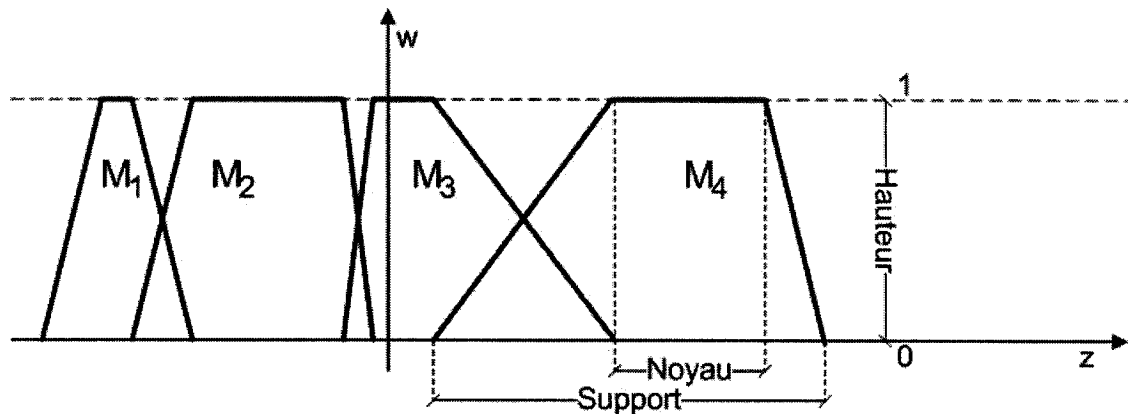


FIG. 2.2 Représentation de termes et concepts flous

Les autres termes apparaissant dans la figure 2.2 sont définies par

Noyau : Le noyau d'un sous-ensemble flou M_i de U est l'ensemble de tous les éléments lui appartenant totalement, aussi appelé le domaine des certitudes.

Support : Le support d'un sous-ensemble M_i de U est l'ensemble de tous les éléments lui appartenant au moins un petit peu, aussi appelé le domaine d'appartenance.

Hauteur : Le support d'un sous-ensemble M_i de U est la valeur maximale atteinte sur le support de M_i . Le sous-ensemble est normalisé si la hauteur est 1.

2.2.2 Établissement des règles

L'établissement des règles comprend la base des règles floues et l'inférence floue. La base des règles floues est composée des variables linguistiques et des règles linguistiques. Les variables linguistiques permettent de caractériser les états associées au système contrôlé et les règles linguistiques sont des règles floues qui ont la forme suivante :

« SI condition ALORS conclusion »

Les conclusions du modèle peuvent être de deux types.

- Les règles de type Mamdani où les conclusions comme les prémisses sont des propositions floues :

« SI x est A et y est B ALORS z est C » ;

- Les règles de type Sugeno où dans les conclusions, une sortie est fonction des entrées :

« SI x est A et y est B ALORS z est $f(x, y)$ ».

L'inférence floue est la détermination d'un mécanisme d'activation des règles floues appropriées. À chacune des règles floues est attribué un degré permettant de déterminer la valeur floue de la sortie apparaissant dans la conclusion.

2.2.3 Combinaison des règles

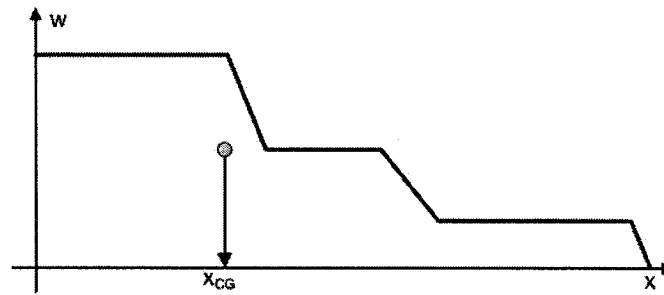
La combinaison des règles, que l'on réfère ici à la défuzzification, est l'étape inverse de la fuzzification. En fait, elle permet de retransformer toutes les variables floues en variables réelles. Il existe plusieurs façons de combiner l'inférence des règles floues. On retient celle du centre de gravité et celle de la moyenne des maxima comme exemple. Mis en équations, ces deux façons de procéder s'expriment par (Fig. 2.3) :

Centre de Gravité

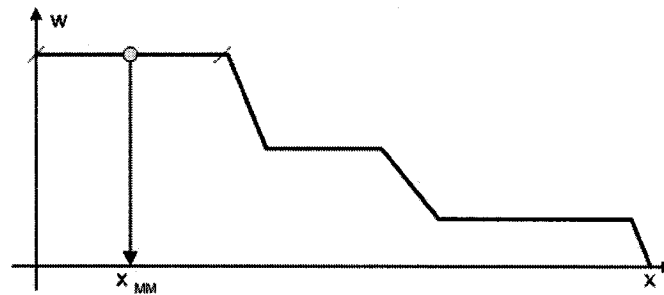
$$x_{CG} = \frac{\sum_{i=1}^n w_i(x) x_i}{\sum_{i=1}^n w_i},$$

Moyenne des Maxima

$$x_{MM} = \frac{\sum_{i=1}^n x_{iMAX}}{n}.$$



(a) Centre de Gravité



(b) Moyenne des Maxima

FIG. 2.3 Méthodes de défuzzification

2.3 Modèle flou TS

La procédure de design proprement dite débute par l'introduction d'une façon de représenter un modèle non linéaire par un modèle flou. Cette façon de faire a été développée par Takagi et Sugeno (1985).

Le modèle flou TS est décrit par des règles SI-ALORS représentant les relations entre les entrées et les sorties d'un système non linéaire. Le but du système flou est d'exprimer la dynamique locale de chaque sous-système par un modèle linéaire. Il prend alors la forme suivante :

Règle du modèle i :

SI $z_1(t)$ est M_{i1} et $\dots$ et $z_p(t)$ est M_{ip} ,

$$\textbf{ALORS} \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t), \\ y(t) = C_i x(t), \end{cases} \quad i = 1, 2, \dots, r. \quad (2.1)$$

où

z_i : prémisses du système ;

M_{ij} : sous-ensembles flous ;

A_i, B_i, C_i : matrices du système dynamique.

Chaque équation de sortie est représentée par l'espace d'état $A_i x(t) + B_i u(t)$. À chaque règle i est attribué un poids $w_i(z(t))$ qui dépend du degré d'appartenance des variables de prémisses $z_i(t)$ aux sous-ensembles flous M_{ij} . Dans ce type de modèle, la connection entre les prémisses $z_i(t)$ est de type « ET », qu'il est possible d'exprimer avec un produit.

En récapitulant, pour une paire donnée $(x(t), u(t))$, le modèle sera une sommation pondérée de plusieurs modèles linéaires

$$\begin{aligned} \dot{x}(t) &= \frac{\sum_{i=1}^r w_i(z(t)) \{A_i x(t) + B_i u(t)\}}{\sum_{i=1}^r w_i(z(t))} \\ &= \sum_{i=1}^r h_i(z(t)) \{A_i x(t) + B_i u(t)\}, \end{aligned} \quad (2.2)$$

$$\begin{aligned} y(t) &= \frac{\sum_{i=1}^r w_i(z(t)) \{C_i x(t)\}}{\sum_{i=1}^r w_i(z(t))} \\ &= \sum_{i=1}^r h_i(z(t)) \{C_i x(t)\}, \end{aligned} \quad (2.3)$$

où

$$\begin{aligned} z(t) &= [z_1(t) \ z_2(t) \ \cdots \ z_p(t)], \\ w_i(z(t)) &= \prod_{j=1}^p M_{ij}(z_j(t)), \\ h_i(z(t)) &= \frac{w_i(z(t))}{\sum_{i=1}^r w_i(z(t))}, \end{aligned}$$

$$\begin{cases} \sum_{i=1}^r w_i(z(t)) > 0, \\ w_i(z(t)) \geq 0, \quad i = 1, 2, \dots, r, \end{cases}$$

$$\begin{cases} \sum_{i=1}^r h_i(z(t)) = 1, \\ h_i(z(t)) \geq 0, \quad i = 1, 2, \dots, r. \end{cases}$$

2.4 Contrôleur PDC

Un gros avantage d'utiliser un modèle flou de ce type pour représenter un système non linéaire est la possibilité d'utiliser la même approche pour synthétiser le contrôleur.

Ainsi, maintenant que nous avons la représentation floue du modèle non linéaire, introduisons la procédure de design, communément appelée la compensation parallèlement distribuée (PDC) (Tanaka et Wang 2001). Les règles de contrôles partagent les mêmes règles SI-ALORS que le modèle flou (2.2). Nous avons alors les règles suivantes pour le contrôleur :

Règle du modèle i :

SI $z_1(t)$ est M_{i1} et $\dots$ et $z_p(t)$ est M_{ip} ,

ALORS $u(t) = -K_i x(t), \quad i = 1, 2, \dots, r.$

À noter qu'à chacune des règles est associé un contrôleur linéaire par retour d'état.

Le contrôleur complet est défini par

$$u(t) = -\frac{\sum_{i=1}^r w_i(z(t)) K_i x(t)}{\sum_{i=1}^r w_i(z(t))} = -\sum_{i=1}^r h_i(z(t)) K_i x(t). \quad (2.4)$$

L'objectif est de concevoir le contrôleur flou représenté par l'équation (2.4) pour faire la commande de systèmes non linéaires introduite au chapitre 1. Concrètement ce problème revient à déterminer les gains K_i assurant la stabilité en boucle fermée du système dynamique décrit par le système (2.2). La formulation utilisée permet de traiter les problèmes multivariables et d'aborder l'étude de la stabilité à l'aide de la méthode de Lyapunov.

2.5 Stabilité

Un des concepts les plus importants concernant les propriétés des systèmes commandés est l'analyse de la stabilité. Dans le contexte de la stabilité générale, la théorie de Lyapunov, qui fait référence aux propriétés de la représentation interne du système en terme de variables d'état est très utilisée. La méthode de Lyapunov est alors utilisable. Une fonction candidate classique de Lyapunov est basée sur une

forme quadratique, soit

$$V(x) = x^T P x, \quad P \in \mathbb{R}^{n \times n}, \quad P = P^T > 0.$$

Trouver la fonction de Lyapunov revient donc à trouver une matrice P symétrique et définie positive.

Les conditions suffisantes, basées sur la théorie de stabilité de Lyapunov, pour garantir la stabilité du système flou représenté par la relation (2.2) sont définies par le théorème suivant :

Théorème 2.5.1 (*Tanaka et Wang 2001*) *L'équilibre du système flou continu (2.2) avec $u(t)=0$ est globalement asymptotiquement stable s'il existe une matrice P symétrique et définie positive commune telle que*

$$A_i^T P + P A_i < 0, \quad i = 1, 2, \dots, r. \quad (2.5)$$

et que cette matrice commune existe pour tous les sous-systèmes.

Preuve du théorème :

Prenons la fonction de Lyapunov $V(x(t)) = x^T(t) P x(t)$. En la dérivant par rapport au temps le long de la trajectoire, nous avons

$$\dot{V}(t) = \dot{x}^T(t) P x(t) + x^T(t) P \dot{x}(t).$$

En remplaçant $\dot{x}$ et $\dot{x}^T$ par la relation (2.2) et sa transposée, et en considérant le

fait que $u(t) = 0$, nous avons

$$\begin{aligned}\dot{V}(t) &= \sum_{i=1}^r h_i(z(t))x(t)^T A_i^T P x(t) + x^T(t) P \sum_{i=1}^r h_i(z(t)) A_i x(t), \\ &= \sum_{i=1}^r h_i(z(t))x(t)^T [A_i^T P + P A_i] x(t).\end{aligned}$$

En considérant $h_i(z(t)) \geq 0$ et $\sum_{i=1}^r h_i(z(t)) = 1$, si une matrice P symétrique et définie positive existe, alors $\dot{V}(t) < 0$ implique que la relation (2.5) soit satisfaite uniquement pour des matrices A_i aux valeurs propres à partie réelle négative, i.e. que le système (2.2) est stable.

Ce théorème offre une condition suffisante pour assurer la stabilité asymptotique du modèle (2.2). Cependant, il implique de trouver une matrice commune à tous les sous-systèmes. Il existe d'autres méthodes n'exigeant pas une matrice P commune, se basant sur des fonctions de Lyapunov non quadratiques (Chilali et al. 2002).

2.6 Stabilité en boucle fermée

Pour la stabilité du système flou en boucle fermée, remplaçons (2.4) dans (2.2). Le système dynamique résultant est le suivant :

$$\dot{x}(t) = \sum_{i=1}^r \sum_{j=1}^r h_i h_j(z(t)) \{A_i x(t) - B_i K_j(t)\}.$$

En notant

$$G_{ij} = A_i - B_i K_j,$$

le système peut alors s'écrire sous la forme (Tanaka et Wang 2001)

$$\dot{x}(t) = \sum_{i=1}^r h_i^2(z(t)) G_{ii} x(t) + 2 \sum_{i=1}^r \sum_{i < j} h_i h_j(z(t)) \left\{ \frac{G_{ij} + G_{ji}}{2} \right\} x(t). \quad (2.6)$$

Et avec les conditions de stabilité du système en boucle ouverte du théorème 2.5, appliquées à (2.6), on peut déduire les conditions de stabilité du système en boucle fermée. À noter que l'expression

$$i < j \text{ s.t. } h_i h_j \neq 0,$$

utilisée dans les théorèmes suivants signifie que la condition tient $\forall i < j$ à l'exception de $h_i \cap h_j = \emptyset$.

Théorème 2.6.1 *L'équilibre du système flou continu décrit par (2.6) est globalement asymptotiquement stable s'il existe une matrice P symétrique et définie positive commune telle que*

$$G_{ii}^T P + P G_{ii} < 0, \quad (2.7)$$

$$\left(\frac{G_{ij} + G_{ji}}{2} \right)^T P + P \left(\frac{G_{ij} + G_{ji}}{2} \right) \leq 0, \quad (2.8)$$

$$i < j \text{ s.t. } h_i h_j \neq 0.$$

La preuve de ce théorème découle directement des conditions de stabilité du théorème 2.5. En effet, avec les résultats du théorème 2.6.1, il est possible de concevoir un contrôleur flou. Ce problème de stabilisabilité est équivalent à déterminer les gains K_i avec les conditions de (2.7) et de (2.8).

Parce que ces conditions ne sont pas convexes en K_i et en P , la variable $X = P^{-1}$

est introduite, et les conditions sont maintenant

$$\begin{aligned} -XA_i^T - A_iX + XK_i^TB_i^T + B_iK_iX &> 0, \\ -XA_i^T - A_iX - XA_j^T - A_jX + XK_j^TB_i^T + B_iK_jX + XK_i^TB_j^T + B_jK_iX &\geq 0, \\ i < j \text{ s.t } h_ih_j &\neq 0. \end{aligned}$$

Aussi, si nous définissons $N_i = K_iX$ tel que pour $X > 0$, nous avons $K_i = N_iX^{-1}$. Le problème de stabilité en boucle fermée est défini de la façon suivante :

Trouver $X > 0$ et N_i ($i = 1, \dots, r$) satisfaisant les relations suivantes :

$$\begin{aligned} -XA_i^T - A_iX + N_i^TB_i^T + B_iN_i &> 0 \quad (2.9) \\ -XA_i^T - A_iX - XA_j^T - A_jX + N_j^TB_i^T + B_iN_j + N_i^TB_j^T + B_jN_i &\geq 0, \\ i < j \text{ s.t } h_ih_j &\neq 0. \quad (2.10) \end{aligned}$$

où

$$X = P^{-1}, \quad N_i = K_iX. \quad (2.11)$$

Les gains K_i et la matrice P sont obtenus par

$$P = X^{-1}, \quad K_i = N_iX^{-1}. \quad (2.12)$$

2.7 Augmentation de la rapidité

Comme critère de performance, la rapidité d'un système est souvent recherchée. Pour augmenter la vitesse de réponse, il est possible d'utiliser une condition par-

ticulière à la dérivée de la fonction de Lyapunov du type $\dot{V}(x(t)) \leq -2\alpha V(x(t))$. Pour permettre une rapidité augmentée, on peut énoncer un problème de minimisation des valeurs propres généralisées (Tanaka et Wang 2001). Il s'énonce ainsi :

minimiser α

$X, N_1, \dots, N_r$

sujet à $X > 0$ et

$$-XA_i^T - A_iX + N_i^T B_i^T + B_i N_i - 2\alpha X > 0, \quad (2.13)$$

$$\begin{aligned} -XA_i^T - A_iX - XA_j^T - A_jX + N_j^T B_i^T + B_i N_j + N_i^T B_j^T + B_j N_i - 4\alpha X &\geq 0, \\ i < j \text{ s.t. } h_i h_j &\neq 0. \end{aligned} \quad (2.14)$$

où

$$X = P^{-1}, \quad N_i = K_i X.$$

Cette nouvelle condition permet d'augmenter de façon maximale la rapidité du système par l'effet de la commande. Cependant, cette vitesse peut occasionner une saturation des actionneurs et endommager le système. Sans toutefois atteindre la rapidité maximale, il est possible de spécifier un α permettant d'améliorer quelque peu la vitesse du système.

2.8 Conditions relaxées

La résolution des LMI dans la procédure de design d'un contrôleur implique la détermination d'une matrice P symétrique et définie positive. Les conditions de

stabilité (2.7)-(2.8) sont très conservatrices car elles impliquent que tous les sous-systèmes soient stables selon une matrice P commune à tous ces sous-systèmes. Si le nombre de règles impliquées dans le modèle flou est élevé, il peut être difficile de satisfaire aux conditions du théorème 2.6.1.

Des méthodes de relaxation de ces conditions ont été énoncées par plusieurs auteurs. On entend par relaxation une diminution du conservatisme de certaines méthodes. À cette fin, la synthèse de LMI modifiées a été réalisée (Tanaka et Wang 2001). Kim et Lee (Kim et Lee 2000) sont allés encore plus loin dans la relaxation, en comparant les conditions de Tanaka avec les leurs.

Le niveau de relaxation varie pour chacune des méthodes, selon les hypothèses utilisées. D'autres conditions ont ainsi été mises en œuvre par Liu et Fang (2003) et par Fang et al. (2006). Bien que la relaxation présentée est importante, la complexité de synthèse des LMI rend l'utilisation des conditions peu commode. Notons aussi que Fang s'attaque au problème de contrôleur par retour de sortie statique dans son analyse.

2.8.1 Conditions relaxées 1

S'il est difficile de trouver une matrice P commune à tous les sous-systèmes, essayons d'utiliser des conditions modifiées.

Pour diminuer ce conservatisme, on prend en considération les interactions entre les différents sous-systèmes. En effet, il est invraisemblable de voir toutes les règles floues activées en même temps. Donc, il est utile de synthétiser des LMI présentant la caractéristique de s'attarder à la quantité maximale de sous-systèmes impliqués à

la fois. On obtient alors des conditions relaxées, énoncées dans le théorème suivant :

Théorème 2.8.1 (*Tanaka et Wang 2001*) : *L'équilibre du système continu décrit par (2.6) est quadratiquement stable dans le sens large s'il existe des matrices symétriques P et N telles que*

$$P > 0, \quad N \geq 0, \quad (2.15)$$

$$G_{ii}^T P + P G_{ii} + (s - 1)N < 0, \quad (2.16)$$

$$\begin{aligned} \left(\frac{G_{ij} + G_{ji}}{2}\right)^T P + P \left(\frac{G_{ij} + G_{ji}}{2}\right) - N &\leq 0, \\ i < j \quad \text{s.t.} \quad h_i h_j &\neq 0. \end{aligned} \quad (2.17)$$

où s est le nombre maximal de sous-systèmes impliqués dans la représentation floue à un instant donné.

La preuve de ce théorème est donnée dans (Kim et Lee 2000), (Tanaka et Wang 2001). En prenant la relation suivante :

$$\sum_{i=1}^r h_i^2(z(t)) - \frac{1}{s-1} \sum_{i=1}^r \sum_{i < j} 2h_i(z(t))h_j(z(t)) \geq 0,$$

et en développant la fonction de Lyapunov quadratique pour le système (2.6), on arrive aux conditions recherchées.

2.8.2 Conditions relaxées 2

Bien que les conditions du théorème 2.8.1 soient moins conservatrices que celles du théorème 2.6.1, les interactions ont été introduites de façon analytique en se basant sur des propriétés quadratiques. Il est possible de représenter ces interactions en

termes de LMI, et d'utiliser la résolution numérique (Kim et Lee 2000).

Théorème 2.8.2 (Kim et Lee 2000) : *L'équilibre du système continu décrit par (2.6) est quadratiquement stable dans le sens large s'il existe des matrices symétriques P et X_{ii} telles que*

$$P > 0, \quad (2.18)$$

$$\Lambda_{ii}^T P + P \Lambda_{ii} + X_{ii} < 0, \quad (i = 1, \dots, r) \quad (2.19)$$

$$\Lambda_{ij}^T P + P \Lambda_{ij} + X_{ij} \leq 0, \quad (i < j \leq r) \quad (2.20)$$

$$\tilde{X} \equiv \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1r} \\ X_{12} & X_{22} & \cdots & X_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ X_{1r} & X_{2r} & \cdots & X_{rr} \end{pmatrix} > 0, \quad (2.21)$$

où $\Lambda_{ii} \equiv G_{ii}, i = 1, \dots, r$ et $\Lambda_{ij} \equiv ((G_{ij} + G_{ji})/2)$.

La preuve complète de ce théorème se trouve dans (Kim et Lee 2000).

Pour la stabilisabilité, il suffit de déterminer les gains K_i avec les conditions de (2.18) à (2.21). En prenant $Q = P^{-1}$, en prémultipliant et postmultipliant les relations (2.18-2.20) par Q , et en multipliant $\tilde{X}$ à gauche et à droite de (2.21) par $\text{diag}(Q, Q, \dots, Q)$, nous définissons le problème de stabilisabilité élargie suivant :

Trouver $Q > 0$, Y_{ii} et $N_i = K_i Q$ ($i = 1, \dots, r$) satisfaisant les LMIs suivants :

$$QA_i^T + A_i Q - N_i^T B_i^T - B_i N_i + Y_{ii} < 0, \quad (i = 1, \dots, r) \quad (2.22)$$

$$\begin{aligned} QA_i^T + A_i Q + QA_j^T - A_j Q \\ - N_j^T B_i^T - B_i N_j - N_i^T B_j^T - B_j N_i + 2Y_{ij} \geq 0, \quad (i < j \leq r), \end{aligned} \quad (2.23)$$

$$\tilde{Y} \equiv \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{1r} \\ Y_{12} & Y_{22} & \cdots & Y_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{1r} & Y_{2r} & \cdots & Y_{rr} \end{pmatrix} > 0. \quad (2.24)$$

Et les gains du contrôleur sont donnés par

$$K_i = N_i Q^{-1}. \quad (2.25)$$

2.9 Système affine

Jusqu'ici, nous avons traité de problème dit homogène, ou ayant la forme suivante :

$$\dot{x} = A_i x + B_i u \quad (2.26)$$

Une autre façon de représenter un système par un modèle flou fait intervenir un terme constant à chacune des règles floues.

La forme (2.26) peut être obtenue en linéarisant un système autour d'un point de fonctionnement (x_i, u_i) . Comme cela est expliqué dans Kim et Lee (2005), la linéarisation introduit souvent un terme constant issu d'une simplification lors de

la linéarisation. Le modèle dynamique correspondant est alors

$$\dot{x} = \sum_{i=1}^r h_i(z(t)) A_i x + B_i u + \mu_i. \quad (2.27)$$

Le contrôleur à synthétiser est alors décrit par

$$u = \sum_{i=1}^r h_i(z(t)) \{-K_i x - \sigma_i\}. \quad (2.28)$$

La synthèse du contrôleur s'effectue par la résolution des conditions du théorème suivant :

Théorème 2.9.1 (*Kim et Lee 2005*) : *Le système affine (2.27) est stabilisable quadratiquement par le contrôleur (2.28) s'il existe $\bar{P} > 0$, Y_i , λ_i et τ_{ij} tel que pour la règle nominale*

$$\bar{A}_i^T \bar{P} + \bar{P} \bar{A}_i - \begin{pmatrix} 0_{n \times (n-m)} & Y_i^T \end{pmatrix} - \begin{pmatrix} 0_{(n-m) \times n} \\ Y_i \end{pmatrix} < 0, \quad (2.29)$$

et pour les autres règles

$$\left(\begin{array}{c|c} \begin{pmatrix} \bar{A}_i^T \bar{P} + \bar{P} \bar{A}_i \\ - \begin{pmatrix} 0_{n \times (n-m)} & Y_i^T \end{pmatrix} - \begin{pmatrix} 0_{(n-m) \times n} \\ Y_i \end{pmatrix} \\ - \sum_{j=1}^n \tau_{ij} \bar{T}_{ij} \end{pmatrix} & \begin{pmatrix} \bar{P} \bar{\mu}_i - \begin{pmatrix} 0_{(n-m) \times 1} \\ \lambda_i \end{pmatrix} \\ - \sum_{j=1}^n \tau_{ij} \bar{u}_{ij} \end{pmatrix} \\ \hline \begin{pmatrix} \bar{P} \bar{\mu}_i - \begin{pmatrix} 0_{1 \times (n-m)} & \lambda_i^T \end{pmatrix} \\ - \sum_{j=1}^n \tau_{ij} \bar{u}_{ij}^T \end{pmatrix} & - \sum_{j=1}^n \tau_{ij} v_{ij} \end{pmatrix} < 0,$$

où les différents termes des relations du théorème 2.9.1 sont donnés par

$$\begin{aligned}
\bar{P} &= \begin{pmatrix} \bar{P}_1 & 0 \\ 0 & \bar{P}_1 \end{pmatrix}, \\
Y_i &= \begin{pmatrix} Y_1 & Y_2 \end{pmatrix}, \\
\Sigma_{j=1}^n \tau_{ij} \bar{T}_{ij} &= \begin{pmatrix} \Sigma_{j=1}^{n-m} \tau_{ij} \bar{T}_{ij} & 0 \\ 0 & \Sigma_{j=n-m+1}^n \tau_{ij} \bar{T}_{ij} \end{pmatrix}, \\
\sum_{j=1}^n \tau_{ij} \bar{u}_{ij} &= \begin{pmatrix} \Sigma_{j=1}^{n-m} \tau_{ij} \bar{u}_{ij} \\ \Sigma_{j=n-m+1}^n \tau_{ij} \bar{u}_{ij} \end{pmatrix}, \\
\bar{\mu}_i &= \begin{pmatrix} 0 \\ \mu \end{pmatrix}.
\end{aligned}$$

La composition de toutes ces matrices est très bien expliquée dans (Kim et Lee 2005). Cependant, en comparaison avec les autres méthodes explicitées précédemment, le théorème 2.9.1 nécessite une plus grande connaissance du système flou à commander. En effet, les valeurs de u_{ij} et de v_{ij} dépendent des sous-ensembles flous choisis pour la caractérisation du système. Si la résolution des conditions du théorème 2.9.1 est impossible, il ne faut pas conclure que le système n'est pas contrôlable, mais plutôt que la caractérisation floue du système est inadéquate.

2.10 Intégrateur

Le contrôleur (2.4) permet la stabilisation de notre système. Cependant, il se peut qu'il ne soit pas en mesure d'assurer une erreur en régime permanent qui tend vers zéro. Pour forcer l'élimination de cette erreur, utilisons l'action intégrale sur cette erreur. Il faudra donc augmenter la dimension de notre système, comme cela est fait par Korba et al. (2003). Le résultat du système flou étendu (EFS), permet

d'effectuer le suivi d'une valeur de référence tout en étant robuste à des erreurs de modélisation. Le nouveau vecteur de variables d'état permettant d'intégrer les erreurs est

$$x_e = x_{ref}(t) - y(t).$$

Le EFS résultant est donné par

$$\dot{x}_\omega(t) = A_\omega(\cdot)x_\omega(t) + B_\omega(\cdot)u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} x_{ref}(t), \quad (2.30)$$

$$y_\omega(t) = C_\omega(\cdot)x_\omega(t), \quad (2.31)$$

où

$$\begin{aligned} x_\omega(t) &= \begin{bmatrix} x \\ x_e \end{bmatrix}, \\ A_\omega(\cdot) &= \sum_{i=1}^r h_i(\delta(t)) \begin{bmatrix} A_i & 0 \\ -C_i & 0 \end{bmatrix}, \\ B_\omega(\cdot) &= \sum_{i=1}^r h_i(\delta(t)) \begin{bmatrix} B_i \\ 0 \end{bmatrix}, \\ C_\omega(\cdot) &= \sum_{i=1}^r h_i(\delta(t)) \begin{bmatrix} C_i & 0 \end{bmatrix}. \end{aligned}$$

Ce contrôleur EFS est composé de deux parties. Une première commande les états du système ; la seconde s'occupe des erreurs de suivi, tels que

$$\begin{aligned} u_1 &= \sum_{i=1}^r h_i(\delta(t)) K_i x(t), \\ u_2 &= \sum_{i=1}^r h_i(\delta(t)) K_{ei} x_e(t). \end{aligned}$$

Le contrôleur complet par retour d'état est

$$\begin{aligned}
 u &= u_2(x_e) - u_1(x), \\
 &= \sum_{i=1}^r h_i(\delta(t)) \begin{bmatrix} -K_i & K_{ei} \end{bmatrix} x_\omega(t), \\
 &= \sum_{i=1}^r h_i(\delta(t)) K_{\omega i} x_\omega(t) = K_\omega(\cdot) x_\omega(t).
 \end{aligned} \tag{2.32}$$

Les relations étendues (2.30)-(2.31) peuvent être utilisées avec d'autres conditions lors du design du contrôleur. Ainsi, il faut déterminer les gains du contrôleur K_i et K_{ei} pour chacune des règles floues. Et, tout comme les autres théorèmes énoncés jusqu'ici, il est possible d'énoncer ce problème sous forme de LMI.

2.11 Placement de pôles

Les conditions relaxées permettent une plus grande souplesse dans la résolution des LMIs permettant de trouver le contrôleur par retour d'état. Cependant, ces conditions ne permettent pas de garantir un niveau de performance du système en boucle fermée. En effet, ces performances étant associées aux pôles du système dans le plan complexe, il faut influencer leur positionnement pour modifier les performances du système. Pour ce faire, toujours dans un aspect d'inégalité matricielle, il faut introduire le concept de région- $\mathcal{D}$. Ce concept a déjà été utilisé par plusieurs auteurs pour améliorer les performances de leur système.

Chilali et al. (1999) présentent le placement de pôles robustes combiné avec un contrôleur dynamique par retour de sortie. La technique est également reprise dans un ouvrage complet sur la logique floue (Farinwata et al. 2000). En plus de l'uti-

lisation d'un intégrateur, on ajuste aussi les performances de son système à l'aide de placement de pôles (Korba et al. 2003). Plus récemment, Assawinchaichote et Nuang. (2006) utilisent le placement de pôles et la perturbation singulière en combinaison avec un contrôleur basé sur la théorie H_∞ .

Tous les travaux cités précédemment utilisent le concept de région- $\mathcal{D}$. Cette région est un lieu symétrique du plan complexe défini par l'inégalité matricielle suivante :

$$\mathcal{D} = \{z \in \mathbb{C} : L + zM + \bar{z}M^T < 0\}, \quad (2.33)$$

où L et M sont des matrices réelles telles que $L^T = L$. En d'autres termes, cette région est caractérisée par une LMI en fonction de z et $\bar{z}$. On appelle aussi fonction caractéristique $f_{\mathcal{D}}$ la fonction qui définit ce lieu du plan complexe par

$$f_{\mathcal{D}}(z) = L + zM + \bar{z}M^T. \quad (2.34)$$

Si on parle de stabilité de cette région au sens de Lyapunov, nous nous retrouvons avec l'inégalité suivante, permettant d'assurer que le système est stable et que les pôles sont également dans la région du plan complexe défini par les matrices L , M et M^T .

$$M_{\mathcal{D}}(A, X) := L \otimes X + M \otimes (XA) + M^T \otimes (A^T X) < 0, \quad (2.35)$$

où $\otimes$ est le produit de Kronecker. La preuve de cette équation de stabilité de région- $\mathcal{D}$ est présentée par Chilali et al. (1996).

Le produit de Kronecker s'effectue de la façon suivante pour une matrice $A_{n \times n}$ et

$B_{p \times p}$ quelconque, soit

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

Chacun des éléments de A est multiplié par la matrice B au complet et donne une matrice de dimension $np \times np$,

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix}.$$

Regardons maintenant quelques régions- $\mathcal{D}$ typiques illustrées à la figure 2.4 (Chilali et al. 1999) (Akhenak, 2004).

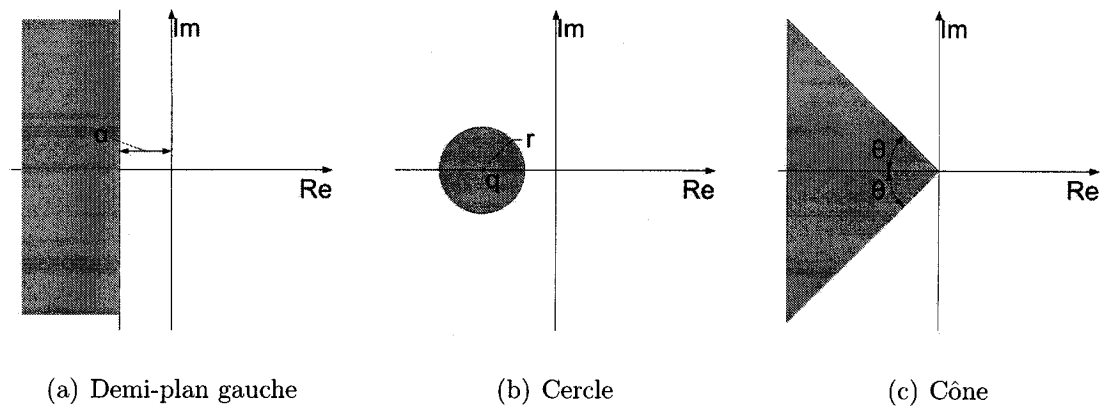


FIG. 2.4 Exemples de régions- $\mathcal{D}$

2.11.1 Région plan gauche

Cette région permet d'éloigner les pôles du système de l'axe imaginaire et ainsi influencer la rapidité du système. Plus les pôles sont éloignés, plus le système est

rapide. Il faut cependant faire attention à la saturation des actionneurs. Elle est présentée à la figure 2.4(a). La fonction caractéristique de cette région est très simple à trouver.

En posant $a = \text{Re}(z)$ et $b = \text{Im}(z)$, il vient

$$a = \frac{z + \bar{z}}{2}, \quad \text{et} \quad b = \frac{z - \bar{z}}{2j}$$

L'inégalité du demi-plan gauche décalé de α pouvant être donnée par $a < -\alpha$, la fonction caractéristique est

$$z + \bar{z} < -2\alpha.$$

En réécrivant cette condition, nous avons la fonction caractéristique suivante :

$$f_{\mathcal{D}} = z + \bar{z} + 2\alpha, \quad (2.36)$$

soit la fonction (2.34) avec

$$L = 2\alpha, \quad M = 1.$$

2.11.2 Région circulaire

La région circulaire permet de positionner les pôles de façon plus ou moins précise dans le plan complexe selon la valeur du rayon de cercle considéré. On peut alors positionner les pôles très précisément pour des performances bien définies. On peut ainsi éviter d'avoir des pôles trop rapides. La région formée par un disque centré à $(-q, 0)$ de rayon r du plan complexe, présentée à la figure 2.4(b), est une région

définie tout simplement par l'inégalité du cercle dans le plan complexe, soit

$$(\bar{z} + q)(z + q) - r^2 < 0.$$

En divisant par r de chaque côté, nous avons

$$(\bar{z} + q)r^{-1}(z + q) - r < 0.$$

Et en appliquant le complément de Schur, nous obtenons la fonction caractéristique suivante :

$$f_{\mathcal{D}} = \begin{bmatrix} -r & q + z \\ q + \bar{z} & -r \end{bmatrix}, \quad (2.37)$$

soit la fonction (2.34) avec

$$L = \begin{bmatrix} -r & q \\ q & -r \end{bmatrix}, \quad M = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

2.11.3 Région conique

Il est possible de définir en terme de région- $\mathcal{D}$ la section conique présentée à la figure 2.4(c). L'utilité de placer les pôles dans cette région permet, en diminuant l'angle θ , d'augmenter l'amortissement du système, et donc de diminuer le dépassement et le temps de réponse du système.

La fonction caractéristique de cette région se trouve en prenant la relation du cône dans le plan complexe, soit

$$a \tan \theta < |b|.$$

On prend,

$$a^2 \tan^2 \theta < b^2.$$

Ce qui donne, en réorganisant,

$$a^2 \sin^2 \theta - b^2 \cos^2 \theta < 0.$$

En divisant par $a \sin \theta$, nous avons

$$a \sin \theta - \frac{b^2 \cos^2 \theta}{a \sin \theta} < 0.$$

En appliquant le complément de Schur, nous avons la fonction caractéristique suivante :

$$f_{\mathcal{D}} = \begin{bmatrix} \sin \theta(z + \bar{z}) & \cos \theta(z - \bar{z}) \\ \cos \theta(-z + \bar{z}) & \sin \theta(z + \bar{z}) \end{bmatrix}, \quad (2.38)$$

soit la fonction (2.34) avec

$$L = 0, \quad M = \begin{bmatrix} \sin \theta & \cos \theta \\ -\cos \theta & \sin \theta \end{bmatrix}.$$

Un avantage certain de la technique de placement de pôles par l'introduction de région- $\mathcal{D}$, est que sa mise en inéquation permet l'utilisation de la résolution de LMI et qu'il est possible de la combiner avec l'ensemble des conditions de stabilité énoncées.

En résumé, tous ces théorèmes et autres outils de synthèse proviennent d'un formalisme mathématique, que nous confrontons à un système réel.

CHAPITRE 3

MODÉLISATION DE LA PLATE-FORME

3.1 Avant-propos

Dans ce chapitre, un exemple concret de système non linéaire de la forme de la classe présentée au chapitre 2 est exposé. Comme exemple de système non linéaire, nous allons considérer la PGS afin de la commander en position grâce aux méthodes se basant sur le modèle TS. Ce chapitre sert donc à modéliser la PGS afin de pouvoir utiliser le modèle TS convenablement sur cet exemple.

Pour la modélisation d'un système robotique, telle que la PGS, il faut définir les relations de cinématique et de dynamique. Ces deux problèmes peuvent être directs ou inverses. Voici quelques définitions usuelles :

La cinématique est la discipline de la mécanique qui étudie le mouvement des corps, en faisant abstraction des causes du mouvement.

La dynamique est une discipline de la mécanique classique qui étudie les corps en mouvement sous l'influence des forces qui leur sont soumises.

Les problèmes **directes** sont d'exprimer les relations de cinématique et de dynamique du manipulateur dans l'espace opérationnel en fonction des coordonnées de l'espace articulaire. Pour la PGS, ceci revient à exprimer les paramètres de la plate-forme pour des paramètres de membrures donnés.

Les problèmes **inverses** sont d'exprimer les relations de cinématique et de dynamique du manipulateur dans l'espace articulaire en fonction des coordonnées de l'espace opérationnel. Pour la PGS, ceci revient à exprimer les paramètres des membrures pour des paramètres de plate-forme donnés.

La plate-forme est un manipulateur de type parallèle, à structure fermée. Les problèmes de cinématique et de dynamique directes sont très difficiles. Cela revient à résoudre beaucoup plus d'équations que d'inconnues. En comparaison avec les manipulateurs sériels, il est très difficile, voir impossible selon la méthode utilisée, de résoudre le problème de cinématique directe de façon analytique. Ceci n'est pas très pratique du point de vue de la commande.

Heureusement, les problèmes de cinématique et de dynamique inverses sont beaucoup plus faciles à résoudre. Nous nous concentrons donc, dans cette section, sur le problème de cinématique et de dynamique inverses. Il est à noter que la façon choisie d'aborder le problème est la formulation de Newton-Euler.

3.2 Structure de la PGS

Avant de considérer les relations mathématiques régissant la cinématique et la dynamique de la PGS, attardons-nous d'abord sur la structure d'un tel système. Considérons une plate-forme de type Gough-Stewart telle que montrée à la figure 3.1. Elle nous montre la structure de la PGS composée de plusieurs corps rigides.

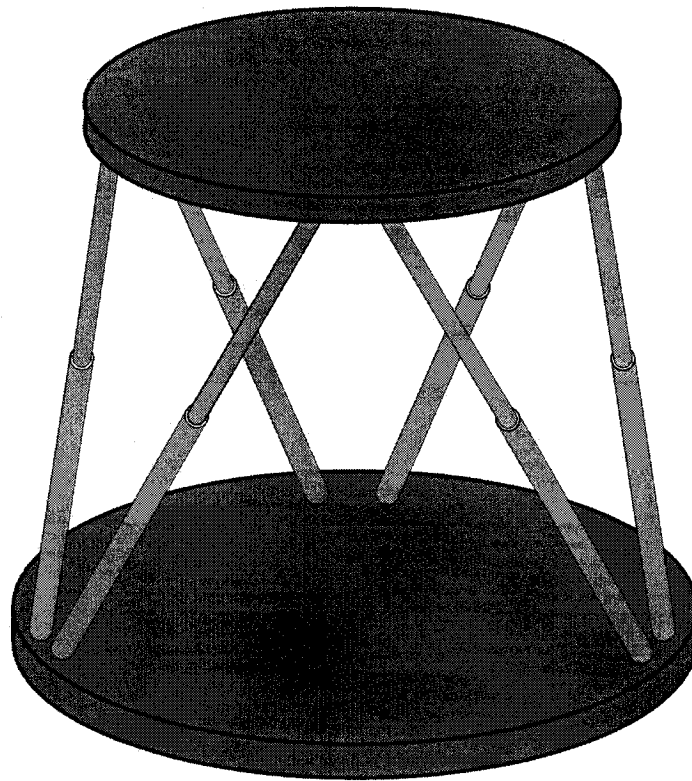


FIG. 3.1 Plate-forme de Gough-Stewart

Ces corps sont reliés entre eux à l'aide d'articulations permettant un mouvement relatif entre ces différents corps. La figure 3.2 illustre les articulations pouvant être présentes dans une PGS. À noter que pour que la plate-forme puisse se mouvoir, il faut que chacune des 6 chaînes cinématiques parallèles du mécanisme possède au moins 6 DDL.

Généralement, la base fixe est jointe aux 6 membrures extensibles à l'aide d'articulations universelles passives qui permettent la rotation selon deux axes.

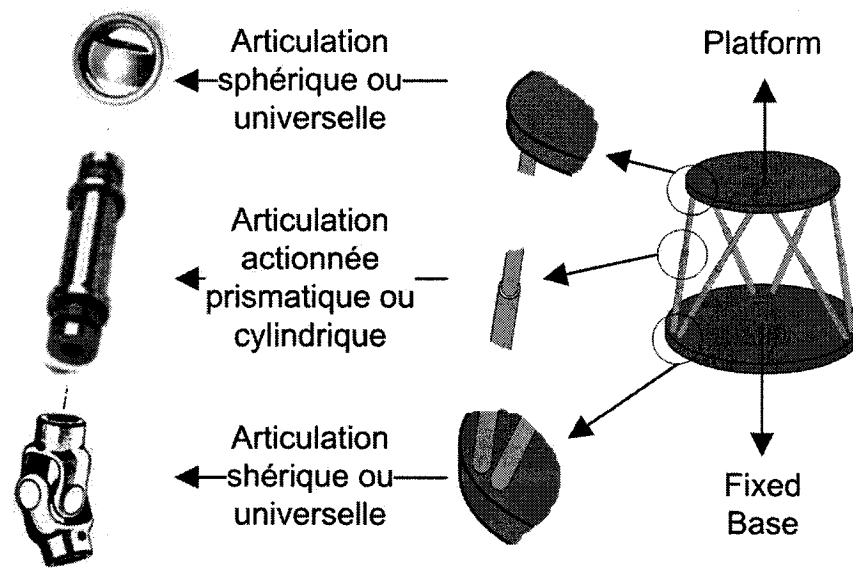


FIG. 3.2 Éléments de la plate-forme de Gough-Stewart

Les parties inférieures et supérieures des membrures sont ensuite jointes par des articulations prismatiques actionnées, permettant leur extension. Différents mécanismes, tels que des actionneurs électriques ou hydrauliques linéaires, peuvent être utilisés pour allonger ou raccourcir les membrures.

Les parties supérieures des membrures sont généralement liées à la plate-forme mobile par des articulations sphériques passives.

À noter que si deux articulations sphériques sont utilisées avec une articulation prismatique, chacune des chaînes possède un DDL non commandable. Aussi, si deux articulations universelles sont utilisées, il faut utiliser une articulation cylindrique entre les deux membrures.

3.3 Choix des coordonnées

Une fois les composantes physiques de la PGS introduite, il faut en arriver à déterminer les paramètres propres à la modélisation. Pour modéliser un système mécanique, il faut connaître le comportement des paramètres de position, de vitesse et d'accélération de tous les corps du mécanisme et leur évolution dans le temps suite à une interaction avec l'environnement. Pour ce faire, il faut d'abord déterminer des repères sur chacun des corps du système. Ceci permet de déterminer la position de tous les points appartenant au corps en question.

Ensuite, il faut introduire des systèmes de coordonnées, permettant de positionner les corps les uns par rapport aux autres.

3.3.1 Repères des corps

La figure 3.3 montre l'attribution des repères pour les 8 corps du système. Tel qu'illustré, un référentiel fixe ($\mathcal{B}$)(Fig. 3.3(a)) est solidaire au centre de la base et un référentiel mobile ($\mathcal{P}$)(Fig. 3.3(c)) est attaché au centre de la plate-forme mobile. Aussi, on définit un repère solidaire à chacun des joints d'une membrure (Fig. 3.3(b)) ; il est possible de représenter ces points d'attache par des vecteurs d'élongation résultant d'une simple analyse géométrique de la PGS.

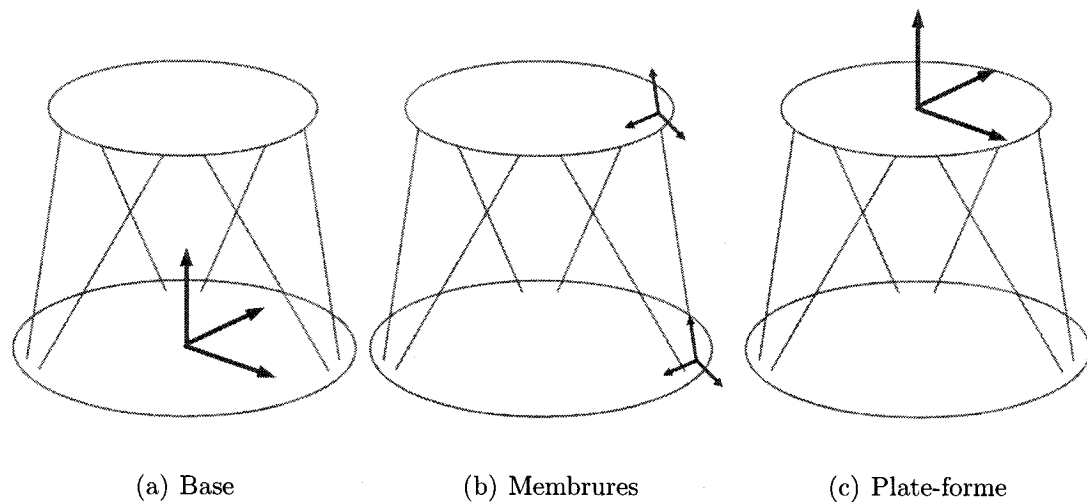


FIG. 3.3 Attribution des repères sur les corps de la PGS

3.3.2 Systèmes de coordonnées

Puisque la PGS est un mécanisme à 6 DDL, il faut donc un système à 6 coordonnées indépendantes pour positionner complètement le mécanisme. Comme tous les systèmes robotiques, on peut modéliser le système selon deux espaces bien distincts. Ces espaces sont l'espace opérationnel et l'espace articulaire.

L'espace de travail représente l'ensemble des combinaisons de positions et d'orientations pouvant être prises dans l'espace par le manipulateur du système robotique. Pour la PGS, le manipulateur étant généralement positionné au centre de la plate-forme mobile, on considère la modélisation de ce point bien précis comme étant l'espace opérationnel.

L'espace articulaire quant à lui est l'ensemble des valeurs que peuvent prendre les articulations d'un système mécanique. Ici, les articulations considérées sont les

joints prismatiques articulés.

3.3.2.1 Tenseur de coordonnées dans l'espace de travail

Il est très difficile d'analyser l'espace de travail d'un mécanisme parallèle à 6 DDL. Pour simplifier l'analyse de la PGS, nous faisons une distinction nette entre la position et l'orientation. La position de la plate-forme par rapport à la base est déterminée par le vecteur h , tel que

$$h = [x \quad y \quad z]^T.$$

L'orientation de la plate-forme est déterminée par trois angles d'Euler. Nous définissons le vecteur d'orientation par

$$e = [\phi \quad \theta \quad \psi]^T.$$

La PGS peut effectivement se déplacer et s'orienter selon les trois axes. Le tenseur q qui correspond à cette réalité est défini par

$$\begin{aligned} q &= [h^T \quad e^T]^T, \\ &= [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T. \end{aligned} \tag{3.1}$$

3.3.2.2 Vecteur des coordonnées dans l'espace articulaire

Chacune des membrures de la PGS comportent 3 articulations, pour un total de 18. Bien évidemment, il n'est pas nécessaire d'obtenir de l'information sur l'ensemble des joints constituant la plate-forme pour modéliser la plate-forme. Seulement 6

coordonnées indépendantes suffisent.

Les coordonnées des articulations prismatiques représentées par l'élongation des membrures sont données par

$$L = [L_1 \quad L_2 \quad L_3 \quad L_4 \quad L_5 \quad L_6]^T. \quad (3.2)$$

3.4 Autres considérations

Avant de commencer les analyses de cinématique et de dynamique, rappelons certains concepts importants pour la modélisation de systèmes mécaniques. Ces concepts dépendent de la méthode utilisée pour représenter la PGS. Ces autres considérations sont d'une importance capitale lorsque vient le temps de faire le passage d'un système de coordonnées à un autre.

3.4.1 Matrice de Rotation

Pour représenter l'orientation de la plate-forme, le choix privilégié est l'utilisation des angles d'Euler. Bien que les angles d'Euler ne soient pas les plus commodes, ils font partie d'un repérage classique en mécanique du solide.

Le but de cette technique est de passer de l'orientation d'un repère à un autre par trois rotations et à l'aide de deux repères intermédiaires. La figure 3.4 montre le passage en trois rotations d'un repère à un autre. La chaîne de rotation élémentaire est décrit de la façon suivante :

1. XYZ est transformé en X'Y'Z par la rotation d'angle ϕ autour de Z.
2. X'Y'Z est transformé en xY'Z' par la rotation d'angle θ autour de Y'.

3. $xY'Z'$ est transformé en xyz par la rotation d'angle ψ autour de x .

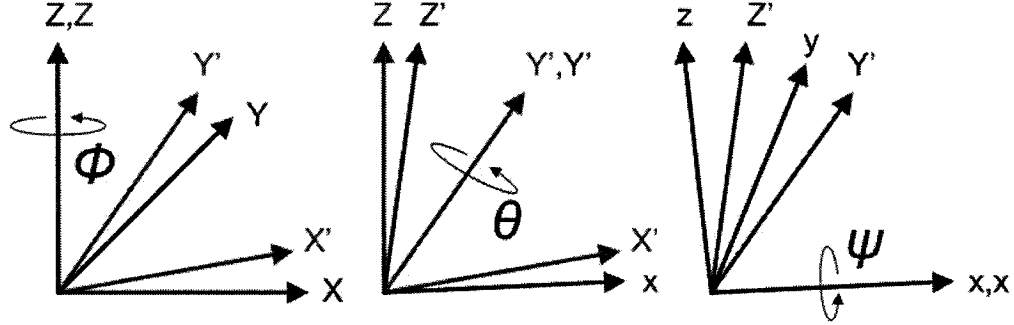


FIG. 3.4 Angles d'Euler

Pour notre PGS, une matrice de passage 3×3 est construite pour exprimer l'orientation de la plate-forme $\mathcal{P}$ selon le référentiel de la base $\mathcal{B}$. Construite avec les trois angles d'Euler ϕ, θ, ψ , elle prend la forme suivante :

$${}^{\mathcal{B}}R_{\mathcal{P}} = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix}, \quad (3.3)$$

où $c\phi$ représente $\cos \phi$ et $s\phi$ représente $\sin \phi$.

3.4.2 Vitesse et accélération angulaire

La matrice de rotation introduite à la section 3.4.1 est utile également lorsque vient le temps d'exprimer les vitesses et les accélérations angulaires de la plate-forme selon les trois axes (x, y, z) en fonction des angles d'Euler (ϕ, θ, ψ) .

La relation donnant la vitesse angulaire ω est donnée par

$$\omega = \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix} = \begin{bmatrix} -s\theta & 0 & 1 \\ c\theta s\psi & c\psi & 0 \\ c\theta c\psi & -s\psi & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (3.4)$$

Pour l'accélération, il suffit de dériver la relation (3.4) pour obtenir

$$\begin{aligned} \alpha &= \begin{bmatrix} \alpha_X \\ \alpha_Y \\ \alpha_Z \end{bmatrix} \\ &= \begin{bmatrix} -s\theta & 0 & 1 \\ c\theta s\psi & c\psi & 0 \\ c\theta c\psi & -s\psi & 0 \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} -\dot{\theta}c\theta & 0 & 0 \\ -\dot{\theta}s\theta s\psi + \dot{\psi}c\theta c\psi & -\dot{\psi}s\psi & 0 \\ -\dot{\theta}s\theta c\psi - \dot{\psi}c\theta s\psi & -\dot{\psi}c\psi & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \end{aligned} \quad (3.5)$$

3.4.3 Matrices jacobienes

Les jacobienes d'un système mécanique sont des matrices très importantes. Dans le cas d'un manipulateur sériel, le tenseur de coordonnées généralisées est celui de l'espace articulaire. Les jacobienes sont les matrices qui lient les vitesses dans l'espace opérationnel à la dérivée du tenseur de coordonnées généralisées.

Pour la PGS, c'est un peu différent. Les jacobienes lient les vitesses dans l'espace articulaire $\dot{L}$ à la dérivée des coordonnées de l'espace de travail $\dot{q}$. Pour être conforme à la définition formelle de la jacobienne, on utilise la forme inverse de la

jacobienne pour la PGS. Cette relation s'exprime par

$$\dot{L} = J^{-1}\dot{q}. \quad (3.6)$$

où J est la jacobienne complète du manipulateur. Elle est déterminée lors de l'analyse subséquente de la cinématique.

3.5 Cinématique

Pour la PGS, exprimer les relations cinématiques de la plate-forme en fonction des paramètres cinématiques des membrures est une tâche très laborieuse. L'utilisation de logiciel de modélisation mécanique permet de simplifier cette modélisation.

Cependant, la cinématique inverse, consistant à déterminer les paramètres cinématiques des membrures résultant d'une cinématique de plate-forme bien définie, est beaucoup plus simple. Dans l'application qui est proposée, ces relations sont très importantes. En effet, il faut être capable d'exprimer les élongations de membrures associées à un positionnement très précis de la plate-forme.

À l'aide de ce qui a été développé aux sections 3.2, 3.3 et 3.4, ainsi que de la figure de synthèse 3.5, les relations de cinématique inverse sont développées.

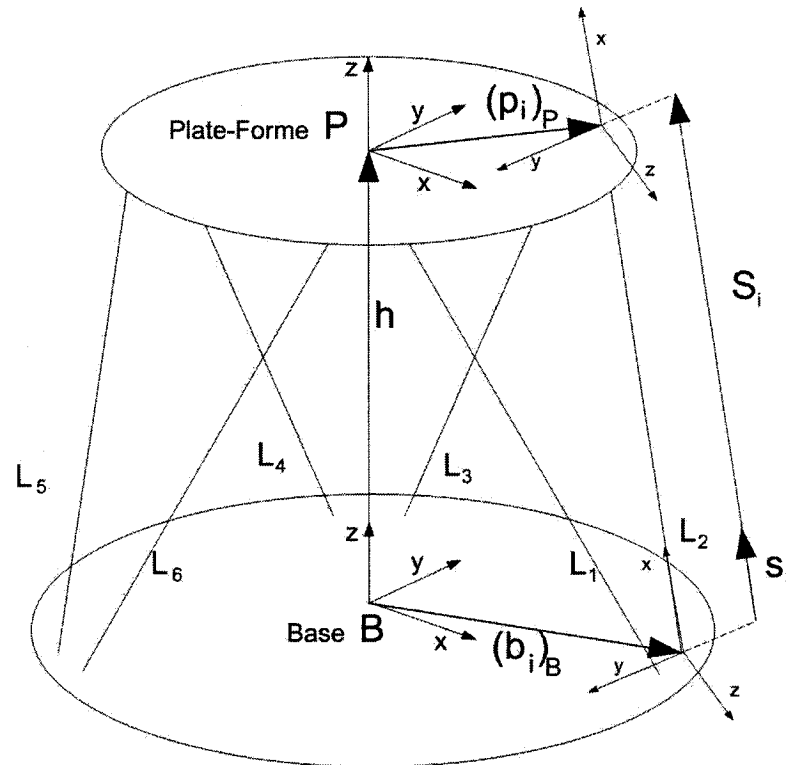


FIG. 3.5 Vecteurs pour exprimer la cinématique du système

3.5.1 Relation de position et d'orientation

Pour définir les élongations des membrures L_i en fonction d'une position de plate-forme q , nous développons l'analyse vectorielle schématisée à la figure 3.5. La position de la plate-forme est donnée par le vecteur h , contenant trois coordonnées pour la position (x, y, z) et les angles d'Euler sont utilisés pour définir son orientation, soit les trois coordonnées du vecteur e (ϕ, θ, ψ) . Le tenseur de coordonnées généralisées q étant donné par rapport au repère fixe de la base $\mathcal{B}$, il nous faut donc ramener tous les autres vecteurs à ce même repère.

Par une simple addition vectorielle, il est possible de trouver les vecteurs d'élongation des membrures S_i . Ces vecteurs sont solidaires à l'axe longitudinal des membrures de normes égales aux élongations respectives des membrures. Par une somme vectorielle, nous avons le vecteur suivant pour une membrure

$$S_i = (p_i)_{\mathcal{B}} - (b_i)_{\mathcal{B}}, \quad S_i > 0.$$

Cependant, on remarque que si le vecteur du point d'attache (p_i) d'une membrure au point supérieur de la membrure est donné en terme du repère local $\mathcal{P}$, i.e $(p_i)_{\mathcal{B}}$, il faut utiliser le vecteur de position h , ainsi que la matrice de rotation ${}^{\mathcal{B}}R_{\mathcal{P}}$ pour ramener le tout dans le même repère. La relation résultante est la suivante :

$$S_i = {}^{\mathcal{B}}R_{\mathcal{P}} (p_i)_{\mathcal{P}} + h - (b_i)_{\mathcal{B}}. \quad (3.7)$$

Pour trouver les vecteurs d'élongations des membrures, il suffit de prendre la norme vectorielle des vecteurs d'élongations, telle que

$$L_i = \|S_i\|, \quad (3.8)$$

où $\|\cdot\|$ est une norme vectorielle. Sous forme de matrice, la relation de positionnement inverse s'exprime par

$$L = [\|S_1\| \ \|S_2\| \ \|S_3\| \ \|S_4\| \ \|S_5\| \ \|S_6\|]^T. \quad (3.9)$$

3.5.2 Relation de vitesse

Pour exprimer la vitesse des élongations de membrures en fonction de la vitesse de la plate-forme, nous utilisons la relation (3.7) déjà développée. La vitesse d'un vecteur d'élongation est sa dérivée par rapport au temps, soit

$$\dot{S}_i = \dot{h} + \omega \times {}^B R_{\mathcal{P}} \cdot (p_i)_{\mathcal{P}}.$$

Pour en déduire la vitesse des élongations de membrures, il faut faire la projection des vecteurs d'élongations sur l'axe longitudinal de chacune des membrures. Il est alors utile de définir des vecteurs unitaires d'élongations, solidaires aux axes des membrures et de norme égale à un. Ce vecteur est

$$s_i = \frac{S_i}{\|S_i\|} = \frac{S_i}{L_i}. \quad (3.10)$$

Il ne reste plus qu'à réaliser cette projection, ce qui nous donne

$$\dot{L}_i = \dot{h} \cdot s_i + \omega \cdot ({}^B R_{\mathcal{P}}(p_i)_{\mathcal{P}}) \times s_i. \quad (3.11)$$

Cette dernière relation nous ramène à la définition des jacobienes introduite à la section 3.4.3. En fait, si nous réécrivons l'équation (3.11), nous introduisons une première jacobienne ${}^1 J^{-1}$, telle que

$$\dot{L} = {}^1 J^{-1} \begin{bmatrix} \dot{h} \\ \omega \end{bmatrix}, \quad (3.12)$$

où

$${}^1J^{-1} = \begin{bmatrix} s_1^T & ({}^BR_{\mathcal{P}}(p_1)_{\mathcal{P}} \times s_1)^T \\ \vdots & \vdots \\ s_6^T & ({}^BR_{\mathcal{P}}(p_6)_{\mathcal{P}} \times s_6)^T \end{bmatrix}.$$

Cependant, cette relation inclut la vitesse angulaire de la plate-forme ω , ce qui n'est pas équivalent à l'équation (3.6). Pour respecter la convention basée sur les angles d'Euler, il est utile d'introduire une seconde matrice jacobienne, permettant d'exprimer la vitesse angulaire ω à l'aide des angles d'Euler. À noter que cette deuxième jacobienne vient de la relation (3.4), soit

$${}^2J^{-1} = \begin{bmatrix} \mathbb{I}_{3 \times 3} & 0_{3 \times 3} & & \\ & -s\theta & 0 & 1 \\ 0_{3 \times 3} & c\theta s\psi & c\psi & 0 \\ & c\theta c\psi & -s\psi & 0 \end{bmatrix}. \quad (3.13)$$

La relation complète entre une vitesse donnée de la plate-forme et les vitesses d'élongation des articulations prismatiques est donnée par

$$\dot{L} = {}^1J^{-1} {}^2J^{-1} \dot{q} = J^{-1} \dot{q}. \quad (3.14)$$

3.5.3 Relation d'accélération

Dans le même ordre d'idée, l'accélération d'élongation est la dérivée de (3.14),

$$\ddot{L} = J^{-1} \ddot{q} + \frac{d(J^{-1})}{dt} \dot{q}. \quad (3.15)$$

La dérivée de la Jacobienne par rapport au temps est

$$\frac{d(J^{-1})}{dt} = \frac{d(^1J^{-1})}{dt} {}^2J^{-1} + {}^1J^{-1} \frac{d(^2J^{-1})}{dt},$$

où

$$\frac{d(^1J^{-1})}{dt} = \begin{bmatrix} (\omega_1 \times s_1^T) & ((\omega_1 \times {}^B R_{\mathcal{P}}(p_1)_{\mathcal{P}}) \times s_1 + {}^B R_{\mathcal{P}}(p_1)_{\mathcal{P}} \times (\omega_1 \times s_1))^T \\ \vdots & \vdots \\ (\omega_6 \times s_6^T) & ((\omega_6 \times {}^B R_{\mathcal{P}}(p_6)_{\mathcal{P}}) \times s_6 + {}^B R_{\mathcal{P}}(p_6)_{\mathcal{P}} \times (\omega_6 \times s_6))^T \end{bmatrix},$$

et

$$\frac{d(^2J^{-1})}{dt} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & & \\ & -\dot{\theta}c\theta & 0 & 0 \\ 0_{3 \times 3} & -\dot{\theta}s\theta s\psi + \dot{\psi}c\theta c\psi & -\dot{\psi}s\psi & 0 \\ & -\dot{\theta}s\theta c\psi - \dot{\psi}c\theta s\psi & -\dot{\psi}c\psi & 0 \end{bmatrix}.$$

Ainsi, avec les relations (3.9), (3.14) et (3.15), il est possible de complètement déterminer la cinématique des articulations prismatiques pour une cinématique donnée de la plate-forme. Ce problème représente la cinématique inverse de la PGS.

Le problème de cinématique directe est celui d'obtenir la cinématique de la plate-forme pour une cinématique donnée des articulations prismatiques. Et puisque ce problème est très difficile à implanter en temps réel, la modélisation mécanique est souhaitée. Aussi, les relations cinématiques à elles seules ne permettent pas la modélisation complète de la PGS. Les relations de dynamique sont également nécessaires.

3.6 Dynamique

Pour un système robotique, l'équation dynamique est assimilable à une équation différentielle du deuxième ordre que l'on retrouve souvent dans la littérature sous la forme

$$\tau = M(q)\ddot{q} + N(q, \dot{q}) + G(q). \quad (3.16)$$

où

τ : vecteur des forces et des couples agissant sur le système ;

$M(q)$: la matrice de masse et d'inertie du système ;

$N(q)$: matrice contenant les forces et les couples centrifuges et de Coriolis ;

$G(q)$: matrice contenant les forces et les couples dus à la gravité.

Pour la PGS, il s'agit de l'équation dynamique de la plate-forme. En fait, il s'agit des relations dynamiques de la PGS dans l'espace opérationnel.

Du fait que les actionneurs soient directement associés aux joints prismatiques, il faut ramener ces relations de forces linéaires en terme de forces et de couples agissant sur la plate-forme. La relation fait intervenir la transposée des jacobienes, telle que

$$\tau = J^{-T}F, \quad (3.17)$$

où F est le vecteur des forces appliquées par les articulations prismatiques.

3.6.1 Dynamique dans l'espace articulaire

Nous pouvons alors obtenir le modèle dynamique dans l'espace des articulations prismatiques en utilisant les équations (3.16) et (3.17), ce qui donne

$$J^{-T}F = M(q)\ddot{q} + N(q, \dot{q}) + G(q). \quad (3.18)$$

Il nous faut ensuite exprimer l'accélération $\ddot{q}$ en terme d'accélération des membrures $\ddot{L}$. En isolant $\ddot{q}$ dans la relation (3.15), nous avons

$$\ddot{q} = J\ddot{L} - J\frac{dJ^{-1}}{dt}\dot{q}. \quad (3.19)$$

En remplaçant l'équation (3.19) dans (3.18), nous avons

$$J^{-T}F = M(q)(J\ddot{L} - J\frac{dJ^{-1}}{dt}\dot{q}) + N(q, \dot{q}) + G(q).$$

Et par une simple réorganisation, nous obtenons l'équation de la dynamique dans l'espace articulaire, telle que

$$F = M_j(q)\ddot{L} + N_j(q, \dot{q}) + G_j(q), \quad (3.20)$$

où

$$\begin{aligned} M_j(q) &= J^T M(q) J, \\ N_j(q) &= J^T N(q, \dot{q}) - J^T M(q) J \frac{dJ^{-1}}{dt} \dot{q}, \\ G_j(q) &= J^T G(q). \end{aligned}$$

3.7 Représentation interne

La représentation interne est d'une grande importance dans l'étude des systèmes mécaniques. Elle permet de modéliser une panoplie de systèmes, qu'ils soient linéaires ou non linéaires, variants ou invariants, à une ou plusieurs grandeurs d'entrée et de sortie. Elle se prête bien à l'étude des systèmes dynamiques et au développement de programmes d'ordinateurs. La façon d'obtenir une représentation interne à partir des équations différentielles de notre système est expliquée dans Boukas (Boukas 1995).

De l'équation (3.20), la représentation interne sera définie pour la PGS. De plus, il est possible de linéariser le modèle afin d'en déterminer un modèle d'état autour d'un point de fonctionnement.

3.7.1 Modèle non linéaire

Pour obtenir le modèle non linéaire, isolons $\ddot{L}$ dans l'équation (3.20), ce qui donne

$$\ddot{L} = M_j(q)^{-1}(F - N_j(q, \dot{q}) - G_j(q)). \quad (3.21)$$

Nous définissons un nouveau vecteur d'état du système, tel que

$$Z = [L \quad \dot{L}]^T.$$

Le modèle non linéaire désiré est donné par la relation suivante

$$\dot{Z} = \begin{bmatrix} 0 & \mathbb{I} \\ 0 & 0 \end{bmatrix} Z - \begin{Bmatrix} 0 \\ M_j(q)^{-1}[N_j(q, \dot{q}) + G_j(q)] \end{Bmatrix} + \begin{Bmatrix} 0 \\ M_j(q)^{-1}F \end{Bmatrix}. \quad (3.22)$$

À noter que l'équation (3.22) fait partie de la classe suivante de système :

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

En connaissant la force F appliquée par les membrures, la cinématique de la plateforme, i.e. $(q, \dot{q})$, et les autres paramètres du système, il est possible d'obtenir l'accélération d'élongation des articulations prismatiques. Encore une fois, si la modélisation du système (3.21) est réalisée par un logiciel de modélisation mécanique, les différents paramètres ne sont pas explicitement requis pour l'obtention du modèle. Toutefois, il est important de bien comprendre les différentes relations, pour une conception adéquate du contrôleur.

3.7.2 Modèle linéaire

Le modèle TS est défini par plusieurs modèles linéaires. Il faut alors linéariser le modèle dans le but d'utiliser le modèle TS. Les techniques de linéarisation sont nombreuses.

3.7.2.1 Développement en série de Taylor

Rappelons d'abord une méthode bien connue pour linéariser le système autour d'un point de fonctionnement (x_r, u_r) . Cette technique se base sur l'utilisation du

développement en série de Taylor.

$$A = \frac{\partial f}{\partial x}(x_r, u_r) = \begin{bmatrix} \frac{\partial f}{\partial x} & \dots & \frac{\partial f}{\partial x} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x} & \dots & \frac{\partial f}{\partial x} \end{bmatrix}_{|x_r, u_r},$$

$$B = \frac{\partial f}{\partial u}(x_r, u_r) = \begin{bmatrix} \frac{\partial f}{\partial u} & \dots & \frac{\partial f}{\partial u} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial u} & \dots & \frac{\partial f}{\partial u} \end{bmatrix}_{|x_r, u_r}.$$

où A et B sont des matrices qui dépendent du point de linéarisation et par conséquent, elles varient selon ce point de linéarisation (Boukas 1995).

3.7.2.2 Linéarisation par MATLAB

Un modèle non linéaire peut également se linéariser directement en utilisant des fonctions MATLAB (2002) telles que *linmod*, *dlinmod* et *linmod2*. Le point d'équilibre peut, quant à lui, être trouvé grâce à la fonction *trim*. Ce point est un point d'équilibre en régime permanent, où les états ne varient plus. Si la dérivée des états par rapport au temps ne s'annule pas, la fonction *trim* prendra la valeur la plus près de 0. L'appel de cette fonction se fait de la façon suivante :

```
[x, u, y]=trim('sys', 0x, 0u, 0y )
```

où x , u et y sont les valeurs de sortie pour le point d'équilibre du système « sys » et $0x$, $0y$ et $0z$ est le point d'essai initial entré par l'utilisateur.

La fonction *linmod* permet d'obtenir un système linéaire à partir d'un modèle dynamique Simulink (Simulink 2002). La fonction *dlinmod* s'utilise généralement

pour des systèmes discrets, qui ne font pas l'objet d'études dans ce mémoire. L'appel de cette fonction se fait de la façon suivante :

```
[A, B, C, D] = linmod ('sys', x, u)
```

où x et u sont les sorties de la fonction *trim* et A , B , C et D sont les matrices résultantes de la linéarisation au point de fonctionnement.

3.7.2.3 Linéarisation multi-modèles

Pour en arriver à la modélisation multi-modèles linéaire TS de la PGS, il suffit de prendre l'équation de la dynamique du système 3.22 et de la linéariser en différents points d'opération. Ces points d'opération peuvent facilement être obtenus par les relations de cinématique inverse développées au chapitre 2.

Une fois les modèles linéaires obtenus, il faut procéder à la méthode systématique de modélisation floue. Les détails de la procédure appliquée à la PGS apparaissent à la section 4.4.

CHAPITRE 4

EXPÉRIMENTATIONS

4.1 Avant-Propos

Dans ce chapitre, les notions développées aux chapitres 2 et 3 sont utilisées pour conduire l'expérimentation via la simulation de la commande de la PGS par la méthode choisie.

Pour contourner les nombreuses difficultés liées à la commande de système non linéaire, les relations de cinématique et de dynamique directes ne sont pas développées directement. En effet, il est possible de créer des modules de simulation en utilisant un logiciel de modélisation mécanique (MATLAB 2002) (Simulink 2002). Seuls les paramètres massiques ainsi que toutes les considérations géométriques sont requis dans l'élaboration des modules de simulation.

La PGS a d'ailleurs déjà été modélisée par Smith et Wendlandt (2002) en utilisant la modélisation mécanique du logiciel SimMechanics (Simulink 2002). Un contrôleur PID fait également partie de cette article, mais le formalisme de synthèse de ce contrôleur n'est pas présenté.

Tout récemment, l'article de Nadimi (Nadimi et al. 2006) montre une autre application du modèle développé par Smith et Wendlandt (2002). Le modèle de la PGS de Smith est utilisé, en y apportant des modifications. Ensuite, la modélisation

du contrôleur est faite par des modules programmés en Simulink (2002). Pour la synthèse des gains des différents contrôleurs, les LMI énoncées au chapitre 3, la programmation est faite à l'aide du logiciel Yalmip (Löfberg 2004), fonctionnant sous l'interface de MATLAB et la résolution est faite par un logiciel d'optimisation (SeDuMi 2005).

4.2 Modèle de simulation

Pour élaborer la PGS avec le logiciel de modélisation mécanique, la procédure utilisée au chapitre 3 s'avère également fort utile. Et dans un but de faire correspondre la procédure du chapitre 3 aux particularités des logiciels utilisés, il faut se servir d'une procédure systématique.

4.2.1 Structures

La structure globale du système à commander est illustrée à la figure 4.1. On y voit les différents modules représentant les corps du système et leur positionnement les uns par rapport aux autres. À noter toutefois que la base n'est pas représentée de façon explicite. Elle sert de point de repère fixe pour les corps mobiles du système.

À la figure 4.1, on dénote aussi que la plate-forme est directement un module de corps par la présence d'un symbole de centre de gravité. Pour ce qui est des 6 membrures, elles n'y sont pas représentées comme des corps. En effet, il faut regarder la figure 4.2, représentant tous les modules composant chacune des membrures, pour visualiser leur structure complète.

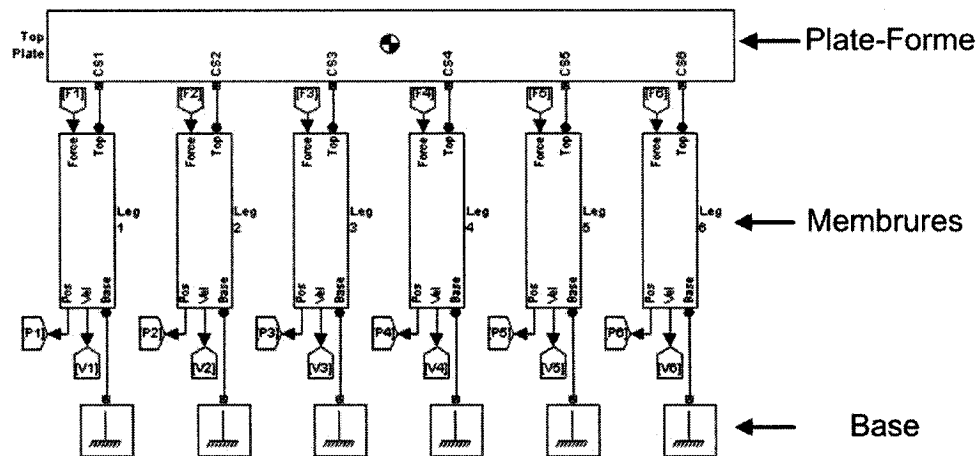


FIG. 4.1 Structure de la PGS (Smith et Wendlandt 2002)

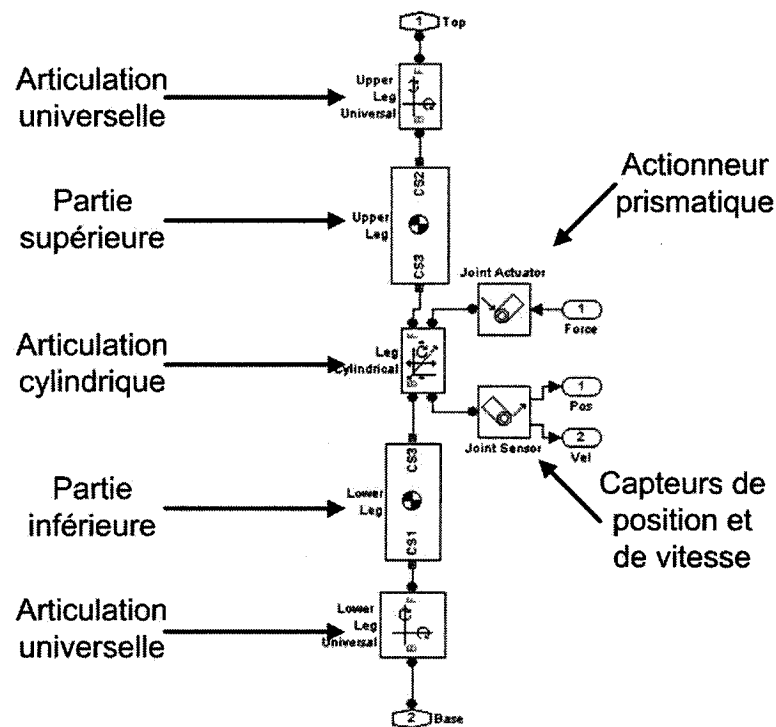
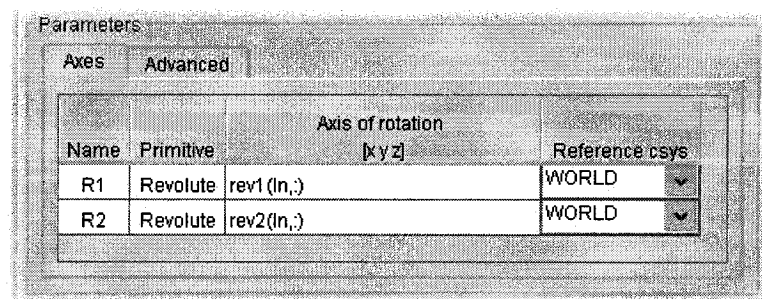


FIG. 4.2 Structure d'une membrure (Smith et Wendlandt 2002)

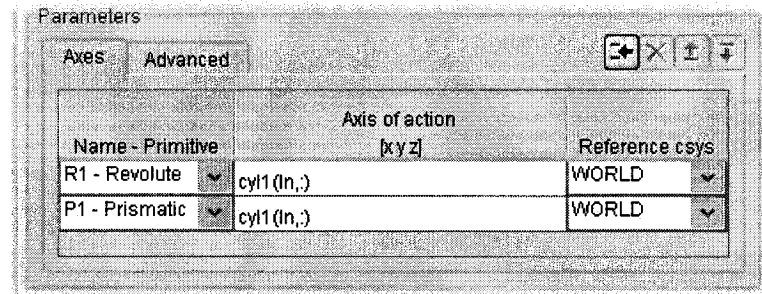
À la figure 4.2, il y a les trois articulations, entre coupées par deux corps. On voit un actionneur ainsi que deux capteurs, tous deux associés à la translation du joint prismatique. La façon dont sont définies les articulations dépend du type d'articulation et des DDL associés à cette même articulation.

Par exemple, pour une articulation universelle, il y a deux possibilités de rotation à déterminer. Pour définir ces axes de rotation (Simulink 2002), le menu illustré à la figure 4.4 permet de spécifier ces deux rotations. D'abord, on définit deux rotations (R1 et R2), puis on fixe les axes de rotations utilisés (rev1 et rev2).

À noter que plutôt que de se servir d'une articulation prismatique et une autre sphérique, le modèle de Smith (Smith et Wendlandt 2002) utilise une articulation cylindrique pour la jonction entre les deux sous-parties de membrures, et une articulation universelle à la jonction des membrures et de la plate-forme. Bien que ce modèle diffère de ceux modélisés généralement, il est plus précis que la plupart de ceux qu'on rencontre dans la littérature (Nadimi et al. 2006). Ce joint est défini par une translation P1 et une rotation R1, par rapport à l'axe longitudinal d'une membrure. Ces paramètres sont inscrits à la figure 4.3.



(a) Articulation universelle

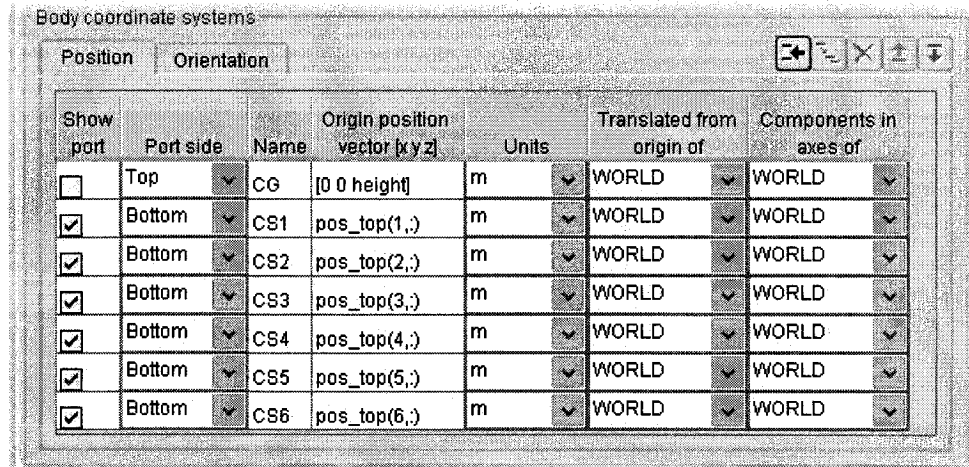


(b) Articulation cylindrique

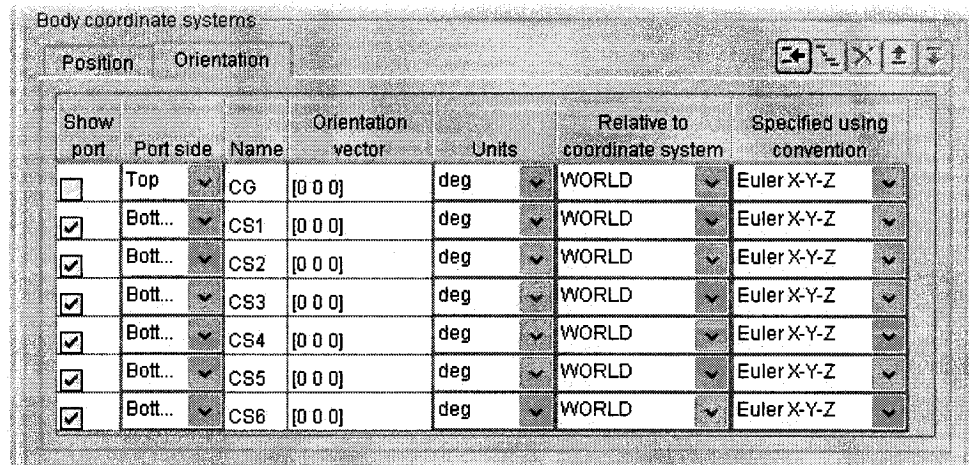
FIG. 4.3 Paramétrisation des articulations

Dans la modélisation, les paramètres relatifs aux différents corps sont également spécifiés. Pour ce faire, des modules permettent de définir spécifiquement toutes les caractéristiques des corps du système. D'abord, des repères sont utilisés pour positionner différents points importants d'un corps.

Par exemple, pour la plate-forme, il faut savoir où se situe le centre de masse ainsi que les 6 points d'attache avec les membrures. La figure 4.4 montre la définition du repère au centre de masse (CG), ainsi que les 6 repères solidaires aux points inférieurs des articulations (CS1 à CS6). La position de ces repères est fixée par des coordonnées de position et d'orientation. Relative à la base de la PGS, il y a les 3 coordonnées de position, ainsi que les 3 coordonnées d'orientation définies par la convention d'Euler.



(a) Position d'un corps



(b) Orientation d'un corps

FIG. 4.4 Repères et coordonnées des corps

Pour définir les repères relatifs aux membrures, la façon de faire est équivalente à ce qui est illustré à la figure 4.4. Cependant, puisque plusieurs repères sont nécessaires, il est important de bien expliquer la procédure. La membrure est séparée en deux parties telles qu'illustrées à la figure 4.5. Chacune des membrures possède un repère solidaire aux centres de masse (CG). De plus, chacune des sous-membrures possède 2 repères (CS1 et CS2) à leurs extrémités. Finalement, un troisième repère

(CS3) solidaire pour les deux membrures, permet éventuellement d'actionner ces membrures et de recueillir de l'information par les capteurs.

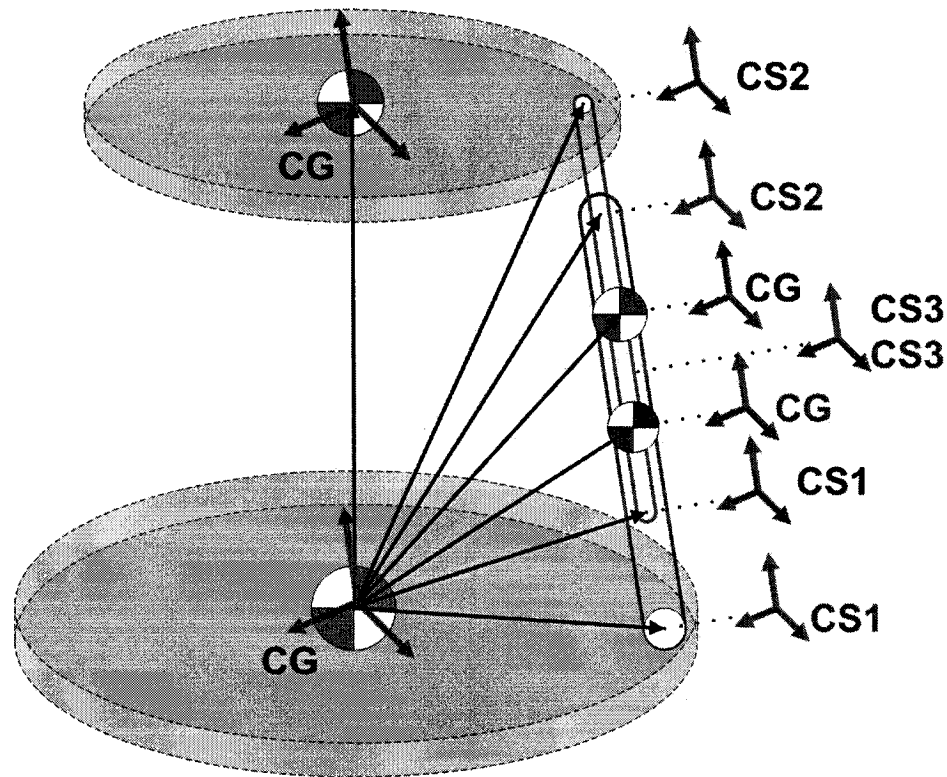


FIG. 4.5 Vecteur de positionnement des membrures

4.2.2 Principaux paramètres

Suite à la définition de la structure complète du modèle de simulation, il faut déterminer les autres paramètres essentiels pour la simulation du comportement de la PGS. Pour ce faire, on se sert des mêmes paramètres utilisés dans (Smith et Wendlandt 2002). Toutes les valeurs des paramètres sont dans le système d'unités international (SI).

D'abord, les positions de chacun des points d'attache des articulations sur la base et la plate-forme doivent être déterminées. Dans le repère local, ces positions sont inscrites dans le tableau 4.1. Les différents paramètres sont précisés au tableau 4.2.

TAB. 4.1 Position des articulations de la PGS

	Positions des articulations sur la plate-forme					
	$(p_1)_P$	$(p_2)_P$	$(p_3)_P$	$(p_4)_P$	$(p_5)_P$	$(p_6)_P$
x	0.6428	0.6428	0.3420	-0.9848	-0.9848	0.3420
y	-0.7660	0.7660	0.9397	0.1736	-0.1736	-0.9397
z	0	0	0	0	0	0
	Positions des articulations sur la base					
	$(b_1)_B$	$(b_2)_B$	$(b_3)_B$	$(b_4)_B$	$(b_5)_B$	$(b_6)_B$
x	2.9971	2.9971	-1.3852	-1.6119	-1.6119	-1.3852
y	-0.1309	0.1309	2.6610	2.5302	-2.5302	-2.6610
z	0	0	0	0	0	0

TAB. 4.2 Paramètres massiques de la PGS

Paramètres	Valeurs
Masse de la plate-forme	1216.9
Inertie de la plate-forme (I_x I_y I_z)	(304.5, 304.5, 608.5)
Masse inférieure des membrures	92.1107
Inertie inférieure des membrures (I_x I_y I_z)	(43.0, 43.0, 0.16)
Masse supérieure des membrures	51.8123
Inertie supérieure des membrures (I_x I_y I_z)	(24.2, 24.2, 0.23)

4.3 Premier contact avec la PGS

Avant d'appliquer la procédure de design proposée au modèle de la figure 4.1, essayons de se familiariser d'abord avec ce modèle à l'aide de tests simples. On introduit quelques notions de base impliquées dans la commande de la PGS, afin d'utiliser le modèle et d'explorer par des simulations le comportement et les particularités propres à la PGS modélisée.

4.3.1 Commande dans l'espace de travail

Bien que l'objectif est de commander la PGS dans l'espace de travail, c'est-à-dire d'amener la PGS à une valeur du vecteur q désirée, les actionneurs et les capteurs agissent sur les articulations cylindriques. Telles que définies à la section 3.3.2, les 6 élongations des joints prismatiques représentent l'espace articulaire du système. Il faut alors utiliser les notions de cinématique inverse introduites au chapitre 2, pour ramener le vecteur q désiré en terme d'élongations L .

Une fois les élongations résultant de la transformation du vecteur q trouvées, il est possible de commander les états L et $\dot{L}$ avec notre contrôleur pour agir sur les élongations par les forces des actionneurs linéaires F , dans le but d'atteindre la position q désirée.

4.3.2 Linéarisation

Pour utiliser le modèle TS tel qu'il est défini, il faut être en mesure de linéariser le modèle de la PGS. Au chapitre 3, nous avons déterminé que le système de la PGS

était du type

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

Les modèles linéarisés peuvent donc être représentés par l'équation dynamique suivante :

$$\begin{aligned}\dot{x}(t) &= A_i x(t) + B_i u(t) + \mu_i, \\ \dot{y}(t) &= C_i x(t).\end{aligned}$$

Puisque C_i est l'identité $\mathbb{I}_{12 \times 12}$ pour tout i . Elle ne sera pas considérée dans l'élaboration des modèles linéaires. Le contrôleur associé à un tel système est, quant à lui, de la forme de la relation (2.28), soit

$$u(t) = \{-K_i x(t) - \sigma_i\}.$$

Puisque le modèle est défini dans l'espace articulaire, on ramène les positions des points de linéarisation dans l'espace articulaire à l'aide des relations (3.8) modifiées, soit

$$L_i = \|S_i\| - L_0, \quad (4.1)$$

où L_0 est l'élongation initiale pour la position centrale de la plate-forme.

En effet, le modèle de simulation ramène les longueurs des membrures en élongations relatives par rapport à une élongation nulle pour la position centrale de plate-forme $q = [0 \ 0 \ 2 \ 0 \ 0 \ 0]^T$. Par exemple, la linéarisation, effectuée direc-

tement par les fonctions *linmod* de MATLAB à ce point, où toutes les élongations $L_i = 0$, nous donne les matrices suivantes :

$$A_1 = \begin{bmatrix} 0_{6 \times 6} & \mathbb{I}_{6 \times 6} \\ a_{21} & 0_{6 \times 6} \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0_{6 \times 6} \\ b_{21} \end{bmatrix},$$

où les éléments $(a_{21})_1$ et $(b_{21})_1$, qui dépendent du point de linéarisation sont donnés par

$$(a_{21})_1 = \begin{pmatrix} 9.0418 & -4.7826 & 1.4778 & -2.8144 & -2.0513 & 2.4682 \\ -4.7826 & 9.0418 & 2.4682 & -2.0513 & -2.8144 & 1.4778 \\ -2.0513 & 2.4682 & 9.0418 & -4.7826 & 1.4778 & -2.8144 \\ -2.8144 & 1.4778 & -4.7826 & 9.0418 & 2.4682 & -2.0513 \\ 1.4778 & -2.8144 & -2.0513 & 2.4682 & 9.0418 & -4.7826 \\ 2.4682 & -2.0513 & -2.8144 & 1.4778 & -4.7826 & 9.0418 \end{pmatrix},$$

$$(b_{21})_1 = \begin{pmatrix} 0.0002 & 0.0002 & -0.0010 & 0.0002 & -0.0001 & 0.0021 \\ 0.0002 & -0.0010 & 0.0002 & 0.0002 & 0.0021 & -0.0001 \\ -0.0010 & 0.0002 & -0.0001 & 0.0021 & 0.0002 & 0.0002 \\ 0.0002 & 0.0002 & 0.0021 & -0.0001 & 0.0002 & -0.0010 \\ -0.0001 & 0.0021 & 0.0002 & 0.0002 & -0.0010 & 0.0002 \\ 0.0021 & -0.0001 & 0.0002 & -0.0010 & 0.0002 & 0.0002 \end{pmatrix}.$$

Les matrices B_i ont une forme que l'on peut modifier afin de permettre l'analyse du système affine. Par la transformation non singulière (Kim et Lee 2005) suivante

$$z = \Phi x,$$

il est possible de modifier les matrices précédentes afin d'utiliser les conditions du

théorème 2.9.1. La matrice de transformation est donnée par

$$\Phi = \begin{pmatrix} \Sigma \\ (B^T B)^{-1} B^T \end{pmatrix},$$

où Σ est une matrice composée de lignes mutuellement indépendantes et perpendiculaires aux colonnes de B . Un choix simple de transformation est le suivant :

$$\Phi = \begin{bmatrix} \mathbb{I}_{6 \times 6} & 0_{6 \times 6} \\ 0_{6 \times 6} & (B^T B)^{-1} B^T \end{bmatrix}.$$

Et le système est transformé en un système équivalent où

$$\bar{A}_i = \Phi A_i \Phi^{-1},$$

$$\bar{B}_i = \Phi B_i,$$

$$\bar{\mu}_i = \Phi \mu_i.$$

Pour notre système, les matrices B_i se ramènent à

$$\bar{B}_i = B = \begin{pmatrix} 0_{6 \times 6} \\ \mathbb{I}_{6 \times 6} \end{pmatrix}.$$

Donc, la transformation rend la forme de B beaucoup plus intéressante. Il faut également prendre soin, lors de la synthèse de contrôleur avec le système équivalent, de ramener les gains du contrôleur $\bar{K}_i$ à des gains K_i permettant de commander le système d'origine. La façon de faire est d'utiliser la matrice de transformation de telle sorte que

$$K_i = \bar{K}_i \Phi.$$

où $\bar{K}_i$ est la matrice des gains obtenus avec les matrices transformées ; K_i est celle du système réel à commander.

4.3.3 Tests simples

L'obtention des matrices du système dynamique linéarisé permet d'effectuer quelques tests importants dans la compréhension du comportement de la PGS. En effet, l'objectif visé est d'utiliser un contrôleur par retour d'état. Il faut s'assurer que ce type de contrôleur est bien adapté à l'application de commande souhaitée. Aussi, ces quelques tests aident à l'implantation du contrôleur le plus adéquat possible. Tous ces tests sont effectués pour un seul modèle linéaire, décrit par les matrices introduites à la section précédente.

4.3.3.1 Stabilité en boucle ouverte

Afin de débiter l'analyse de la PGS avec la méthodologie introduite, la stabilité d'un modèle linéaire de la PGS est analysée grâce au théorème 2.5. En effet, les valeurs propres de la matrice A_1 , inscrites au tableau 4.3, ne sont pas uniquement à parties réelles positives. Il est donc impossible de trouver une matrice P symétrique et définie positive respectant l'inégalité de Lyapunov. Ceci implique que certains pôles du système se retrouvent dans le plan complexe de droite, et que ces pôles sont instables.

TAB. 4.3 Partie réelle des valeurs propres de la matrice A_1

3.6874	-3.6874	1.8274	3.8806	3.8806	1.8969
1.8969	-1.8274	-3.8806	-3.8806	-1.8969	-1.8969

Il faut faire attention cependant de ne pas généraliser ce résultat à l'ensemble des positions de la PGS, mais en ce point, le modèle linéaire n'est pas un point stable. Comme il fallait s'y attendre, la PGS présente un caractère d'instabilité; il faut donc la munir d'un contrôleur afin de la commander.

4.3.3.2 Stabilité en boucle fermée

De la même manière que pour le test de stabilité, nous utilisons un seul modèle pour trouver les gains du contrôleur. À l'aide du théorème 2.9.1, on trouve des gains pour le contrôleur par retour d'état avec constante ajoutée.

On voit que la plate-forme maintient sa position d'équilibre en $q = [0 \ 0 \ 2 \ 0 \ 0 \ 0]^T$, tout en ayant permis de l'atteindre pour une position initiale de $q_0 = [-0.05 \ 0 \ 2 \ 0 \ 0 \ 0]^T$.

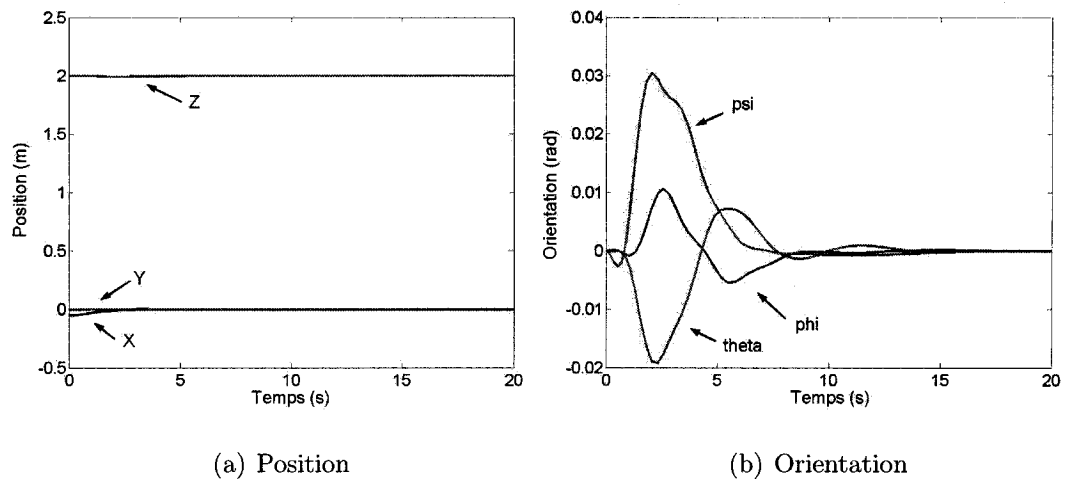


FIG. 4.6 Stabilité du point $L_i = 0$

Le contrôleur par retour d'état avec constante élaboré dans cette section permet la stabilisation de la plate-forme au point de linéarisation choisie. De plus, il serait

bien d'étendre la stabilité de la PGS pour une plus grande portion de son espace de travail, par une modélisation plus représentative que par un seul modèle linéaire.

4.4 Approximation par un modèle TS

Une approche par la logique floue est utilisée pour la modélisation représentative de la PGS, afin de la commander en position. On débute par une application à un nombre réduit de règles floues. Les trois étapes de représentation floue sont implicitement détaillés dans l'élaboration du modèle.

Pour la fuzzyfication, il faut projeter les variables relatives à notre système sur des sous-ensembles flous. Les variables réelles de notre système sont la position et l'orientation, ainsi que les différentes vitesses des composantes du système. En tout, le modèle de la PGS possède 12 variables d'état.

Pour réussir à avoir un modèle qui garantit la représentation floue exacte de notre système non linéaire, il faudrait utiliser la méthode de non linéarité par section (Tanaka et Wang 2001). Cette méthode prend en considération que toutes les variables des phénomènes physiques sont bornées. Le modèle est ensuite construit en considérant les valeurs de ces non linéarités. Bien que cette méthode soit très rigoureuse, il est parfois très difficile de l'appliquer. Pour la PGS, l'application de cette approche dans l'élaboration du modèle est très laborieuse. D'un côté, les nombreuses non linéarités sont très difficiles à extraire et à traduire sous forme de sous-ensembles flous. De l'autre, puisque le modèle de la PGS retenu possède 12 variables d'états, il faudrait construire le modèle floue en combinant des règles sur ces 12 variables. Ceci conduirait à un modèle flou au nombre de règles très grand.

Pour cette raison, la méthode exacte de fuzzyfication n'est pas retenue.

Afin de réduire le nombre de règles du système, la méthode d'approximation locale en décomposition floue de l'espace est utilisée (Tanaka et Wang 2001). La philosophie de cette méthode permet de faire l'approximation des non linéarités du système par un choix judicieux des modèles linéaires. Comme avantage certain, cette méthode conduit à une résolution simplifiée des conditions sous forme de LMI utilisées pour l'analyse et le design. On se sert de l'information recueillie sur le comportement du modèle, dans la section 4.3, afin de procéder adéquatement au design de contrôleur par le modèle TS.

4.4.1 Construction d'un modèle à 5 règles

Comme première analyse, le positionnement dans le plan XY est souhaité. La plateforme est donc commandée selon les axes X et Y. Ceci équivaut à un déplacement horizontal de la PGS, en maintenant une orientation constante. Pour utiliser la méthode approximative, nous utilisons donc uniquement les position x et y comme prémisses.

Les fonctions dont on se sert pour représenter les sous-ensembles flous sont illustrées à la figure 4.7. Ce choix est déterminé par l'approche floue utilisée. En effet, admettons que l'on désire transposer les valeurs de x et y en variables linguistiques pour des valeurs comprises entre -1 et 1, tout en ayant un nombre réduit de règles. En respectant la méthode de décomposition de l'espace, ce choix est fait. La transition entre deux valeurs de prémisses se veut progressive par l'utilisation de support très large. Il est toujours possible de modifier ces fonctions si jamais elle

ne permettent pas une modélisation assez représentative de la PGS. On peut aussi restreindre l'espace à décomposer si jamais elle est trop grande pour le nombre de règles impliquées. La conclusion résultant des conditions est une approximation locale linéaire du système réel.

Ces règles s'énoncent ainsi

SI $x \in M_{i1}$ ET $y \in M_{i2}$ ALORS Modèle linéaire i .

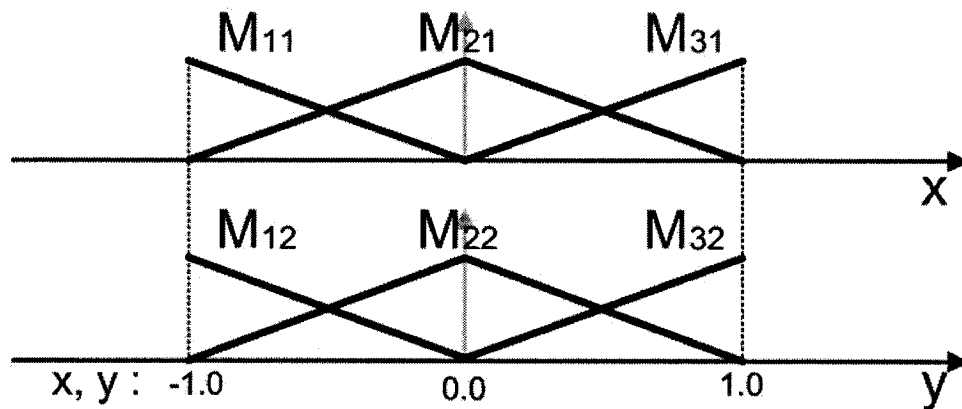


FIG. 4.7 Sous-ensembles flous pour le modèle à 5 règles

Pour établir les règles du système flou, nous utilisons les variables linguistiques résultant de la fuzzification illustrée à la figure 4.7. Les conditions sont établies selon l'appartenance des prémisses à l'un ou l'autre des sous-ensembles flous. Et pour chacune de ces prémisses, il y a trois valeurs où la fonction d'appartenance est maximale. Il y a donc une possibilité d'utiliser 9 points de linéarisation, tels qu'illustrés à la figure 4.8(a). Le plan XY est ainsi décomposé en 4 régions floues, qu'il est possible de représenter par 4 points chacune. Mais plus il y a de points impliqués dans la représentation d'une région floue, plus il y a de LMI à résoudre et plus le contrôleur est complexe (Sun et al. 2004).

Le modèle est donc linéarisé selon cinq points dans le plan XY. La représentation des 4 régions se fait par trois points uniquement. Pour construire le modèle flou sur les bases de ce qui a été fait à la section 4.3, on ajoute 4 points tout autour de celui en $q = [0 \ 0 \ 2 \ 0 \ 0 \ 0]^T$.

Les points de linéarisation sont illustrés à la figure 4.8(b) ainsi qu'à la figure 4.9 de façon plus schématique. Les valeurs de ces élongations, calculés à l'aide de la relation (4.1) sont inscrites au tableau 4.4.

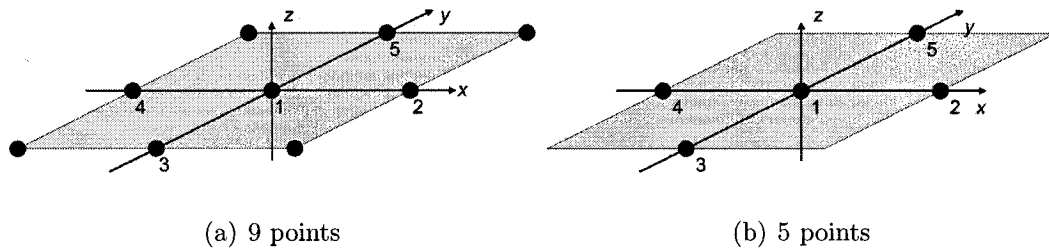


FIG. 4.8 Points de linéarisation possibles pour le modèle à 5 règles

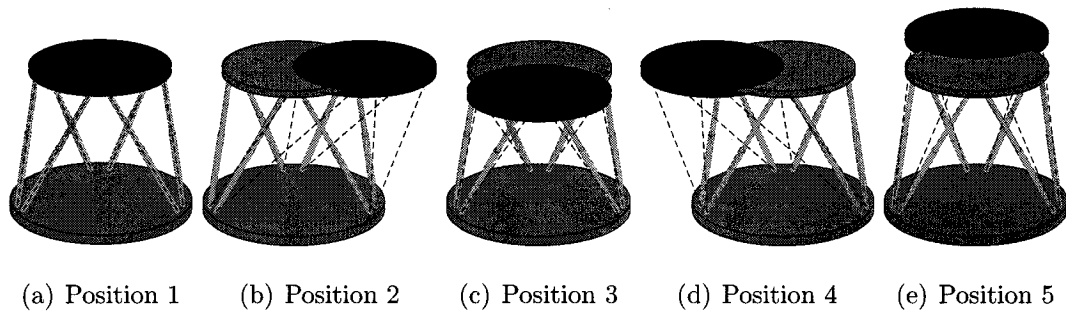


FIG. 4.9 Linéarisation aux 5 points schématiques

TAB. 4.4 Élongations L_i aux points (1 à 5) de linéarisation

i	1	2	3	4	5
L_1	0.0000	-0.6563	0.3415	0.8029	-0.04316
L_2	0.0000	-0.6563	-0.04316	0.8029	0.3415
L_3	0.0000	0.6411	0.6395	-0.4167	-0.4145
L_4	0.0000	0.3391	0.8034	-0.04056	-0.6571
L_5	0.0000	0.3391	-0.6571	-0.04056	0.8034
L_6	0.0000	0.6411	-0.4145	-0.4167	0.6395

On peut faire une approximation du modèle (3.22) avec un modèle TS à cinq règles de type SI-ALORS. Pour simplifier l'écriture, à noter que même si la transformation est effectuée, nous conservons la notation d'origine pour exprimer les différents éléments du modèle. Les règles du modèle sont :

Règle 1 :

SI $q_1(t)$ est M_{21} et $q_2(t)$ est M_{22} , ALORS $\dot{x}(t) = A_1x(t) + B_1u(t) + \mu_1$;

Règle 2 :

SI $q_1(t)$ est M_{31} et $q_2(t)$ est M_{22} , ALORS $\dot{x}(t) = A_2x(t) + B_2u(t) + \mu_2$;

Règle 3 :

SI $q_1(t)$ est M_{21} et $q_2(t)$ est M_{12} , ALORS $\dot{x}(t) = A_3x(t) + B_3u(t) + \mu_3$;

Règle 4 :

SI $q_1(t)$ est M_{11} et $q_2(t)$ est M_{22} , ALORS $\dot{x}(t) = A_4x(t) + B_4u(t) + \mu_4$;

Règle 5 :

SI $q_1(t)$ est M_{21} et $q_2(t)$ est M_{32} , ALORS $\dot{x}(t) = A_5x(t) + B_5u(t) + \mu_5$;

Toutes les matrices A_i et B_i apparaissent en annexe.

Le contrôleur partage les mêmes règles que le modèle flou comme suit :

Règle i : SI x est M_{i1} et y est M_{j2}

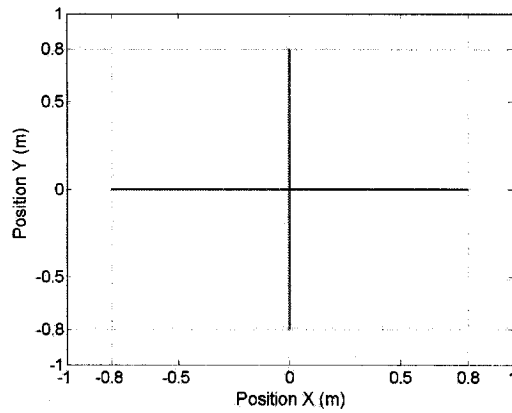
$$ALORS \ u(t) = -K_i x(t) - \sigma_i.$$

où K_i et σ_i sont les valeurs qu'il reste à déterminer. La façon de procéder est l'objet de la prochaine section.

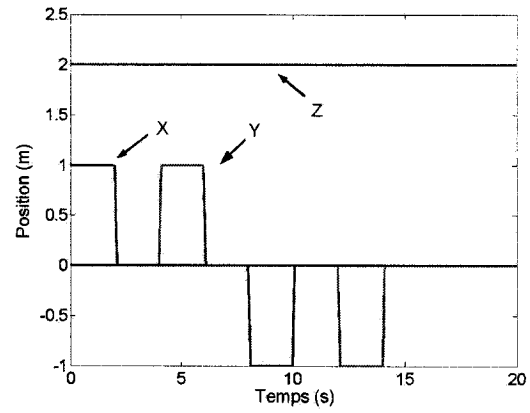
4.4.2 Résultats préliminaires

Une fois le modèle flou à 5 règles obtenu, on procède à la résolution d'algorithmes afin d'analyser et de synthétiser différents contrôleurs. L'objectif de cette section est donc de tester les théorèmes appropriés, énoncés au chapitre 2, sur la PGS. Il est alors possible de tester la méthode proposée. Les premières applications choisies sont le suivi d'une croix et le suivi d'un cercle dans le plan XY. Les trajectoires sont illustrées à la figure 4.10.

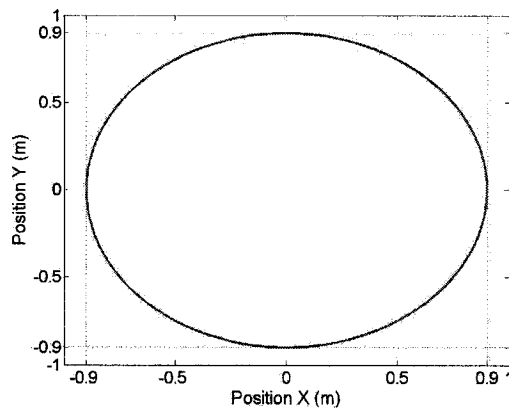
La croix permet de vérifier si le passage d'un point de linéarisation à un autre est possible. Aussi, on vérifie le comportement stationnaire du système en maintenant la PGS quelque peu en différentes positions. Le cercle, quant à lui, permet de vérifier le comportement aux points éloignés dans le plan des points de linéarisation.



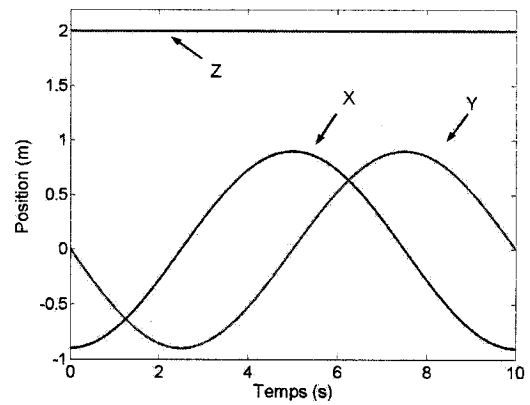
(a) Croix : plan xy



(b) Croix : Référence vs temps



(c) Cercle : plan xy



(d) Cercle : Référence vs temps

FIG. 4.10 Trajectoires dans le plan XY

4.4.2.1 Système affine

Pour une condition d'opération stationnaire au point (x_r, u_r) , on pourrait s'attendre à avoir l'équation dynamique $A_i x_r + B_i u_r = 0$ respectée. Cependant, ce calcul est inexact, pour un système affine de type $x(t) = f(x(t)) + g(x, u)$, lorsqu'il est effectué. On voit apparaître des résidus, de la même forme que le système affine (2.27). Ainsi,

par la relation

$$A_i x_r + B_i u_r + \mu_i = 0,$$

nous obtenons la valeur du résidu μ_i . Le tableau 4.5 nous donne les résidus pour les 5 points de linéarisation.

TAB. 4.5 Résidus de linéarisation

Membrure associée	μ_1	μ_2	μ_3	μ_4	μ_5
L_1	6.6264	7.3631	6.2397	5.8072	6.5864
L_2	6.6264	7.1459	6.6360	5.9775	6.1250
L_3	6.6264	6.6451	6.2946	6.6785	6.3462
L_4	6.6264	6.6892	6.2803	6.6908	7.1769
L_5	6.6264	6.5525	6.9641	6.8850	6.3657
L_6	6.6264	5.8452	7.1176	7.3539	6.7114

Pour tenir compte de ces résidus, la formulation en un système affine flou est utilisée. Cependant, la transformation qu'elle impose pour les résidus (Kim et Lee 2005) fait ressortir de l'information importante. La transformation suivante :

$$\bar{\mu} = \Phi \mu,$$

fait ressortir un $\bar{\mu}$ tout simplement équivalent à la transmission des forces statiques des actionneurs pour compenser les forces de gravité, soit u_{ri} . Donc, lors de la synthèse d'un contrôleur du type

$$u = \sum_{i=1}^r h_i(z(t)) \{-K_i x - u_{ri}\},$$

la constante ajoutée permet de prendre en considération le poids de la plate-forme et des parties supérieures de membrures et de commander le système suivant :

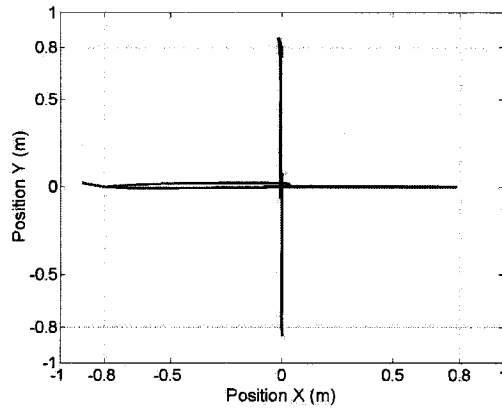
$$\dot{x}(t) = \sum_{i=1}^r h_i(z(t)) \{A_i - B_i K_i\} x(t) + (\bar{\mu} - B_i u_{ri})\}.$$

Formellement, on se retrouve donc avec des conditions de synthèse qui prennent en considération la constante de compensation de la gravité de notre système. En effet, la comportement statique du système (3.22) au point d'équilibre nous donne

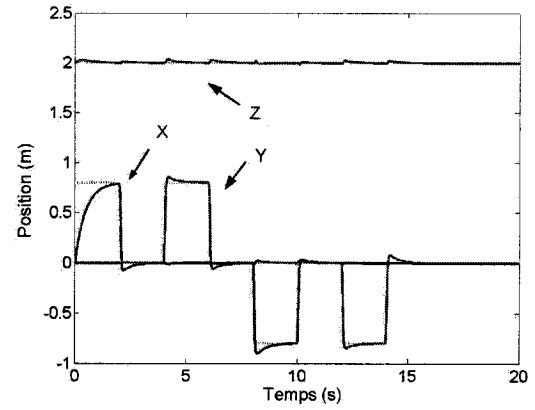
$$J^T G(q_i) = -F_i = -B_i u_{ri}$$

où les u_{ri} sont données en annexe.

Lors de la simulation, la PGS exhibe le comportement illustré à la figure 4.11.



(a) Suivi de la croix



(b) Position x, y, z

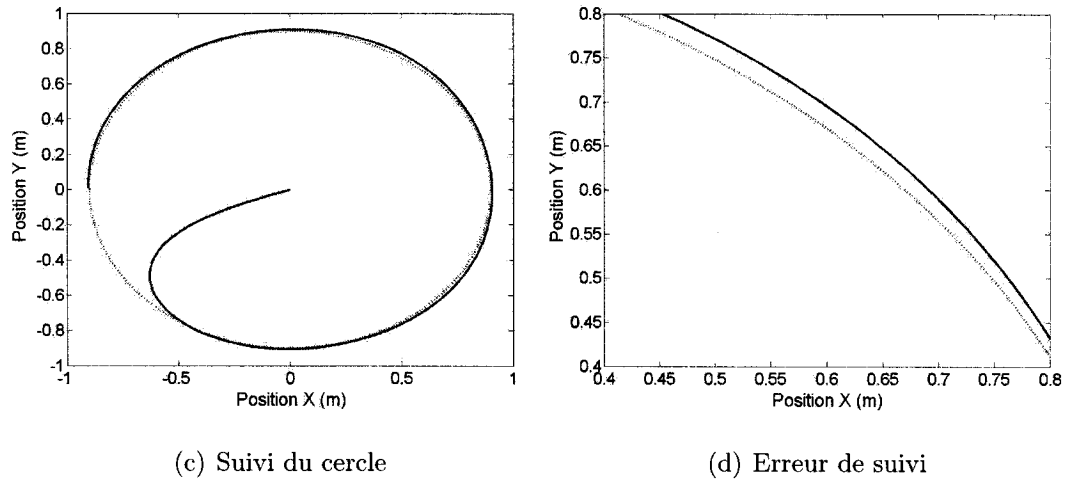


FIG. 4.11 Suivi de trajectoire : Conditions affines

On voit que la commande en position est fonctionnelle. Cependant, pour le suivi de la croix, on voit apparaître un certain dépassement des valeurs de référence (Fig.4.11(b)), ainsi qu'une certaine variation en z lors des accélérations de la PGS. Pour le suivi de la croix, encore une fois, on voit que la PGS n'a pas de difficulté à rejoindre la trajectoire (Fig.4.11(c)). Cependant, la figure 4.11(d) montre une erreur importante entre la trajectoire et le parcours réel. Puisqu'une constante est ajoutée, et qu'elle est une combinaison des différentes règles impliquées dans la commande, l'erreur ne s'annule qu'en ces points de linéarisation. En effet, lors du suivi du cercle, cette erreur diminue en se rapprochant des points de linéarisation.

Pour diminuer ces erreurs, on a deux choix possibles si on veut continuer à utiliser les conditions du théorème 2.9.1 uniquement. Premièrement, il faudrait utiliser des fonctions d'appartenance des sous-ensembles flous en considérant les relations non linéaires intervenant dans la compensation de la gravité. On aurait alors une défuzzification qui donnerait la valeur exacte de constante au point de référence. Deuxièmement, on pourrait tout simplement augmenter le nombre de modèles, pour

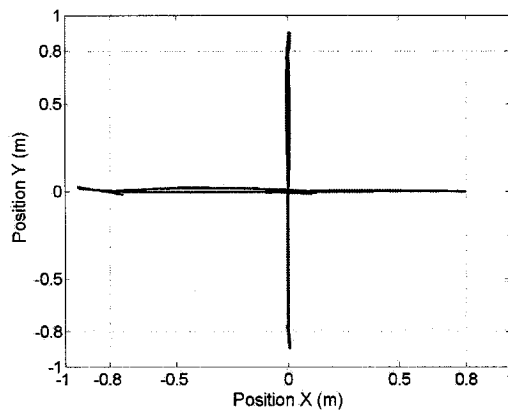
que la référence se rapproche d'un point de linéarisation et que l'erreur s'en trouve diminuée.

Cependant, ces deux choix demandent une analyse poussée pour l'une, et une synthèse de contrôleur longue pour l'autre.

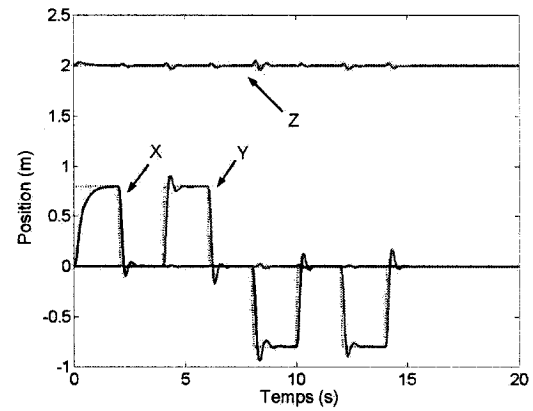
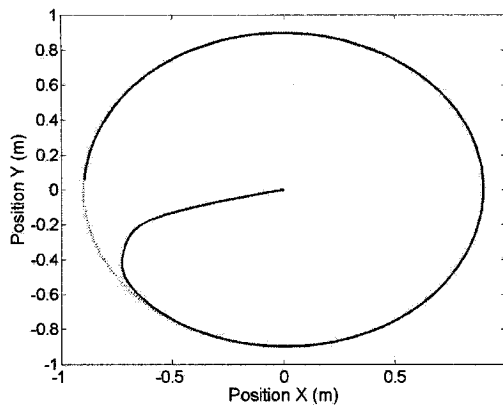
4.4.2.2 Ajout de l'intégrateur

Afin de réduire et peut-être éliminer complètement les erreurs de position, sans compliquer outre mesure le design du correcteur, on introduit un intégrateur dans le contrôleur (Korba et al. 2003). La résolution, avec les matrices augmentées, permet de trouver les gains du contrôleur flou.

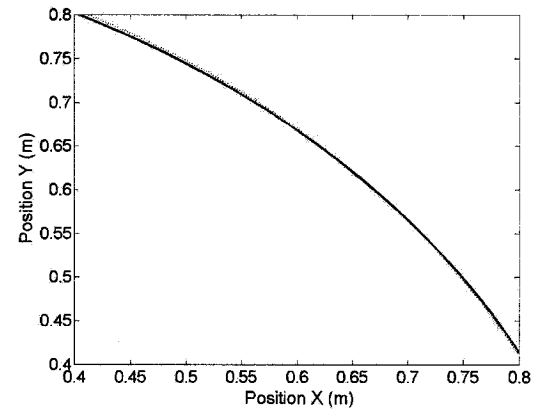
Appliqué à la PGS, le contrôleur donne les résultats illustrés à la figure 4.12. On y voit la stabilisation en différentes positions de la croix (Fig. 4.12(a)) tout en maintenant une erreur statique nulle (Fig. 4.12(b)). Et ce, même si on se trouve loin des points de linéarisation. En effet, les figures 4.12(c) et 4.12(d) montrent que l'erreur s'annule, pour des valeurs éloignées de la référence, lors du suivi du cercle.



(a) Suivi de la croix

(b) Position x, y, z 

(c) Suivi du cercle



(d) Erreur de suivi

FIG. 4.12 Suivi de trajectoire : Ajout d'un intégrateur

Cependant, l'ajout de l'intégrateur fait apparaître un autre problème. En effet, en regardant les figures 4.12(b), on voit qu'avant d'atteindre la position d'équilibre, il y a un dépassement important et une oscillation autour de cette valeur de référence. Pour certaines applications, ce dépassement n'est pas envisageable. Il faut donc penser à le diminuer, ou l'éliminer si possible, par un meilleur choix des pôles du système.

4.4.2.3 Placements de pôles

Pour augmenter la rapidité de notre système, il est possible d'utiliser les conditions de la section 2.7. Cependant, pour réduire le dépassement, il faut augmenter l'amortissement du système. Pour ce faire, en ramenant les pôles du système près de l'axe réel, la region- $\mathcal{D}$ du cône est utilisée. Aussi, pour l'augmentation de la rapidité, cette région est combinée avec le region- $\mathcal{D}$ du demi-plan complexe. Les résultats obtenus, illustrés à la figure 4.13 montrent l'efficacité le comportement après le placement de pôles.

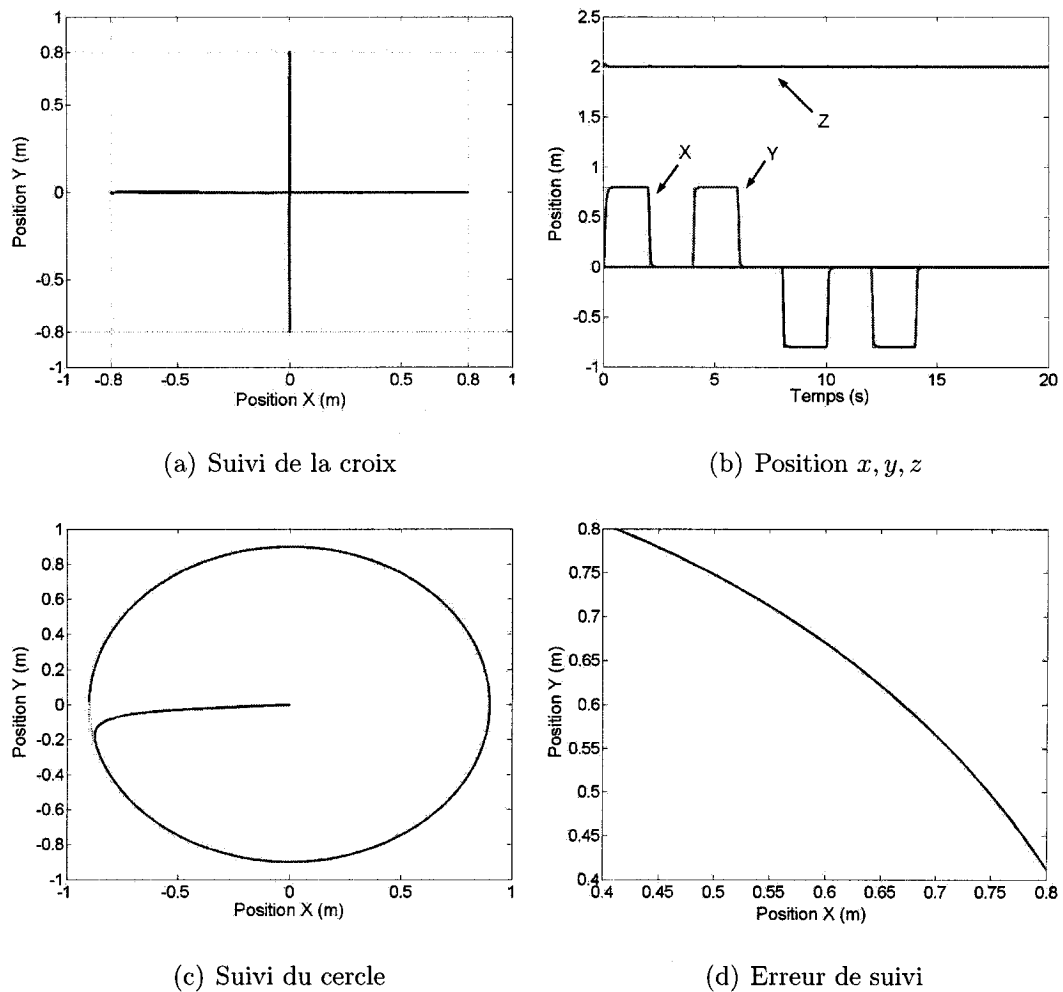


FIG. 4.13 Suivi de position : Intégrateur et placement de pôles

Les performances obtenues pour le suivi de position de la croix nous montrent des résultats améliorés (Fig. 4.13(a) et 4.13(b)). De plus, on remarque que le système est rapide, tout en maintenant un dépassement nul (Fig. 4.13(b)). Et ce, en tout point le long de l'axe des x et de y . Pour le suivi du cercle, on voit l'effet de l'augmentation de rapidité par la vitesse à laquelle la PGS atteint la trajectoire du cercle (Fig. 4.13(c)). Aussi, on voit que l'erreur de suivi de ce cercle est nulle Fig. (4.13(d)).

4.4.2.4 Autres trajectoires

On en arrive à commander la PGS pour des trajectoires selon l'axe des x et l'axe des y . En utilisant le même contrôleur, différentes trajectoires sont testées pour vérifier les limites de ce contrôleur à 5 règles.

Tout d'abord, on essaie un mouvement vertical et une stabilisation en différents points le long de l'axe des z . Aussi, une trajectoire en forme de triangle est poursuivie par la PGS afin de voir son comportement pour des points très éloignés des points de linéarisation. Cette trajectoire est illustrée à la figure 4.14.

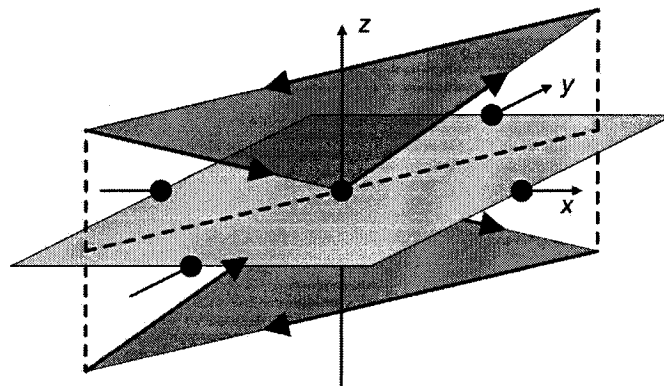
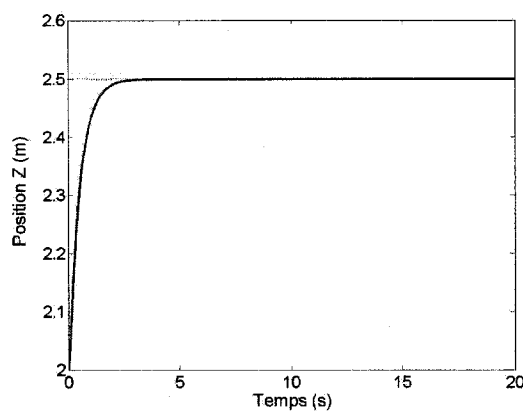
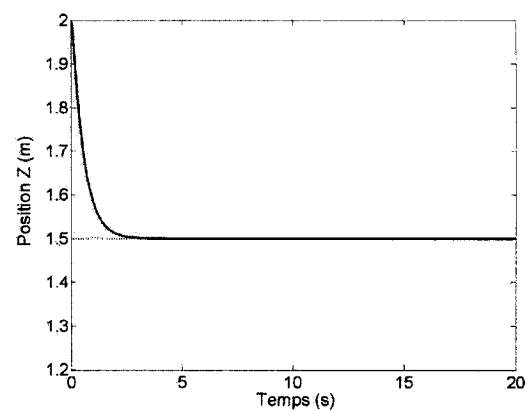


FIG. 4.14 Trajectoire triangulaire

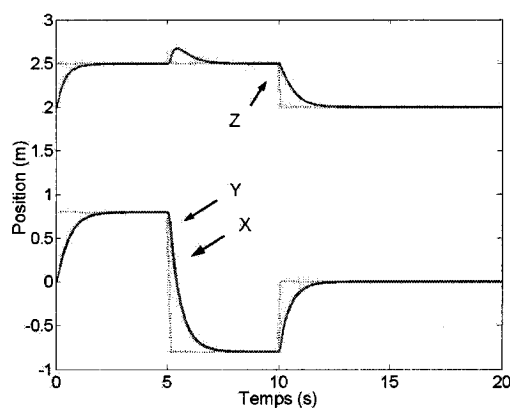
Pour le mouvement vertical, on remarque à la figure 4.15 et que le positionnement le long de l'axe des z de la PGS, pour le contrôleur synthétisé pour le plan XY, est possible. En effet, l'intégrateur compense très bien pour les erreurs d'approximation du modèle à ces points, et ce aussi bien pour un mouvement ascendant (Fig. 4.15(a)) que descendant (Fig. 4.15(b)). Cependant, le suivi de la trajectoire triangulaire, montré à la figure (Fig. 4.15(d)) ne fonctionne pas aussi bien.



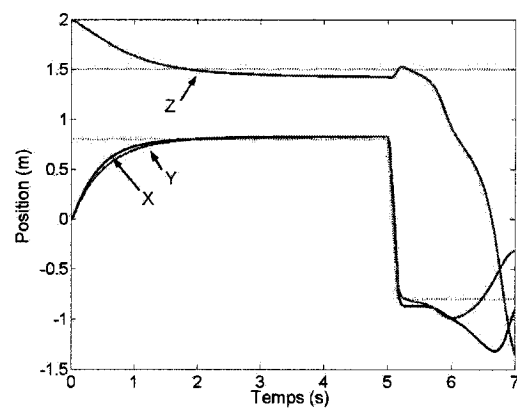
(a) Mouvement vertical ascendant



(b) Mouvement vertical descendant



(c) Suivi du triangle supérieur



(d) Suivi du triangle inférieur

FIG. 4.15 Suivi de position : Intégrateur et placement de pôles

En effet, pendant que le contrôleur est efficace pour la portion supérieure du suivi

(Fig. 4.15(c)), il ne l'est pas pour le suivi de la portion inférieure (Fig. 4.15(d)). En effet, lors du passage par le point le plus éloigné du suivi, la plate-forme décroche et tombe. Le fait est que pour compenser la gravité de la plate-forme, plus on descend, plus la demande en force des actuateurs est grande. En effet, l'angle entre les membrures et la base diminue, rendant la transmission des forces à la plate-forme moins efficace. Ainsi, lors de la compensation de gravité à ce point de décrochage, la constante de compensation, calculée par une combinaison floue des constantes au 5 points de linéarisation, n'est pas suffisante pour maintenir la PGS en position d'équilibre, malgré l'apport de l'intégrateur.

Il faut donc penser, pour généraliser davantage le mouvement, à modifier quelque peu le contrôleur.

4.5 Modèle à 15 règles

Afin de faire suite aux résultats obtenus dans l'étude préliminaire et lors de la synthèse à 5 règles, nous allons élargir la procédure pour la synthèse d'un contrôleur plus complet. Aux 5 points de linéarisation utilisés jusqu'ici, nous introduisons des nouvelles règles basées sur la position verticale de la plate-forme dans l'espace. Ainsi, nous avons une nouvelle prémisse, z , dans l'établissement de nos règles floues. De plus pour permettre des mouvements combinés selon plusieurs axes, nous ajoutons des points dispersés sur les 3 axes. Le but de ces ajouts est de mouvoir la PGS dans l'espace, en respectant les limites suivantes :

$$-1 < x < 1, \quad -1 < y < 1, \quad 1 < z < 3.$$

. De façon similaire à ce qui a été fait dans la procédure préliminaire, les trois prémisses retenues sont projetées sur les ensembles flous de la figure 4.16. Effectivement, on utilise plus de sous-ensembles flous que lors de l'étape précédente. Il ne devrait pas avoir de problème lié à l'utilisation de fonctions triangulaires de la figure 4.16. Si tel est le cas, il faudra penser à les modifier.

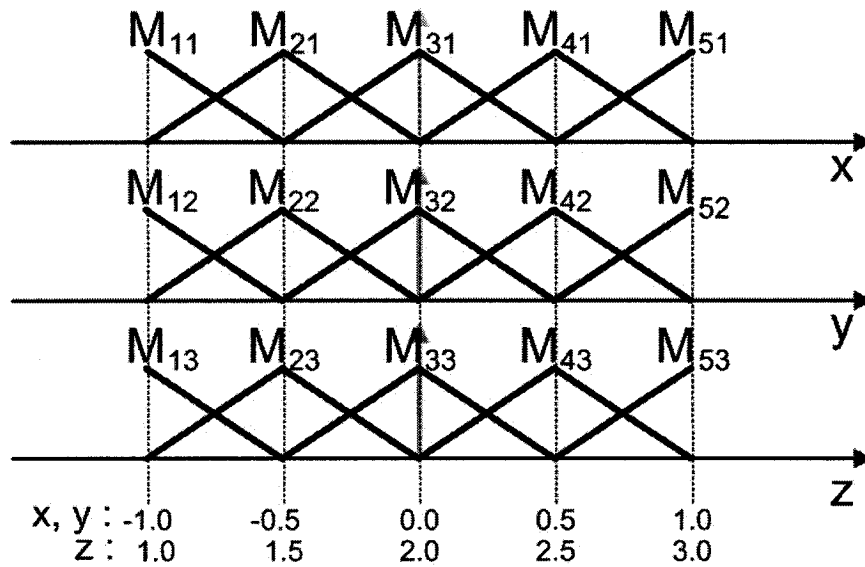


FIG. 4.16 Sous-ensembles flous pour le modèle à 15 règles

Pour établir les règles floues selon la méthode de décomposition de l'espace complète avec les sous-ensembles flous choisis, il faudrait utiliser les 125 points de linéarisation, illustrés à la figure 4.17(a), définissant 64 régions floues, chacune représentées par 8 points de linéarisation (Sun et al. 2004). Évidemment, la synthèse d'un contrôleur est très complexe si on ne réduit pas le nombre de règles.

15 points de linéarisation, pour un modèle à 15 règles, sont donc utilisés pour construire le modèle flou TS. En considérant les 5 points de linéarisation déjà introduit à l'étape préliminaire, 10 nouveaux points de linéarisation sont utilisés pour définir 15 régions floues. Ils sont représentés à la figure 4.17(b).

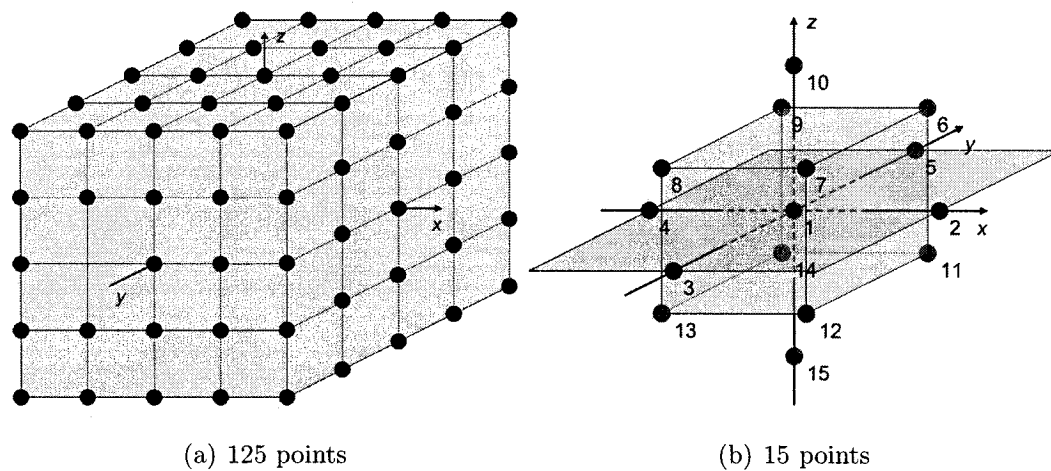


FIG. 4.17 Points de linéarisation possibles pour le modèle à 15 règles

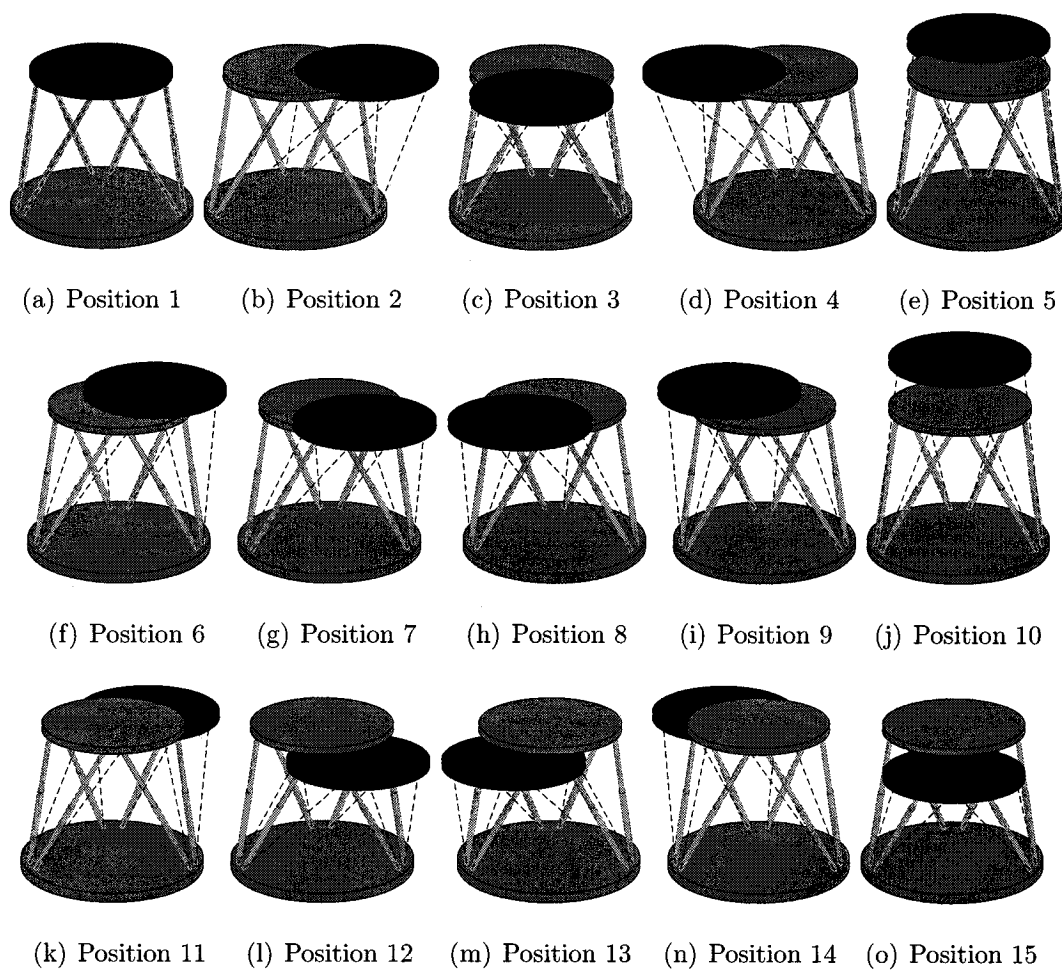


FIG. 4.18 Linéarisation aux 15 points schématiques

Le tableau 4.6 résume la projection des prémisses sur les sous-ensembles flous ainsi que sur l'ensemble des règles impliquées dans le modèle. Aussi, les points de linéarisation correspondant sont également inclus dans ce tableau.

TAB. 4.6 Règles flous pour le modèle à 15 règles

SI...			ALORS	
$x \in \dots$	$y \in \dots$	$z \in \dots$	le modèle est $i \mapsto$	Point de linéarisation (x, y, z)
M_{31}	M_{32}	M_{33}	$1 \mapsto$	$(0.0, 0.0, 2.0)$
M_{51}	M_{m2}	M_{m3}	$2 \mapsto$	$(1.0, 0.0, 2.0)$
M_{m1}	M_{12}	M_{m3}	$3 \mapsto$	$(0.0, -1.0, 2.0)$
M_{11}	M_{m2}	M_{m3}	$4 \mapsto$	$(-1.0, 0.0, 2.0)$
M_{m1}	M_{52}	M_{m3}	$5 \mapsto$	$(0.0, 1.0, 2.0)$
M_{41}	M_{42}	M_{43}	$6 \mapsto$	$(0.5, 0.5, 2.5)$
M_{41}	M_{22}	M_{43}	$7 \mapsto$	$(0.5, -0.5, 2.5)$
M_{21}	M_{22}	M_{43}	$8 \mapsto$	$(-0.5, -0.5, 2.5)$
M_{21}	M_{42}	M_{43}	$9 \mapsto$	$(-0.5, 0.5, 2.5)$
M_{m1}	M_{m2}	M_{53}	$10 \mapsto$	$(0.0, 0.0, 3.0)$
M_{41}	M_{42}	M_{23}	$11 \mapsto$	$(0.5, 0.5, 1.5)$
M_{41}	M_{22}	M_{23}	$12 \mapsto$	$(0.5, -0.5, 1.5)$
M_{21}	M_{22}	M_{23}	$13 \mapsto$	$(-0.5, -0.5, 1.5)$
M_{21}	M_{42}	M_{23}	$14 \mapsto$	$(-0.5, 0.5, 1.5)$
M_{m1}	M_{m2}	M_{13}	$15 \mapsto$	$(0.0, 0.0, 1.0)$
où M_{mi} signifie que la fonction est activée pour tout l'intervalle				

La transformation en élongation équivalente à chacun des points de linéarisation est donnée au tableau 4.7. Les 5 autres points sont donnés au tableau 4.4.

TAB. 4.7 Élongations L_i aux points (6 à 15) de linéarisation

i	6	7	8	9	10
L_1	-0.0382	0.1594	0.8068	0.6430	0.7123
L_2	0.1594	-0.0382	0.6430	0.8068	0.7123
L_3	0.4102	0.8643	0.4086	-0.1128	0.7123
L_4	0.1578	0.8060	0.6443	-0.0372	0.7123
L_5	0.8060	0.1578	-0.0372	0.6443	0.7123
L_6	0.8643	0.4102	-0.1128	0.4086	0.7123

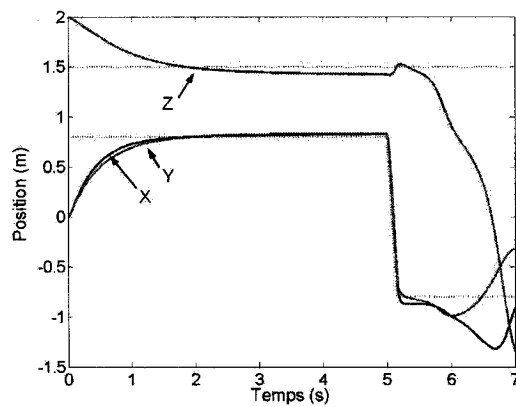
i	11	12	13	14	15
L_1	-0.7649	-0.5123	0.2647	0.07352	-0.5182
L_2	-0.5123	-0.7649	0.07352	0.2647	-0.5182
L_3	-0.2038	0.3312	-0.2058	-0.8630	-0.5182
L_4	-0.5143	0.2638	0.07511	-0.7636	-0.5182
L_5	0.2638	-0.5143	-0.7636	0.07511	-0.5182
L_6	0.3312	-0.2038	-0.8630	-0.2058	-0.5182

4.5.1 Résultats

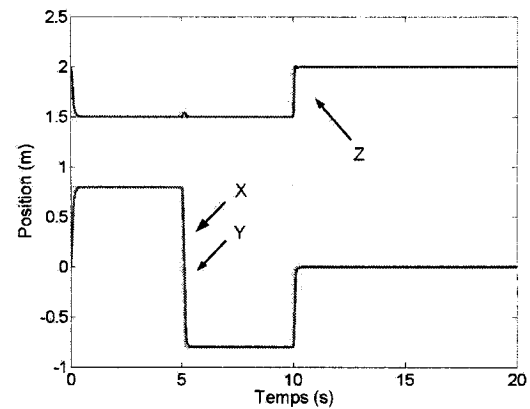
Afin de procéder à la commande, nous utilisons l'analyse préliminaire pour faire le bon choix de contrôleur. Le contrôleur avec intégrateur et placement de pôles est celui privilégié, pour ses capacités à annuler l'erreur et pour sa rapidité. Il faut cependant composer avec des dimensions de matrice augmentées lors de la simulation.

Malgré le nombre augmenté de règles, la résolution des LMI permet de synthétiser un contrôleur pour le modèle flou à 15 règles. Une fois transposé au système réelle de la PGS, il est possible de comparer le comportement du nouveau contrôleur avec celui à 5 règles. Effectivement, pendant que le contrôleur à 5 règles n'était pas en

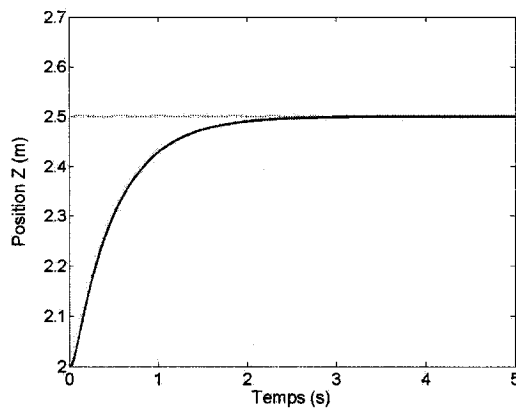
mesure d'effectuer le suivi de trajectoire triangulaire inférieure, celui à 15 règles le permet (4.19(b)).



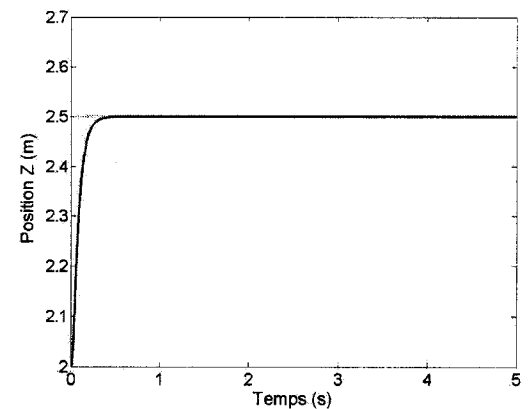
(a) 5 règles : triangle inférieur



(b) 15 règles : triangle inférieur



(c) 5 règle : mouvement ascendant



(d) 15 règle : mouvement ascendant

FIG. 4.19 Suivi de trajectoire simples : 5 vs 15 règles

Aussi, on remarque que la rapidité du système pour ce type de trajectoire est aussi augmentée. En fait, nous avons une meilleure représentation de la PGS, ce qui se traduit par un contrôleur synthétisé plus efficace. On voit aussi la vitesse du système augmenté à la figure (4.19(b)).

4.5.1.1 Suivi de position

Afin de bien identifier le comportement du contrôleur et ses aptitudes face à la commande dans l'espace, nous complexifions les mouvements demandés. Les résultats de la figure 4.20 sont obtenus pour le suivi d'une position quelconque. Le résultat de suivi de position (Fig. 4.20(a),(c),(d)) montre l'efficacité du contrôleur. En effet, malgré des changements très rapides de référence, on note que la PGS tend en tout temps vers cette référence. De plus, les effets de changement brusque de référence n'ont que peu d'effets sur l'orientation de la PGS (Fig. 4.20(b)).

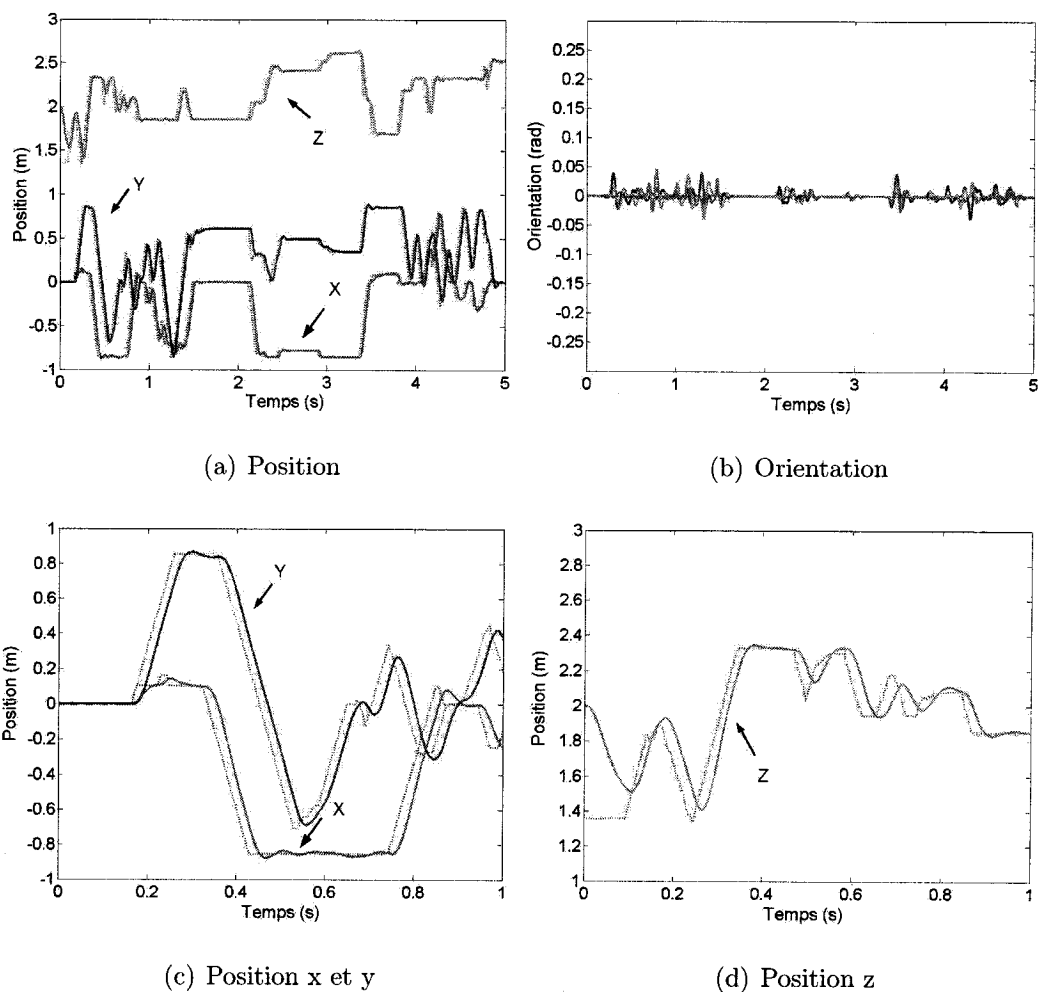


FIG. 4.20 Résultats pour un suivi de position quelconque

4.5.1.2 Suivi d'orientation

Le contrôleur semble en effet permettre à l'orientation de demeurer constante lors des changements de positions. Il est intéressant de regarder le comportement de la PGS si un suivi d'orientation est tenté. Bien que le contrôleur n'a vraisemblablement pas été conçu pour des changements dans l'orientation de la plate-forme, il permet une certaine mobilité dans l'orientation, comme le témoigne la figure 4.21.

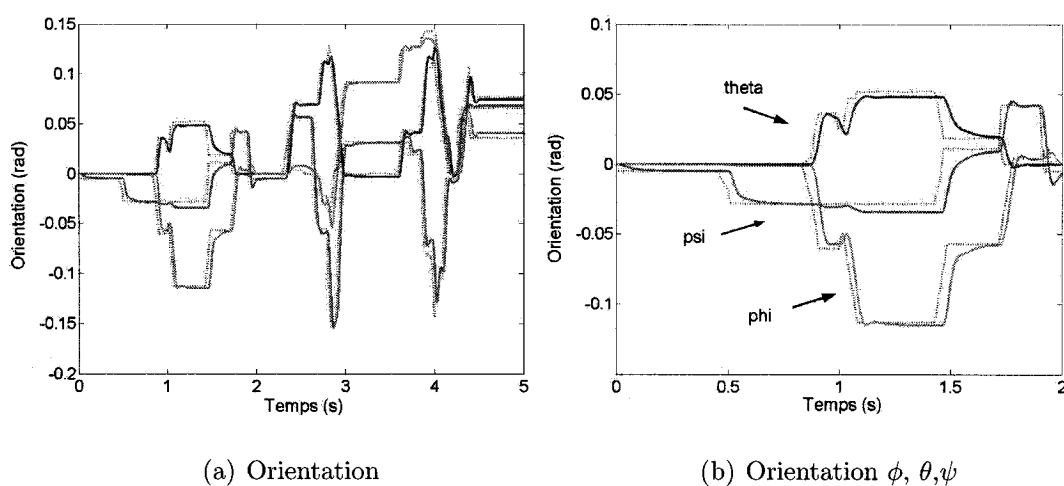


FIG. 4.21 Résultats pour un suivi d'orientation quelconque

La possibilité de commander l'orientation vient de l'ajout de l'intégrateur. Une modification de la référence d'orientation entraîne une modification dans les références des elongations. Ainsi, chacune des membrures tente individuellement d'atteindre la position d'équilibre qu'on lui assigne, en ne distinguant pas si le mouvement de la plate-forme est une translation ou une rotation. Cependant, la commande en orientation est limitée à de faibles angles puisqu'il n'y a aucune considération d'orientation. Il y a aussi présence d'erreurs de suivi.

4.5.1.3 Mouvement général

Le mouvement général mérite d'être essayé. On entend par mouvement général la commande en position de la plate-forme, dans les limites spécifiées par le design du contrôleur, combinée avec des modifications d'orientation modérées. La figure 4.22 montre le suivi en position et en orientation de la plate-forme.

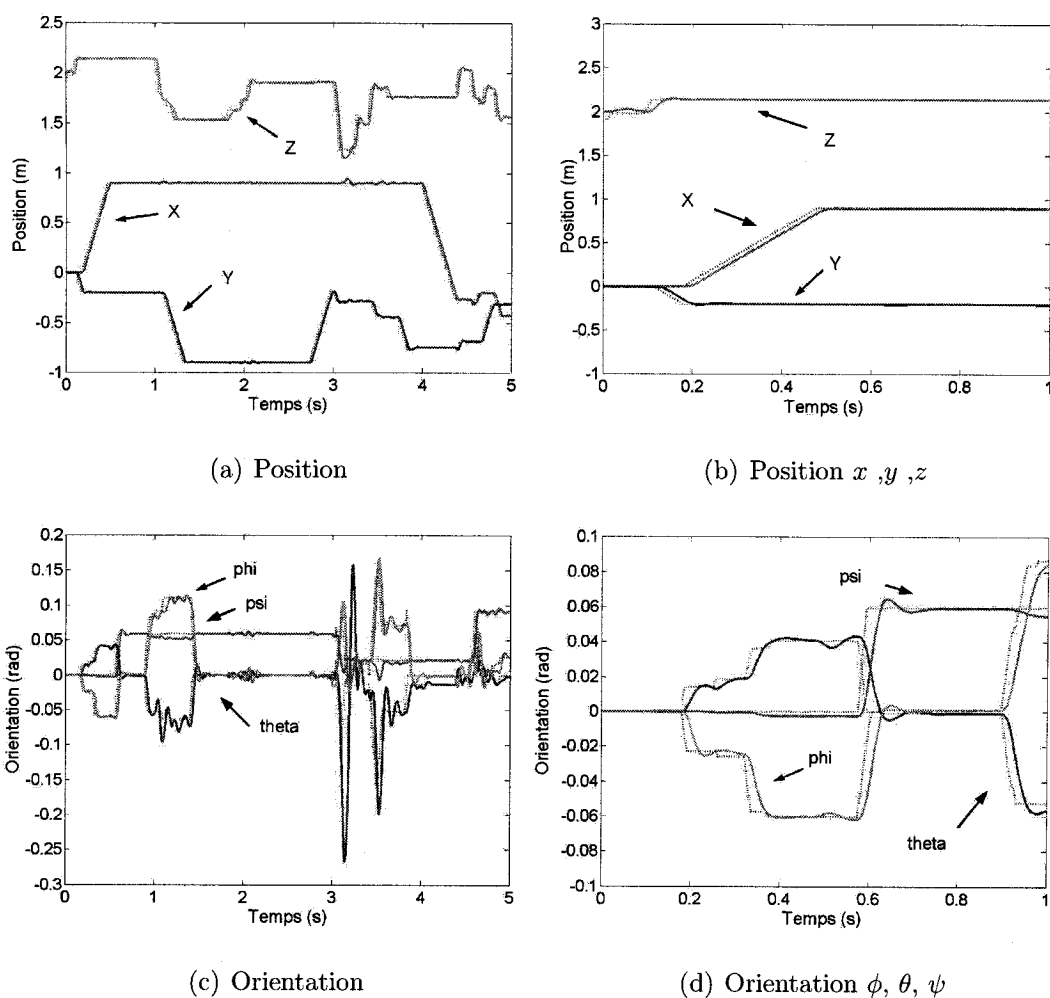


FIG. 4.22 Résultats pour un mouvement complet

On peut donc confirmer que le contrôleur développé permet la commande en position de la plate-forme. Ajout intéressant cependant, on remarque aussi la possibilité de commander la plate-forme en orientation.

4.6 Comparaison des résultats

Les résultats de la section précédente ont montré que le contrôleur permettait de commander la plate-forme efficacement. Pour situer la solution retenue par rapport à d'autres solutions déjà envisagées pour la commande de la PGS, une petite étude comparative est effectuée.

En effet, deux travaux proposent l'utilisation du même modèle de simulation. Cependant, les techniques de commande sont différentes. La méthode basée sur le modèle TS est comparée avec ces deux autres méthodes. Nadimi et al. (2006) utilisent le concept de commande prédictive et Smith et Wendlandt (2002) proposent un PID. Afin de permettre une comparaison, nous définissons les critères de comparaison suivants :

Stabilité : Capacité du système à maintenir la référence désirée pour la position et pour l'orientation. Il est primordial que cette condition soit remplie, avant de pouvoir utiliser les autres critères de comparaison.

Erreur statique : Erreur entre la référence et la position réelle. Ce critère est très important pour la PGS, du fait que ce système est considéré comme étant très précis, il faut rechercher à diminuer ou annuler cette erreur.

Temps de réponse à 2% : Temps mis pour que le système atteigne la valeur en finale $\pm 2\%$. Ce critère est directement lié à la rapidité du système. Également, la PGS est considérée comme un système très rapide. Il faut donc rechercher à maximiser la vitesse du système en minimisant le temps de réponse.

Dépassement : Valeur maximale (en mm) de dépassement entre une valeur de référence et la valeur réelle. Ce critère est parfois très important à minimiser. En effet, lors de l'utilisation de la PGS comme machine-outil, il ne faut pas que l'outil dépasse sa valeur de référence.

Commande maximale : Valeur maximale prise par la variable de commande $u(t)$. Il peut être important de ne pas excéder une certaine valeur de commande, afin d'éviter la saturation de la commande ou tout simplement pour ne pas abîmer certaines composantes du système.

Afin de bien comparer les méthodes, il faut avoir accès à l'information sur la technique utilisée. Puisque cette information est limitée, on se contente de la comparaison par rapport aux tests effectués dans (Nadimi et al. 2006). La figure 4.23 permet de comparer les trois méthodes pour une application identique. L'information sur cette figure n'est pas explicite, mais c'est la seule disponible pour les trois méthodes. Il faut se référer au tableau 4.8 pour plus d'informations de comparaisons. L'objectif de l'application est de positionner toutes les membrures à une extension relative de $L_i = 0.65$.

TAB. 4.8 Comparaison des résultats

Méthode*	Erreur statique (mm)	Temps de réponse (sec)	Dépassement maximal (mm)	Commande maximale (N)
Méthode TS	0	0.5	0	44.016e3
(Nadimi et al. 2006)	0	1.5	0	4.0788e3
(Smith et Wend- landt 2002)	0	0.1	25	1304.0e3

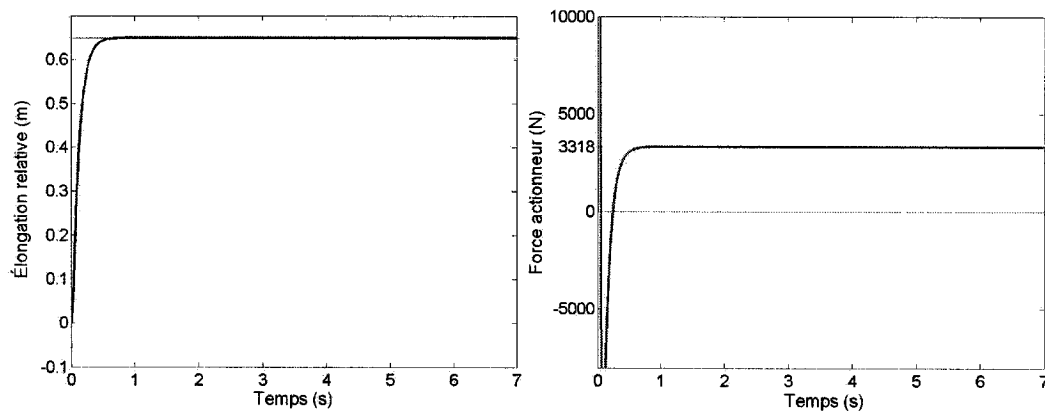
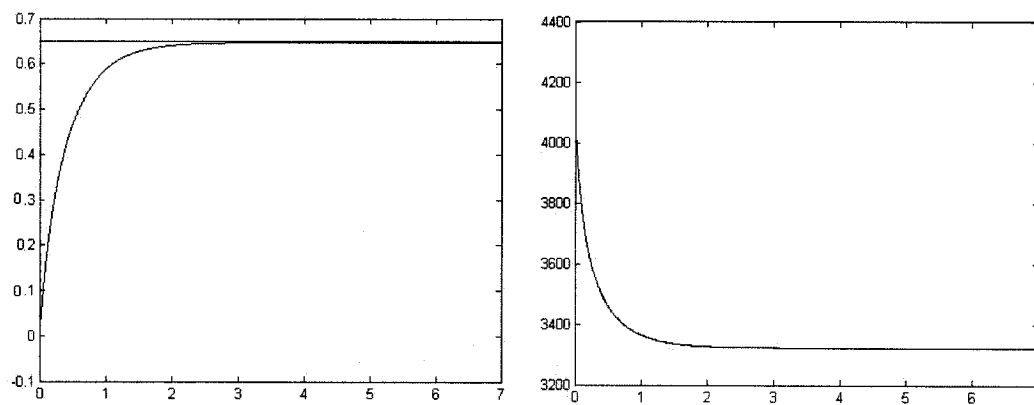
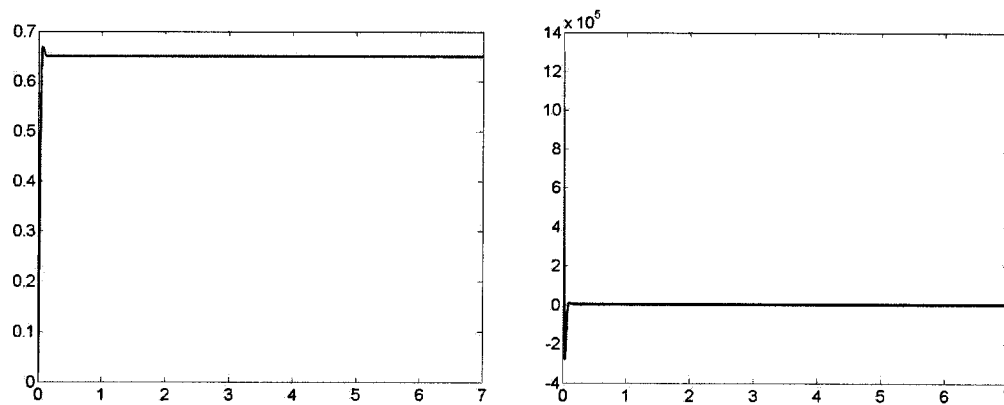
(a) L_1 (Méthode TS proposée)(b) F_1 (Méthode TS proposée)(c) L_1 (Nadimi et al. 2006)(d) F_1 (Nadimi et al. 2006)(e) L_1 (Smith et Wendlandt 2002)(f) F_1 (Smith et Wendlandt 2002)

FIG. 4.23 Comparaison avec d'autres contrôleurs

Par rapport à la commande prédictive, on voit clairement (Fig. 4.23(a)) que la méthode TS présente une plus grande rapidité. Cette augmentation dans la vitesse de réponse se traduit par une demande en force plus grande des actionneurs. Pour ce qui est du dépassement, il est nul dans les deux cas.

Le PID étant conçu par rapport à un seul modèle linéaire de la PGS, les gains utilisés ne représentent pas l'ensemble des comportements possibles de la PGS. Par rapport à ce PID, la méthode TS est moins rapide. Cependant, la très grande rapidité du PID entraîne un dépassement important lors de la montée, en plus de demander des forces énormes aux actionneurs.

Il est très difficile de conclure sur la meilleure des méthodes pour commander la PGS. On peut cependant, en connaissant les méthodes disponibles, faire un choix éclairé de celle qui convient le mieux pour l'application désirée, selon des critères bien définis. Par exemple, si on veut la sollicitation la plus faible des actionneurs, on peut choisir la méthode prédictive. Aussi, si c'est seulement la vitesse du système qui nous intéresse, le contrôleur PID est le contrôleur à préconiser.

Une chose est sûre, la méthode TS proposée permet de commander la PGS en position. Le développement de cette méthode s'appuie sur un formalisme mathématique combiné à la souplesse de la logique floue. De la façon dont elle a été développée, avec les outils de synthèse utilisés, il est toujours possible d'améliorer certains critères de commande. Mais puisqu'un critère de performance est souvent amélioré au détriment d'un autre, il faut déterminer les performances que l'on recherche a priori.

CHAPITRE 5

CONCLUSION ET EXTENSIONS POSSIBLES

Dans le présent travail, il a été question de la problématique de commande des systèmes dynamiques non linéaires. Comme ces systèmes ont la particularité d'être très difficiles à commander puisqu'ils exhibent souvent un comportement imprévisible, il a fallu déterminer une méthode afin de prendre en considération ces difficultés.

En effet, quelques méthodes ont déjà fait leur preuve pour la commande de ce type de systèmes, mais il n'existe pas encore de théorie unifiée permettant leur analyse. Certaines méthodes nécessitent des hypothèses très strictes pour leur élaboration, ce qui peut être très restrictif dans l'étendue des systèmes non linéaires à commander. D'autres méthodes nécessitent des simplifications et/ou des approximations drastiques afin de permettre la synthèse d'un contrôleur adéquat. Ces simplifications peuvent engendrer des limitations dans les possibilités de commande ou encore des performances déficientes. Enfin, selon le système non linéaire considéré, une méthode peut s'avérer tantôt efficace, tantôt inefficace.

Dans le but d'étendre l'analyse à un grand nombre de systèmes non linéaires le choix d'une méthode assez générale a été étudiée. Cette procédure puise ces fondements dans la théorie et les concepts de la logique floue. En effet, le modèle flou TS permet de représenter les systèmes non linéaires par une multitude de modèles localement linéarisés. Sous certaines conditions, le modèle flou TS peut représenter universellement tous les systèmes différentiables à n'importe quel niveau de préci-

sion. De la même façon, un contrôleur non linéaire général par retour d'état peut aussi se faire représenter par un contrôleur flou (PDC). La forme particulière du modèle TS et du contrôleur PDC rend l'analyse et les techniques de design sur des bases de la théorie de Lyapunov adaptées à la commande de système. Dans ce mémoire, on énonce ces problèmes sous forme de LMI, rendant leur résolution possible numériquement. L'ajout d'un terme constant lors de la modélisation, référé à modèle affine TS, a également été étudié.

Pour confronter les concepts étudiés à un problème concret, ils ont été appliqués au manipulateur parallèle de type PGS. Ce système est reconnu pour être très difficile à commander, tel que soulignées lors du survol de la littérature. La modélisation non linéaire de la PGS a été effectuée afin d'en arriver à une forme de modèle adaptée à la procédure de synthèse par le modèle flou TS.

La modélisation de la PGS par des logiciels de modélisation mécanique a été utilisée pour les simulations, conformément au modèle théorique développé. Une fois le modèle de simulation mis sous la forme d'un modèle flou TS simple, on a appliqué plusieurs résultats de stabilité en boucle fermée. Les contrôleurs obtenus ont tous été testés pour l'application du suivi de position selon deux axes. Comme on peut le remarquer, bien que la résolution des conditions de stabilité soient fonctionnelle, la procédure ne garantit pas la stabilité du système réel. En ajoutant quelques contraintes, on réussit cependant à effectuer la tâche planifiée.

Un contrôleur sort du lot, pour ces avantages d'annuler l'erreur en régime permanent et de présenter d'excellentes performances, et ce, pour une programmation assez simple. Il s'agit du contrôleur par retour d'état avec l'ajout de l'intégrateur

et de placement de pôles dans le plan complexe. Il faut toutefois composer avec un contrôleur plus volumineux, car il est synthétisé avec un système dynamique augmenté.

Pour pousser l'analyse et vraiment mettre à profit la modélisation floue, le contrôleur avec l'intégrateur est testé pour une application de positionnement dans l'espace. Cette tâche s'effectue sans aucune difficulté. On remarque que même l'orientation, par l'ajout de l'intégrateur, est commandable quelque peu. Un mouvement quelconque est également réussi avec le contrôleur. Finalement, afin de comparer les résultats obtenus, une petite étude montre que le contrôleur proposé, comparé à un contrôleur prédictif et un contrôleur PID, présente aussi de bons résultats. Cependant, puisque les objectifs précis de simulation ne sont pas connus, il est très difficile de comparer les résultats correctement.

Par ailleurs, ce travail a présenté une méthode de commande pour des systèmes non linéaires, en plus d'une application sur la PGS. Des extensions possibles à ce travail sont énoncés et regroupés en deux catégories. Les extensions associées aux contrôleurs basés sur le modèle flou TS et les extensions associées à la commande de la PGS par cette méthode :

Premièrement, les conditions présentées font uniquement référence au retour d'état. Il est possible d'utiliser le même formalisme pour la synthèse de contrôleur par retour de sorties ou même pour des observateurs de sortie. On peut aussi synthétiser des contrôleurs rejetant un certain niveau de perturbations et étant robuste face aux incertitudes paramétriques. Bref, des méthodes traditionnellement développées pour les modèles linéaires sont transposables dans l'analyse des systèmes modélisés par le modèle flous TS. Enfin, le modèle affine TS est encore très peu étudié, et

mériterait à lui seul une attention toute particulière.

Deuxièmement, le modèle de la PGS utilisé est un modèle au comportement dynamique fidèle, mais présentant plusieurs simplifications. Le modèle flou développé ne tient pas trop compte de la nature des non linéarités de la PGS. Des améliorations sont donc possibles dans l'établissement des règles du modèle flou TS utilisées, afin de maximiser l'approximation. Il serait intéressant de tester la méthode pour une PGS présentant à tout le moins quelques singularités et des limitations spatiales dues aux articulations. La friction dans les différents joints est également quelque chose à étudier. Encore mieux, il faudrait envisager les modifications et les améliorations à la méthode permettant une implantation sur une PGS réelle.

Ce travail est basé sur une étude de concepts théoriques destinés à la commande de systèmes non linéaires. La vérification de ces concepts s'est faite par une application à la commande de la PGS, système fortement non linéaire très difficile à commander. Il serait donc intéressant de tester cette théorie sur d'autres systèmes non linéaires difficilement commandables.

RÉFÉRENCES

ASSAWINCHAICHOTE, W., NUANG, S.K., 2006, "Fuzzy H_∞ Output Feedback Control Design for Singularly Perturbed Systems With Pole Placement Constraints : An LMI Approach," *IEEE Transactions on Fuzzy Systems*, vol.14(3), pp. 361-371.

AKHENAK, A., 2004 *Conception d'observateurs non linéaires par approche multimodèle : application au diagnostic*, Thèse de doctorat, Institut National Polytechnique de Lorraine.

BARON, L., ANGELES, J., Mai 1994, "The decoupling of the direct kinematics of parallel manipulators using redundant sensors," *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, vol.2, pp. 974 - 979.

BARON, L., ANGELES, J., Dec 2000, "The kinematic decoupling of parallel manipulators using joint-sensor data," *Proceedings of 2000 IEEE International Conference on Robotics and Automation*, vol.16(6), pp. 644 - 651.

BLACK, M., 1937, "Vagueness, an exercise in logical analysis," *Philosophy of Science*, vol.4(4), pp. 427-455.

BONEV, M., 2006, Low-Inertia Parallel-Kinematics Systems for Submicron Alignment and Handling. *ParallelMIC-the Parallel Mechanisms Information Center*, [En ligne]. <http://www.parallemic.org> (le 7 juin 2006)

BOUCHON-MEUNIER, B., MARSALA, C., 2003, *Logique floue, principes, aide à la décision*, Lavoisier.

BOUKAS, E.K., 1995, *Système Asservis*, Édition de l'École Polytechnique de Montréal.

- BRUNET, S., 2003, *Synthèse géométrique d'un manipulateur parallèle sphérique à trois degrés de liberté*, Mémoire de maîtrise en génie mécanique, École Polytechnique de Montréal.
- CHERDCHOOSILPA, S., KUNTANAPREEDA, S., CHAIYARATANA, N., 2002, "MIMO Controller Design for a Parallel Manipulator System : A Practitioner's Approach," *IEEE International Conference on Industrial Technology*, pp. 673-677.
- CHILALI, M., GAHINET, P., 1996, " H_∞ Design with Pole Placement Constraints : An LMI Approach," *IEEE Transactions on Automatic Control*, vol.41(5), pp. 358-367.
- CHILALI, M., GAHINET, P., APKARIAN, P., 1999, "Robust Pole Placement in LMI Regions," *IEEE Transactions on Automatic Control*, vol.44(12), pp. 2257-2270.
- CHILALI, M., MAQUIN, D., RAGOT, J., 2002, "Nonquadratic stability analysis of Takagi-Sugeno models," *IEEE Conference on Decision and Control*, vol.2, pp. 2143-2148.
- CHUNG, I.-F., CHANG, H.-H., LIN, C.-T., 1999, "Fuzzy control of a six-degree motion platform with stability analysis," *Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics*, vol.1, pp. 325-330.
- CIOCAN, A., 2005, *Contribution à l'analyse et à la commande d'un véhicule automobile*, Thèse de doctorat, Université de Picardie Jules Verne-Amiens.
- COLLINS, C. L., LONG, G. L., 1995, "On the duality of twist/wrench distributions in serial and parallel chain robot manipulators," *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol.1, pp. 526-531.
- DASGUPTA, B., MRUTHYUNJAVA, T.S., 1996, "A constructive predictor-corrector algorithm for the direct position kinematic problem for a general 6-6 Stewart platform," *Mechanism and Machine Theory*, vol. 31(6), pp. 799-811.

- DASGUPTA, B., MRUTHYUNJAVA, T.S., 1998, "A Newton-Euler Formulation for the Inverse Dynamics of the Stewart Platform Manipulator," *Mechanism and Machine Theory*, vol. 33(8), pp. 1135-1152.
- DASGUPTA, B., MRUTHYUNJAVA, T.S., 2000, "The Stewart Platform Manipulator : a Review," *Mechanism and Machine Theory*, vol. 35, pp. 15-40.
- DO, W.Q.D., YANG, D.C.H., Juin 1988, "Inverse dynamic analysis and simulation of a platform type of robot," *Journal of Robotic Systems*, vol.5(3), pp. 209-227.
- FANG, C.-H., LIU, Y.-S., KAU, S.-W., HONG, L., LEE, C.-H., 2006, "A New LMI-Based Approach to Relaxed Quadratic Stabilisation of TS Fuzzy Control Systems," *IEEE Transaction on Fuzzy Systems*, vol.14(3), pp. 386-397.
- FARINWATA, S.S., FILEV, D., LANGARI, R., *Fuzzy Control*, Wiley, 2000.
- FICHTER, E. F., 1986, "A Stewart platform-based manipulator : General theory and practical construction," *Int. J. Robotics Research*, vol.5, pp. 157-186.
- GARAGIC, D., SRINIVASAN, K., 2004, "Contouring Control of Stewart Platform Based Machine Tools," *Proceedings of the 2004 American Control Conference*, pp. 3831-3838.
- GENG, Z., HAYNES, L.S., LEE, J.D., CARROLL, R.L., 1992, "On the dynamic model and kinematic analysis of a class of Stewart platforms," *Robotics and Autonomous Systems*, vol.9(4), pp. 237-254.
- GOSSELIN, C.M., 1996, "Parallel computational algorithms for the kinematics and dynamics of planar and spatial parallel manipulators," *Transactions of the ASME. Journal of Dynamic Systems*, vol.118(1), pp. 22-28.
- GOUGH, V.E., WHITEHALL, S.G., May 1962, "Universal tyre test machine," *Proceedings of the FISITA Ninth International Technical Congress*, pp. 117-137.
- HAICHANG, L., JIHAI, J., CELESTINE, O., 2006, "Nonlinear Control via Exact Linearization for Hydrostatic Transmission System with Secondary Regulation," *Systems and Control in Aerospace and Astronautics*, pp. 868 - 871.

- HANN-SHING, J., CHING-CHIH, T., CHI-TAI, L., 2005, "Flight path control design for glide-slope tracking by backstepping," *IEEE International Conference on Mechatronics*, pp. 887-892.
- HARIB, K., SRIVASAN, K., 2003, "Kinematic and Dynamic Analysis of Stewart Platform-based Machine Tool Structures," *Robotica*, vol. 21, pp. 541-554.
- HONG, S.K., NAM, Y., 2003, "Stable fuzzy control system design with pole-placement constraint : an LMI approach," *Computers in Industry*, vol. (51) pp. 1-12.
- HSIAO, C.C., SU, S.-F., CHUANG, C.-C., 2004, "A Sliding manner Compensation Control for Affine TSK Fuzzy Control Systems," *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, pp.4374-4379.
- HSIAO, C.C., SU, S.-F., CHUANG, C.-C., 2004, "Hybrid Compensation Control for Affine TSK Fuzzy Control Systems," *IEEE Transactions on Systems, Man, and Cybernetics* vol. 34(4), pp. 115-123, August 2004.
- HUANG, C.-I., CHANG, C.-F., YU, M.-Y., 2004, "Sliding-Mode Tracking Control of the Stewart Platform," *5th Asian Control Conference*, pp. 562-569.
- HUNT, K.H., PRIMROSE, E.J.F., YU, M.-Y., 1993, "Assembly configurations of some in-parallel actuated manipulators," *Mechanism and Machine Theory*, vol.28(1) pp. 31-42.
- HWANG, T.W., TAHK, M., PARK, C., 2005, "Adaptive sliding mode neural net control for missile autopilot," *International Conference on Control and Automation*, vol.1(1) pp. 26-31.
- INNOCENTI, C., PARENTI-CASTELLI, V., 1993, "Closed-form direct position analysis of a 5-5 parallel mechanism," *ASME Journal of Mechanical Design*, vol. 115(3), pp. 515-521.
- JI, Z., 1994, "Dynamic decomposition for Stewart platforms," *ASME Journal of Mechanical Design*, vol. 116(1), pp. 67-69.

- JIANYUN, Z., YUANZHANG, S., 2005, "Backstepping Design of Nonlinear Optimal Control," *Transmission and Distribution Conference and Exhibition : Asia and Pacific, 2005 IEEE/PES*, pp. 1-6.
- KANG, J.-Y., KIM, D.H., LEE, K.-I., 1996, "Robust tracking control of Stewart Platform," *Proceedings of the 35th Conference on Decision and Control*, pp. 3014-3019.
- KHALIL, H.K., 2002, *Nonlinear systems, Third Edition*, Prentice-Hall.
- KHALIL, W., GUEGAN, S., 2002, "A novel solution for the dynamic modeling of Gough-Stewart manipulators," *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 817-822.
- KHALIL, W., GUEGAN, S., 2004, "Inverse and direct dynamic modeling of Gough-Stewart robots," *IEEE Transactions on Robotics*, vol. 1, pp. 754-761.
- KIM, N., LEE, C., May 1998, "High speed tracking control of Stewart Platform manipulator via enhanced sliding mode control," *Proceedings 1998 IEEE International Conference on Robotics and Automation*, pp. 817-822.
- KIM, E., LEE, C.-H., 2000, "New Approches to Relaxed Quadratic Stability Condition of Fuzzy Control Systems," *IEEE Transactions on Fuzzy Systems*, vol. 8(5), pp. 523-534.
- KIM, E., LEE, C.-H., 2005, "Analysis and Design of an Affine Fuzzy System via Bilinear Matrix Inequality," *IEEE Transactions on Fuzzy Systems*, vol. 13(1), pp. 115-123.
- KORBA, P., BABUSKA, R., VERBRUGGEN, H.B., FRANK, P.M., 2003, "Fuzzy Gain Scheduling : Controller and Observer Design Based on Lyapunov Method and Convex Optimization," *IEEE Transactions on Fuzzy Systems*, vol. 11(3), pp. 285-298.

- KOSHKOUEI, A.J., BURNHAM, K.J., ZINOBER, A.S.I., FRANK, P.M., 2003, "Dynamic sliding mode control design," *Control Theory and Applications, IEE Proceedings*, vol. 152(4), pp. 392-396.
- KRSTIC, M., KANELLAKOPOULOS, I., KOKOTOVIC, P.V., 1995, *Nonlinear and adaptative Control Design*, John Wiley & Sons.
- LEE, S.-H., SONG, J.-B., CHOI, W.-C., 2003, "Position Control of a Stewart Platform Using Inverse Dynamics Control with Approximate Dynamics," *Mechanics*, vol. 13, pp. 605-619.
- LIN, L.-C., TSAY, M.-U., 2000, "Modeling and Control of Micropositioning Systems Using Stewart Platforms," *Journal of Robotic Systems*, vol. 17(1), pp. 17-52.
- LIU, M.-J., LI, C.-X., LI, C.-N., 2000, "Dynamics Analysis of the Gough-Stewart Platform Manipulator," *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 94-98.
- LIU, Y.-S., FANG, C.-H., 2003, "A New LMI-BASED Approach to Relaxed Quadratic Stabilization of T-S Fuzzy Control Systems," ..
- LÖFBERG, J., 2004, *YALMIP 3*, [En ligne]. <http://control.ee.ethz.ch/~joloef/yalmip.msql> (2004).
- MAMDANI, E.H., 1974, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the IEE*, vol. 121(12), pp. 1585-1588.
- MANN, G.K.I., SURGENOR, B.W., September 2002, "Model-free Intelligent Control of a 6-dof Stewart-Gough Based Parallel Manipulator," *IEEE International Conference on Control applications*, 18-20.
- MATLAB, 2002, *Getting Started with MATLAB*, The Math Works, Inc.
- MERLET, J.-P., 1987 "Parallel manipulators : Part I : Theory ; design, kinematics, dynamics and control," *Technical Report 646, INRIA, Sophia Antipolis, France*,.
- NA, W., GOU, B., DIONG, B., 2005 "Nonlinear control of PEM fuel cells by exact linearization," *Industry Applications Conference*, vol. 4.

- NADIMI, E.S., BAK, T., IZADI-ZAMANABADI, R., 2006, "Model Predictive Controller Combined with LQG Controller and Velocity Feedback to Control the Stewart Platform," *International Workshop on Advanced Motion Control*, pp. 44-49.
- NGUYEN, H.T., WALKER, E.A., 2006, *A First Course in Fuzzy Logic*, Third edition, Chapman & Hall/CRC.
- OUASSAID, M., CHERKAOUI, M., ZIDANI, Y., 2004, "A nonlinear speed control for a PM synchronous motor using an adaptive backstepping control approach," *Industrial Technology*, vol. 3, pp. 1287-1292.
- RAGHAVAN, M., 1993, "The Stewart platform of general geometry has 40 configurations," *ASME Journal of Mechanical Design*, vol. 115(2), pp. 277-282.
- SEDUMI, 2005, *SeDuMi, Let SeDuMi seduce you, too!*, [En ligne]. <http://sedumi.mcmaster.ca/> (le 20 juin 2005).
- SHEN, P.-H.; LIN, F.-J., 2005, "Intelligent backstepping sliding-mode control using RBFN for two-axis motion control system," *Electric Power Applications, IEE Proceedings*, vol. 152(5), pp. 1321-1342.
- SIMULINK, 2002, *Using Simulink*, The Math Works, Inc.
- SIROUSPOUR, M.R., SALCUDEAN, S.E., 2001, "Nonlinear Control of Hydraulic Robots," *IEEE Transactions on Robotics and Automation*, vol. 17(2), pp. 173-182.
- SLOTINE, J.E., LI, W., 1992, *Applied Nonlinear Systems*, Prentice-Hall.
- SMITH, N., WENDLANDT, J., 2002, "Creating a stewart platform model using simmechanics," *The MathWorks - MATLAB Digest* vol. 10(5).
- STEWART, D., 1965-66, "A Platform with Six Degrees of Freedom," *UK Institution of Mechanical Engineers Proceedings (Part I)*, vol. 180(15), pp. 371-386.
- SUGENO, M., KANG, G.T., 1986, "Fuzzy Modeling and Control of Multilayer Incinerator," *Fuzzy Sets Systems*, vol. 18, pp. 329-346.

- SUN, C.-C., CHUNG, H.-Y., CHANG, W.-J., 2004, "Design of Takagi-Sugeno Fuzzy-Region Controller Based on Fuzzy-Region Concept, Rule Reduction and Robust Control Technique," *IEEE International Conference on Networking, Sensing and Control*, pp. 914-919.
- TAKAGI, T., SUGENO, M., 1985, "Fuzzy Identification of Systems and its Applications to Modeling and Control," *IEEE Trans. Syst. Man. Cyber*, vol. 15, pp. 116-132.
- TANAKA, K., WANG, H.O., 2001, *Fuzzy control systems desing and analysis : a linear matrix inequality approach*, Wiley-InterScience.
- VASQUEZ, R. ; KRSTIC, M., 2005, "Thermal convection loop control by continuous backstepping and singular perturbations Vasquez," *American Control Conference*, pp. 3882-3887.
- WALDRON, K.J. and HUNT, K.H, 1991, "Series-parallel dualities in actively coordinated mechanisms," *Int. J. of Robotics Research*, vol. 10(2), pp. 473-480.
- WANG, J., GOSSELIN, C.M., September 1998, "A new approach for the dynamic analysis of parallel manipulators," *Multibody System Dynamics*, vol. 2(3), pp. 317-334.
- WANG, J., GOSSELIN, C.M., LI, C., March 2002, "Modeling and simulation of robotic systems with closed kinematic chains using the virtual spring approach," *Multibody System Dynamics*, vol. 7(2), pp. 145-170.
- WEN, F., LIANG, C., 1994, "Displacement Analysis of the 6-6 Stewart Platform Mechanisms," *Mechanism and Machine Theory*, vol. 29(4), pp. 547-557.
- XIAO-TAO, C., YUAN-CHUN, L., 2005, "Distributed parameter singular perturbation model and cooperative control of flexible manipulators," *Machine Learning and Cybernetics*, vol. 2(4), pp. 1009-1014.
- ZADEH, L., 1965, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338-353.

ANNEXE I

OUTILS MATHÉMATIQUES

I.1 LMI

Dans le but d'utiliser la puissance du calcul numérique dans la recherche des gains du contrôleur, on introduit la formulation sous forme de LMI. Une LMI est une expression de la forme

$$F(x) \triangleq F_0 + x_1 F_1 + \cdots + x_m F_m > 0 \quad (\text{I.1})$$

où

$x = (x_1, \dots, x_m)$: Vecteur de nombres réels ;

$F_0, \dots, F_m$: Matrices symétrique, i.e., $F_i = F_i^T \in \mathbb{R}^{n \times n}$;

$F(x) > 0$: Matrice définie-positive, i.e., $u^T F(x) u > 0$ pour tout $u \in \mathbb{R}^n$, $u \neq 0$.

Un exemple très connu d'inégalité matricielle est l'inégalité de Lyapunov suivante :

$$F(P) = A^T P + P A < 0. \quad (\text{I.2})$$

Il s'agit d'une inégalité selon la variable P. S'il existe une matrice P symétrique et définie positive telle que la condition (I.2) est respectée, alors le système décrit par la matrice A est stable, i.e. que toutes les valeurs propres de A sont à partie réelle négative. La matrice P alors résultante est définie comme la matrice de Lyapunov.

Si les dimensions du système rendent la recherche d'une matrice P difficile, on peut se servir des logiciels d'optimisation. Le logiciel SeDuMi (2005) utilise des « semidefinite programm » (SDP) pour résoudre des problèmes avec des objectifs linéaires, des LMI et des contraintes sous forme d'égalités linéaires de type suivant :

minimiser	$c^T x$	Objectif
sujet à	$F_0 + x_1 F_1 + \cdots + x_m F_m > 0$	LMI
et	$Ax = b$	Égalité

I.2 Complément de Schur

Soit une matrice M telle que

$$M = \begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix}$$

où X et Z sont des matrices symétriques. La matrice M est définie positive si et seulement si

$$\begin{cases} Z > 0 \\ X - YZ^{-1}Y^T > 0 \end{cases}$$

ou

$$\begin{cases} X > 0 \\ Z - Y^T X^{-1}Y > 0 \end{cases}$$

La matrice $X - YZ^{-1}Y^T$ est le complément de Schur $X(Z)$ dans M .

ANNEXE II

VALEURS NUMÉRIQUES

Membrure associée	u_{r1}	u_{r2}	u_{r3}	u_{r4}	u_{r5}
L_1	4.0788	5.4506	1.8857	1.5427	6.3799
L_2	4.0788	5.4506	6.3799	1.5427	1.8857
L_3	4.0788	0.7191	3.3727	6.5598	4.6265
L_4	4.0788	5.2330	0.5313	3.3884	6.0968
L_5	4.0788	5.2330	6.0968	3.3884	0.5313
L_6	4.0788	0.7191	4.6265	6.5598	3.3727
Membrure associée	u_{r6}	u_{r7}	u_{r8}	u_{r9}	u_{r10}
L_1	5.2442	3.5816	1.4482	3.6783	3.2811
L_2	3.5816	5.2442	3.6783	1.4482	3.2811
L_3	2.6558	1.7076	4.6557	4.9310	3.2811
L_4	5.1946	2.5571	1.8353	4.3670	3.2811
L_5	2.5571	5.1946	4.3670	1.8353	3.2811
L_6	1.7076	2.6558	4.9310	4.6557	3.2811
Membrure associée	u_{r11}	u_{r12}	u_{r13}	u_{r14}	u_{r15}
L_1	6.8116	4.8606	2.1713	5.3014	6.9841
L_2	4.8606	6.8116	5.3014	2.1713	6.9841
L_3	3.7631	2.5496	6.5249	6.2949	6.9841
L_4	7.0124	3.7598	2.6954	5.6839	6.9841
L_5	3.7598	7.0124	5.6839	2.6954	6.9841
L_6	2.5496	3.7631	6.2949	6.5249	6.9841

TAB. II.1 Commande u_{ri} aux points d'équilibre ($\times 10^3$)

TAB. II.2 Matrices $(a_{21})_i$ et $(b_{21})_i$

$(a_{21})_1 =$	$\begin{bmatrix} 9.0418 & -4.7826 & -2.0513 & -2.8144 & 1.4778 & 2.4682 \\ -4.7826 & 9.0418 & 2.4682 & 1.4778 & -2.0513 & -2.8144 \\ 2.4682 & 1.4778 & -2.0513 & -2.8144 & 9.0418 & 2.4682 \\ -2.8144 & 2.4682 & 9.0418 & 1.4778 & -4.7826 & -2.0513 \\ -2.0513 & -2.8144 & -4.7826 & 9.0418 & -2.0513 & -2.8144 \\ -2.8144 & -2.0513 & -4.7826 & 9.0418 & -2.0513 & -2.8144 \end{bmatrix}$	$(b_{21})_1 =$	$\begin{bmatrix} 0.0021 & 0.0001 & 0.0002 & -0.0010 & 0.0002 & 0.0002 \\ -0.0001 & 0.0021 & 0.0002 & 0.0002 & -0.0010 & 0.0002 \\ 0.0002 & 0.0002 & -0.0010 & 0.0002 & 0.0002 & -0.0001 \\ -0.0010 & 0.0002 & 0.0002 & -0.0010 & 0.0002 & 0.0002 \\ 0.0002 & -0.0010 & 0.0002 & 0.0002 & -0.0001 & 0.0021 \\ 0.0002 & 0.0002 & 0.0002 & 0.0002 & -0.0001 & 0.0021 \end{bmatrix}$
$(a_{21})_2 =$	$\begin{bmatrix} 3.4824 & -1.2666 & 3.2631 & -4.1495 & -0.6491 & -1.6396 \\ -0.4380 & 3.6427 & 1.3066 & -3.3848 & -4.2957 & 2.7035 \\ 0.4971 & -0.4208 & 8.5494 & -5.5879 & -1.3513 & -0.5100 \\ -1.1762 & 0.0295 & -6.7990 & 12.8341 & 5.0526 & -3.7862 \\ 3.4004 & -3.5971 & -4.8455 & 7.9828 & 10.7451 & -6.7486 \\ 0.8602 & 1.1940 & -1.2949 & 0.3655 & -3.6832 & 7.3346 \end{bmatrix}$	$(b_{21})_2 =$	$\begin{bmatrix} 0.0023 & 0.0000 & 0.0003 & 0.0007 & 0.0001 & 0.0003 \\ 0.0000 & 0.0021 & 0.0007 & 0.0021 & -0.0001 & 0.0010 \\ 0.0003 & 0.0007 & 0.0021 & 0.0021 & 0.0001 & 0.0003 \\ -0.0009 & 0.0001 & -0.0001 & -0.0001 & 0.0021 & 0.0002 \\ 0.0001 & -0.0010 & 0.0001 & 0.0001 & 0.0003 & 0.0023 \\ 0.0003 & -0.0001 & -0.0010 & 0.0002 & -0.0001 & 0.0023 \end{bmatrix}$
$(a_{21})_3 =$	$\begin{bmatrix} 6.4228 & -2.2318 & 2.5509 & -2.9828 & 1.1034 & -0.1288 \\ -7.9442 & 11.7858 & 9.2841 & -7.3552 & -5.6718 & 5.5483 \\ -2.4051 & 3.7871 & 13.6712 & -7.6851 & -1.5594 & 0.3761 \\ 0.6544 & -2.5597 & -7.2435 & 10.6891 & 0.1453 & 0.1739 \\ 4.6668 & -5.8836 & -6.1679 & 4.3152 & 6.1982 & -3.0713 \\ -1.4776 & -0.1118 & -4.8953 & 4.1288 & -0.4084 & 3.4972 \end{bmatrix}$	$(b_{21})_3 =$	$\begin{bmatrix} 0.0023 & -0.0001 & 0.0004 & 0.0004 & -0.0009 & 0.0002 \\ -0.0001 & 0.0022 & 0.0002 & 0.0019 & 0.0002 & -0.0008 \\ 0.0004 & 0.0002 & 0.0019 & -0.0001 & 0.0005 & 0.0005 \\ -0.0009 & 0.0002 & -0.0001 & 0.0002 & 0.0005 & 0.0021 \\ 0.0002 & -0.0010 & 0.0002 & 0.0002 & 0.0005 & 0.0023 \\ 0.0002 & -0.0000 & -0.0008 & 0.0005 & 0.0000 & 0.0023 \end{bmatrix}$
$(a_{21})_4 =$	$\begin{bmatrix} 9.2598 & -6.6524 & -1.8284 & -0.0260 & -0.8844 & 3.9928 \\ -6.6210 & 11.7826 & 1.9824 & 0.5187 & 0.1947 & -1.2035 \\ -4.9481 & 4.9106 & 7.4757 & -4.4484 & 4.6941 & -5.0000 \\ -4.5107 & 3.8319 & 0.0899 & 3.7597 & 0.9013 & -0.8709 \\ 1.0630 & -3.2330 & -0.0627 & -1.4535 & 5.5886 & -1.0497 \\ 3.2189 & -5.5884 & -4.3831 & 2.1188 & -4.9460 & 9.2640 \end{bmatrix}$	$(b_{21})_4 =$	$\begin{bmatrix} 0.0018 & -0.0001 & -0.0001 & -0.0010 & 0.0002 & 0.0004 \\ -0.0001 & 0.0021 & -0.0002 & 0.0023 & -0.0001 & 0.0010 \\ -0.0001 & -0.0002 & 0.0023 & -0.0001 & 0.0002 & -0.0010 \\ 0.0002 & -0.0002 & -0.0001 & 0.0002 & 0.0006 & 0.0001 \\ 0.0002 & -0.0009 & 0.0002 & 0.0002 & 0.0022 & -0.0000 \\ 0.0004 & 0.0004 & -0.0010 & 0.0001 & -0.0000 & 0.0022 \end{bmatrix}$
$(a_{21})_5 =$	$\begin{bmatrix} 12.0829 & -7.5676 & 1.4284 & -2.5543 & -5.3910 & 4.9705 \\ -3.0762 & 7.5920 & -2.0283 & 1.3561 & -2.1117 & 0.5180 \\ -3.3934 & 2.5689 & 2.5959 & 0.6903 & 6.1835 & -6.6844 \\ -7.7856 & 7.1047 & -2.3664 & 4.8638 & 5.2793 & -6.4134 \\ -1.1835 & -0.0758 & 1.3723 & -0.3981 & 11.1867 & -6.5694 \\ 5.2493 & -2.6696 & 0.0773 & -1.0937 & -8.0457 & 11.4203 \end{bmatrix}$	$(b_{21})_5 =$	$\begin{bmatrix} 0.0020 & -0.0000 & -0.0001 & 0.0005 & 0.0005 & 0.0004 \\ -0.0000 & 0.0021 & 0.0006 & 0.0001 & -0.0009 & 0.0003 \\ -0.0001 & 0.0006 & 0.0006 & 0.0025 & -0.0000 & 0.0009 \\ -0.0009 & 0.0001 & -0.0000 & 0.0025 & 0.0002 & -0.0001 \\ 0.0005 & -0.0009 & 0.0001 & 0.0002 & 0.0021 & -0.0000 \\ 0.0004 & 0.0003 & -0.0009 & -0.0001 & -0.0000 & 0.0017 \end{bmatrix}$
$(a_{21})_6 =$	$\begin{bmatrix} 6.1356 & -3.3073 & 2.0257 & -2.8943 & -2.1844 & 0.8148 \\ -1.3237 & 4.3973 & -0.5496 & -0.5014 & -2.5714 & 1.3269 \\ -0.1792 & 0.0368 & 4.8812 & -2.1965 & 1.4571 & -2.4564 \\ -3.0060 & 2.3476 & -4.1334 & 7.0958 & 3.1892 & -3.4046 \\ 0.4964 & -1.1148 & -0.9296 & 1.9063 & 8.4060 & -5.1765 \\ 1.9155 & -0.4516 & -0.5385 & -0.4278 & -4.6536 & 7.2474 \end{bmatrix}$	$(b_{21})_6 =$	$\begin{bmatrix} 0.0023 & -0.0000 & 0.0000 & 0.0000 & -0.0010 & 0.0008 \\ -0.0000 & 0.0023 & 0.0010 & 0.0010 & 0.0000 & 0.0000 \\ 0.0000 & 0.0010 & 0.0010 & 0.0024 & -0.0000 & 0.0010 \\ -0.0010 & 0.0000 & -0.0000 & 0.0024 & 0.0007 & -0.0001 \\ 0.0002 & -0.0010 & 0.0000 & 0.0007 & 0.0023 & -0.0000 \\ 0.0008 & 0.0000 & -0.0010 & -0.0001 & -0.0000 & 0.0022 \end{bmatrix}$
$(a_{21})_7 =$	$\begin{bmatrix} 4.3888 & -1.5839 & 2.7584 & -3.1929 & 0.0431 & -0.8097 \\ -3.6033 & 6.1192 & 3.5975 & -4.0716 & 4.0219 & 3.3235 \\ -0.4217 & 0.3447 & 9.0210 & -5.6528 & -1.1962 & 0.0065 \\ -0.0998 & -0.9481 & -5.9014 & 9.7303 & 1.3010 & -1.2990 \\ 3.5201 & -3.9852 & 4.8161 & 7.5197 & -4.9465 & 0.0001 \\ -0.4269 & 0.3773 & -2.6453 & 2.2311 & -2.1023 & 5.0806 \end{bmatrix}$	$(b_{21})_7 =$	$\begin{bmatrix} 0.0024 & 0.0000 & 0.0003 & -0.0009 & -0.0001 & 0.0006 \\ 0.0000 & 0.0023 & 0.0008 & 0.0001 & -0.0010 & -0.0002 \\ 0.0003 & 0.0008 & 0.0021 & -0.0001 & 0.0000 & -0.0009 \\ -0.0009 & 0.0001 & 0.0001 & 0.0021 & 0.0007 & 0.0002 \\ 0.0001 & -0.0010 & 0.0000 & 0.0007 & 0.0023 & -0.0000 \\ 0.0006 & -0.0002 & -0.0009 & 0.0002 & -0.0000 & 0.0024 \end{bmatrix}$
$(a_{21})_8 =$	$\begin{bmatrix} 6.5762 & -3.8260 & 0.0783 & -1.1641 & 0.0430 & 1.1766 \\ -5.4923 & 8.7765 & 3.5337 & -2.2786 & -1.9502 & 1.4794 \\ -2.7271 & 3.0132 & 8.6115 & -5.0158 & 1.0876 & -1.6956 \\ -1.4856 & 0.2888 & -3.4164 & 6.2425 & -0.1475 & 0.2626 \\ 1.8351 & -3.1446 & -1.7299 & 0.4253 & 4.3469 & -1.4020 \\ 0.0743 & -1.7186 & -3.4263 & 2.3645 & -2.1772 & 4.9267 \end{bmatrix}$	$(b_{21})_8 =$	$\begin{bmatrix} 0.0023 & -0.0001 & 0.0000 & 0.0000 & -0.0010 & 0.0008 \\ -0.0001 & 0.0023 & 0.0005 & 0.0000 & 0.0000 & 0.0001 \\ 0.0000 & 0.0005 & 0.0022 & -0.0001 & 0.0001 & -0.0010 \\ -0.0010 & 0.0000 & 0.0022 & 0.0008 & 0.0002 & 0.0002 \\ -0.0001 & -0.0010 & 0.0001 & 0.0008 & 0.0023 & 0.0000 \\ 0.0008 & 0.0001 & -0.0010 & 0.0002 & 0.0000 & 0.0024 \end{bmatrix}$

TAB. II.3 Matrices $(a_{21})_i$ et $(b_{21})_i$ (...suite)

$(a_{21})_9 =$	$\begin{bmatrix} 9.3275 & -6.2695 & 0.0667 & -0.9716 & -2.8116 & 3.5885 \\ -4.7168 & 8.3555 & -0.9248 & 0.8927 & -0.7266 & -0.2120 \\ -3.4593 & 2.8030 & 4.2341 & -1.9043 & 4.5531 & -4.9121 \\ -4.6763 & 4.2513 & -1.1076 & 3.5897 & 1.9779 & -2.5225 \\ 0.2163 & -1.2913 & 0.9472 & -1.4522 & 7.2141 & -3.6280 \\ 3.3701 & -3.4531 & -1.7299 & 0.8218 & -5.9701 & 9.3001 \end{bmatrix}$	$(b_{21})_9 =$	$\begin{bmatrix} 0.0021 & -0.0000 & -0.0001 & -0.0010 & 0.0002 & 0.0007 \\ -0.0000 & 0.0022 & 0.0006 & 0.0025 & -0.0000 & 0.0001 \\ -0.0001 & 0.0006 & -0.0001 & 0.0025 & -0.0000 & -0.0001 \\ -0.0010 & 0.0001 & -0.0000 & 0.0025 & -0.0000 & -0.0001 \\ 0.0002 & -0.0009 & 0.0001 & 0.0008 & 0.0022 & -0.0000 \\ 0.0007 & 0.0003 & -0.0010 & -0.0001 & -0.0000 & 0.0021 \end{bmatrix}$
$(a_{21})_{10} =$	$\begin{bmatrix} 5.7399 & -3.2890 & 1.2043 & -1.8450 & -1.1123 & 0.7734 \\ -3.2890 & 5.7399 & 0.7734 & -1.1123 & -1.8450 & 1.2043 \\ -1.1123 & 0.7734 & 5.7399 & -3.2890 & 1.2043 & -1.8450 \\ -1.8450 & 1.2043 & -3.2890 & 0.7734 & -1.1123 & 1.2043 \\ 1.2043 & -1.8450 & -1.1123 & 0.7734 & 5.7399 & -3.2890 \\ 0.7734 & -1.1123 & -1.8450 & 1.2043 & -3.2890 & 5.7399 \end{bmatrix}$	$(b_{21})_{10} =$	$\begin{bmatrix} 0.0024 & 0.0000 & -0.0000 & -0.0010 & -0.0000 & 0.0011 \\ 0.0000 & 0.0024 & 0.0011 & -0.0000 & -0.0010 & -0.0000 \\ -0.0000 & 0.0011 & 0.0024 & -0.0000 & -0.0010 & -0.0000 \\ -0.0010 & -0.0000 & 0.0011 & 0.0024 & 0.0011 & -0.0000 \\ -0.0000 & -0.0010 & -0.0000 & 0.0011 & 0.0024 & 0.0000 \\ 0.0011 & -0.0000 & -0.0010 & -0.0000 & 0.0000 & 0.0024 \end{bmatrix}$
$(a_{21})_{11} =$	$\begin{bmatrix} 10.7132 & -5.5428 & 2.4690 & -4.7801 & -4.3211 & 2.9564 \\ -2.3601 & 8.0719 & 1.4481 & -2.9734 & -4.8993 & 1.9273 \\ -2.6987 & 2.8254 & 8.5132 & -3.2978 & 2.9810 & -4.8925 \\ -5.3012 & 3.3761 & -6.4941 & 12.8031 & 7.0826 & -5.3778 \\ 0.1387 & -1.3327 & -0.5997 & 4.3855 & 13.4256 & -7.1309 \\ 4.0426 & -0.3830 & -0.7921 & -1.0426 & -6.7213 & 11.3478 \end{bmatrix}$	$(b_{21})_{11} =$	$\begin{bmatrix} 0.0021 & -0.0001 & 0.0002 & -0.0009 & 0.0004 & -0.0002 \\ -0.0001 & 0.0021 & 0.0002 & 0.0002 & -0.0001 & 0.0002 \\ 0.0002 & 0.0002 & 0.0021 & -0.0001 & 0.0004 & 0.0001 \\ -0.0009 & 0.0002 & -0.0001 & 0.0021 & -0.0004 & -0.0001 \\ 0.0004 & -0.0009 & 0.0002 & -0.0004 & 0.0020 & -0.0002 \\ -0.0002 & 0.0002 & -0.0010 & 0.0001 & -0.0002 & 0.0018 \end{bmatrix}$
$(a_{21})_{12} =$	$\begin{bmatrix} 7.6978 & -2.2501 & 4.3239 & -5.6463 & -0.4718 & 0.4256 \\ -6.2294 & 11.3901 & 8.1035 & -8.6815 & -7.2025 & 5.6666 \\ -1.5873 & 2.9200 & 15.1763 & -9.4526 & -2.5233 & -0.0971 \\ -0.1067 & -2.1072 & -9.5695 & 16.7218 & 4.7028 & -2.3921 \\ 5.3268 & -6.2975 & -7.9044 & 9.6501 & 13.6925 & -8.0725 \\ 1.0898 & 0.1886 & -4.6119 & 3.6245 & -2.9164 & 8.9665 \end{bmatrix}$	$(b_{21})_{12} =$	$\begin{bmatrix} 0.0021 & -0.0001 & 0.0005 & -0.0009 & 0.0001 & -0.0004 \\ -0.0001 & 0.0019 & 0.0019 & -0.0000 & 0.0003 & -0.0010 \\ 0.0005 & -0.0000 & 0.0018 & -0.0001 & 0.0003 & -0.0009 \\ -0.0009 & 0.0003 & -0.0001 & 0.0018 & -0.0002 & 0.0005 \\ 0.0001 & -0.0010 & 0.0003 & -0.0002 & 0.0019 & -0.0002 \\ -0.0004 & 0.0000 & -0.0009 & 0.0005 & -0.0002 & 0.0021 \end{bmatrix}$
$(a_{21})_{13} =$	$\begin{bmatrix} 10.5198 & -5.1871 & 0.1164 & -2.3067 & -0.4311 & 3.4638 \\ -7.9240 & 14.3296 & 6.3041 & -1.7871 & -1.7045 & 0.8135 \\ -2.7149 & 5.8815 & 14.5480 & -7.7562 & 0.4511 & -1.6038 \\ -2.7924 & 0.5258 & -5.0603 & 10.1336 & 1.9193 & -1.1966 \\ 3.1612 & -7.4262 & -7.0449 & 3.9364 & 8.7863 & -3.0195 \\ 2.7576 & -5.5003 & -7.6242 & 3.7084 & -3.4860 & 8.8394 \end{bmatrix}$	$(b_{21})_{13} =$	$\begin{bmatrix} 0.0019 & -0.0003 & 0.0003 & -0.0010 & 0.0001 & -0.0003 \\ -0.0003 & 0.0019 & -0.0007 & 0.0003 & -0.0007 & 0.0003 \\ 0.0003 & -0.0007 & 0.0018 & -0.0002 & 0.0003 & -0.0009 \\ -0.0010 & 0.0003 & -0.0002 & 0.0019 & 0.0000 & 0.0004 \\ 0.0001 & -0.0009 & 0.0003 & 0.0000 & 0.0021 & -0.0001 \\ -0.0003 & 0.0003 & -0.0009 & 0.0004 & -0.0001 & 0.0021 \end{bmatrix}$
$(a_{21})_{14} =$	$\begin{bmatrix} 15.9616 & -10.3685 & -0.4124 & -1.2678 & -4.8656 & 7.5839 \\ -7.2919 & 13.7966 & 1.2118 & 0.8864 & -0.9837 & -1.0944 \\ -5.9941 & 5.5735 & 7.9418 & -2.7966 & 7.7252 & -8.8335 \\ -8.9679 & 8.0575 & -1.0582 & 6.7922 & 4.9304 & -5.4591 \\ -0.2120 & -2.2788 & 0.2830 & 0.6881 & 12.3625 & -5.7720 \\ 8.1461 & -7.1336 & -2.9832 & 0.8759 & -10.4642 & 16.9959 \end{bmatrix}$	$(b_{21})_{14} =$	$\begin{bmatrix} 0.0017 & -0.0001 & 0.0001 & -0.0009 & 0.0005 & -0.0001 \\ -0.0001 & 0.0019 & -0.0005 & 0.0003 & -0.0008 & 0.0005 \\ 0.0001 & -0.0005 & 0.0022 & -0.0002 & 0.0003 & -0.0010 \\ -0.0009 & 0.0003 & -0.0002 & 0.0022 & -0.0002 & 0.0001 \\ 0.0005 & -0.0008 & 0.0003 & -0.0002 & 0.0020 & -0.0001 \\ -0.0001 & 0.0005 & -0.0010 & 0.0001 & -0.0001 & 0.0018 \end{bmatrix}$
$(a_{21})_{15} =$	$\begin{bmatrix} 19.0040 & -9.0828 & 2.1845 & -5.7608 & -4.8777 & 7.9474 \\ -9.0828 & 19.0040 & 7.9474 & -4.8777 & -5.7608 & 2.1845 \\ -4.8777 & 7.9474 & 19.0040 & -9.0828 & 2.1845 & -5.7608 \\ -5.7608 & 2.1845 & -9.0828 & 19.0040 & 7.9474 & -4.8777 \\ 2.1845 & -5.7608 & -4.8777 & 7.9474 & 19.0040 & -9.0828 \\ 7.9474 & -4.8777 & -5.7608 & 2.1845 & -9.0828 & 19.0040 \end{bmatrix}$	$(b_{21})_{15} =$	$\begin{bmatrix} 0.0018 & -0.0002 & 0.0004 & -0.0009 & 0.0004 & -0.0009 \\ -0.0002 & 0.0018 & -0.0009 & 0.0004 & -0.0009 & 0.0004 \\ 0.0004 & -0.0009 & 0.0018 & -0.0002 & 0.0004 & -0.0009 \\ -0.0009 & 0.0004 & -0.0002 & 0.0018 & -0.0009 & 0.0004 \\ 0.0004 & -0.0009 & 0.0004 & -0.0009 & 0.0018 & -0.0002 \\ -0.0009 & 0.0004 & -0.0009 & 0.0004 & -0.0002 & 0.0018 \end{bmatrix}$

TAB. II.4 Gains K_i du contrôleur

$K_1 =$	3.3120	-0.1177	-0.5087	1.5703	-0.5084	-0.7492	0.0939	-0.0024	-0.0134	0.0450	-0.0134	-0.0202	-0.0134	-0.0202
	-0.1189	3.3178	-0.7520	-0.5106	1.5726	-0.5108	-0.0024	0.0939	-0.0202	-0.0134	0.0450	-0.0134	-0.0202	-0.0134
	-0.5069	-0.7477	3.3094	-0.1167	-0.5075	1.5691	-0.0134	-0.0202	0.0939	-0.0024	-0.0134	0.0450	-0.0134	0.0450
	1.5669	-0.5052	-0.1156	3.3025	-0.7449	-0.5050	0.0450	-0.0134	-0.0024	0.0938	-0.0202	-0.0134	-0.0202	-0.0134
	-0.5088	1.5715	-0.5096	-0.7497	3.3143	-0.1190	-0.0134	0.0450	-0.0134	-0.0202	0.0939	-0.0024	-0.0134	-0.0024
	-0.7505	-0.5099	1.5717	-0.5097	-0.1191	3.3162	-0.0202	-0.0134	0.0450	-0.0134	-0.0024	0.0939	-0.0024	0.0939
$K_2 =$	3.2076	-0.1605	-0.8740	1.4883	-0.4949	-1.0370	0.0908	-0.0037	-0.0239	0.0425	-0.0131	-0.0282	-0.0131	-0.0282
	-0.1485	3.8065	-1.5534	-0.5835	1.7501	-0.2933	-0.0036	0.1077	-0.0429	-0.0157	0.0500	-0.0074	-0.0157	-0.0074
	-0.8696	-1.5546	3.9555	-0.2626	-0.6738	1.6691	-0.0239	-0.0429	0.1118	-0.0066	-0.0180	0.0478	-0.0066	0.0478
	1.4868	-0.5865	-0.2584	3.3438	-0.8430	-0.7251	0.0425	-0.0157	-0.0066	0.0948	-0.0230	-0.0195	-0.0230	-0.0195
	-0.4806	1.7497	-0.6648	-0.8378	3.2396	0.1321	-0.0129	0.0500	-0.0179	-0.0229	0.0918	0.0046	-0.0229	0.0046
	-1.0236	-0.2918	1.6816	-0.7218	0.1296	3.2115	-0.0281	-0.0074	0.0478	-0.0195	0.0046	0.0909	-0.0195	0.0909
$K_3 =$	3.2183	0.0627	-1.1174	1.6529	0.0004	-1.0539	0.0911	0.0029	-0.0306	0.0471	0.0010	-0.0289	0.0010	-0.0289
	0.0740	3.3118	-0.6626	-0.7270	1.7742	-0.1127	0.0030	0.0939	-0.0180	-0.0196	0.0509	-0.0025	-0.0196	-0.0025
	-1.1205	-0.6760	3.8145	-0.5023	-0.6639	1.5748	-0.0307	-0.0181	0.1078	-0.0134	-0.0176	0.0450	-0.0134	0.0450
	1.6428	-0.7393	-0.5038	4.1231	-1.1024	-1.1811	0.0470	-0.0198	-0.0134	0.1167	-0.0301	-0.0326	-0.0301	-0.0326
	0.0111	1.7729	-0.6504	-1.0917	3.6157	-0.0958	0.0011	0.0508	-0.0175	-0.0300	0.1025	-0.0020	-0.0300	-0.0020
	-1.0641	-0.1314	1.5772	-1.1882	-0.1140	3.2234	-0.0290	-0.0027	0.0450	-0.0327	-0.0021	0.0911	-0.0327	0.0911
$K_4 =$	4.4466	-0.3382	-0.1588	2.2025	-1.0698	-1.0076	0.1260	-0.0086	-0.0034	0.0628	-0.0295	-0.0277	-0.0295	-0.0277
	-0.3585	3.3441	-0.2425	-0.8011	1.5157	-0.6517	-0.0088	0.0945	-0.0058	-0.0215	0.0431	-0.0175	-0.0215	-0.0175
	-0.1656	-0.2302	2.8549	0.0976	-0.3270	1.2930	-0.0035	-0.0057	0.0812	0.0037	-0.0087	0.0371	0.0037	0.0371
	2.1866	-0.7870	0.0965	3.6812	-1.4724	-0.4098	0.0627	-0.0214	0.0037	0.1046	-0.0410	-0.0109	-0.0410	-0.0109
	-1.1021	1.5007	-0.3519	-1.5003	3.4656	-0.1640	-0.0298	0.0429	-0.0089	-0.0412	0.0978	-0.0039	-0.0412	-0.0039
	-1.0351	-0.6584	1.2781	-0.4279	-0.1582	3.3068	-0.0279	-0.0176	0.0369	-0.0111	-0.0038	0.0934	-0.0111	0.0934
$K_5 =$	3.7007	-0.3803	-0.0478	1.4338	-1.1439	-0.7621	0.1047	-0.0101	-0.0005	0.0412	-0.0315	-0.0204	-0.0315	-0.0204
	-0.3990	3.8846	-1.6182	-0.6201	1.8039	-1.5130	-0.0103	0.1094	-0.0445	-0.0165	0.0512	-0.0413	-0.0165	-0.0413
	-0.0437	-1.5917	3.2453	0.2595	-0.8840	1.9800	-0.0005	-0.0443	0.0921	0.0082	-0.0243	0.0568	-0.0243	0.0568
	1.4341	-0.5941	0.2602	2.7340	-0.8584	0.0621	0.0412	-0.0163	0.0082	0.0779	-0.0236	0.0028	-0.0236	0.0028
	-1.1557	1.8090	-0.9002	-0.8777	3.6807	-0.5766	-0.0316	0.0512	-0.0244	-0.0238	0.1038	-0.0150	-0.0238	-0.0150
	-0.7489	-1.4797	1.9953	0.0705	-0.5516	4.4963	-0.0203	-0.0410	0.0569	0.0029	-0.0148	0.1271	-0.0148	0.1271

TAB. II.5 Gains K_i du contrôleur (...suite)

$K_6 =$	3.6871	-0.4508	-0.9037	1.4284	-1.0881	-1.9573	0.4352	-0.0479	-0.1044	0.1722	-0.1255	-0.2263
	0.2150	4.0036	-2.5886	-0.5735	1.5063	-1.5336	0.0311	0.4736	-0.3001	-0.0656	0.1807	-0.1772
	-0.7484	-1.9124	3.6457	-0.4165	-0.9243	1.5920	-0.0868	-0.2218	0.4291	-0.0427	-0.1053	0.1909
	1.8230	-1.0665	0.0538	3.3621	-1.7736	-0.9435	0.2213	-0.1242	0.0146	0.3972	-0.2074	-0.1043
	-0.7870	1.8603	-1.1103	-1.4742	3.5139	-0.7336	-0.0896	0.2236	-0.1266	-0.1721	0.4122	-0.0781
	-1.6166	-0.9247	1.7725	-0.6281	-0.0836	3.6255	-0.1885	-0.1052	0.2128	-0.0678	0.0003	0.4269
$K_7 =$	3.5129	-0.4238	-1.3441	1.6982	-0.5730	-1.9664	0.4146	-0.0439	-0.1554	0.2018	-0.0666	-0.2264
	0.2397	3.6266	-2.0923	-0.7828	1.4746	-0.7915	0.0348	0.4289	-0.2444	-0.0878	0.1792	-0.0923
	-1.1709	-1.7962	4.0798	-0.7388	-1.1436	1.5449	-0.1351	-0.2093	0.4750	-0.0784	-0.1291	0.1827
	1.5207	-1.1182	-0.0499	3.8518	-1.7400	-1.1220	0.1829	-0.1251	0.0051	0.4497	-0.2017	-0.1259
	-0.7966	1.9323	-0.7342	-2.0351	3.6856	-0.1263	-0.0937	0.2356	-0.0798	-0.2396	0.4354	-0.0069
	-1.6494	-0.5218	1.7590	-1.2713	-0.0505	3.3130	-0.1889	-0.0583	0.2102	-0.1448	0.0037	0.3905
$K_8 =$	3.5599	-0.4578	-1.0176	1.6932	-0.7106	-1.6666	0.4172	-0.0462	-0.1154	0.2013	-0.0817	-0.1956
	0.3671	3.0896	-1.3403	-0.5717	1.1117	-1.0822	0.0533	0.3620	-0.1579	-0.0610	0.1354	-0.1269
	-0.5050	-1.3323	3.2209	-0.2539	-0.9342	1.0648	-0.0523	-0.1567	0.3767	-0.0219	-0.1082	0.1289
	1.8023	-0.9598	-0.1362	3.9413	-1.9821	-0.9663	0.2162	-0.1068	-0.0061	0.4639	-0.2295	-0.1115
	-0.8120	1.8683	-0.8059	-2.0728	3.7841	-0.4196	-0.0913	0.2257	-0.0922	-0.2385	0.4494	-0.0453
	-2.2987	-0.6371	1.8401	-1.5975	0.0592	3.7566	-0.2642	-0.0712	0.2225	-0.1850	0.0141	0.4460
$K_9 =$	3.9606	-0.7135	-0.4948	1.5046	-1.3544	-1.4318	0.4624	-0.0761	-0.0555	0.1805	-0.1553	-0.1688
	0.2576	3.6568	-1.8022	-0.4951	1.3243	-1.8774	0.0426	0.4262	-0.2082	-0.0545	0.1561	-0.2156
	-0.0664	-1.6527	2.9807	0.0564	-0.9762	1.2051	-0.0026	-0.1914	0.3538	0.0124	-0.1142	0.1463
	2.2923	-1.3804	0.2284	3.6580	-2.3974	-0.6336	0.2766	-0.1615	0.0331	0.4352	-0.2796	-0.0716
	-0.9982	2.0536	-1.2243	-1.6746	3.8569	-1.1583	-0.1101	0.2437	-0.1425	-0.1917	0.4512	-0.1283
	-2.1232	-0.7951	1.4704	-0.9514	0.2467	3.8675	-0.2480	-0.0860	0.1797	-0.1083	0.0398	0.4541
$K_{10} =$	3.9562	-0.3809	-1.0368	1.7767	-1.0365	-2.3218	0.4658	-0.0393	-0.1202	0.2125	-0.1202	-0.2710
	0.3746	3.9099	-2.5327	-0.6773	1.5636	-1.4977	0.0517	0.4608	-0.2957	-0.0766	0.1876	-0.1748
	-0.7735	-2.1685	3.7757	-0.2992	-1.0321	1.5618	-0.0883	-0.2523	0.4445	-0.0293	-0.1193	0.1871
	2.0940	-1.2569	0.0543	4.0938	-2.5566	-1.0877	0.2517	-0.1454	0.0140	0.4835	-0.2980	-0.1251
	-0.7922	2.0923	-1.1837	-2.1923	3.9580	-0.6189	-0.0900	0.2513	-0.1369	-0.2546	0.4670	-0.0669
	-2.4057	-0.8247	1.8806	-1.2543	0.1369	3.9152	-0.2795	-0.0933	0.2265	-0.1447	0.0242	0.4624

TAB. II.6 Gains K_i du contrôleur (...suite)

$K_{11} =$	3.3054	-1.0826	-0.5501	0.6368	-1.4019	-0.2269	0.3959	-0.1218	-0.0609	0.0865	-0.1617	-0.0276
	0.0550	3.5659	-1.9763	-0.5819	0.7887	-2.0575	0.0187	0.4342	-0.2276	-0.0606	0.1067	-0.2359
	-0.3344	-1.1710	3.8007	-1.0881	-1.6125	1.2301	-0.0331	-0.1263	0.4245	-0.1151	-0.1765	0.1384
	1.5617	-0.5577	-0.3800	3.3242	-0.4683	-1.3517	0.1977	-0.0614	-0.0263	0.3891	-0.0598	-0.1407
	-0.8426	2.1567	-1.6934	0.0395	4.0671	-2.3392	-0.0836	0.2583	-0.1915	0.0051	0.4676	-0.2558
	-0.3314	-0.7346	1.5334	-0.4776	-0.1203	3.6519	-0.0406	-0.0798	0.1951	-0.0375	0.0145	0.4435
$K_{12} =$	3.3714	-1.1919	-1.6515	1.3409	-0.4504	-0.6692	0.4054	-0.1336	-0.1888	0.1634	-0.0566	-0.0759
	0.2616	3.5125	-0.8949	-0.9827	0.8726	-0.4959	0.0452	0.4308	-0.1058	-0.1019	0.1262	-0.0564
	-1.4957	-0.6700	4.9409	-2.1112	-2.0834	1.3971	-0.1617	-0.0747	0.5404	-0.2224	-0.2226	0.1520
	1.1655	-0.9610	-0.5884	4.0403	-0.4417	-1.7160	0.1416	-0.0900	-0.0391	0.4582	-0.0526	-0.1826
	-0.6431	2.2717	-0.5532	-0.6261	4.0922	-0.9817	-0.0714	0.2869	-0.0497	-0.0800	0.4791	-0.0984
	-0.5888	-0.1124	1.6914	-1.7029	-0.3379	3.3617	-0.0633	-0.0024	0.2077	-0.1855	-0.0097	0.4044
$K_{13} =$	3.8293	-1.5853	-1.7283	1.6015	-0.8247	-0.2415	0.4591	-0.1741	-0.1926	0.1984	-0.0970	-0.0320
	0.2124	3.8387	0.5959	-0.8040	0.6922	-0.8610	0.0513	0.4628	0.0732	-0.0812	0.0964	-0.0959
	-0.9648	1.3751	4.7727	-1.4790	-0.7729	0.9912	-0.0940	0.1511	0.5247	-0.1525	-0.0825	0.1103
	1.9156	-1.2557	-0.5771	4.2748	-1.1580	-1.0970	0.2340	-0.1268	-0.0402	0.5012	-0.1318	-0.1197
	-0.9663	2.0371	0.0198	-1.0679	3.5572	-1.1564	-0.0986	0.2581	0.0190	-0.1189	0.4201	-0.1243
	-1.1918	0.1441	1.7202	-1.7880	-0.1657	3.1733	-0.1375	0.0406	0.2272	-0.2026	0.0040	0.3838
$K_{14} =$	4.6306	-1.9467	-0.3958	1.1247	-2.0186	0.9069	0.5431	-0.2140	-0.0382	0.1445	-0.2298	0.0950
	-0.0410	3.8729	-0.7002	-0.4484	0.8626	-2.5486	0.0282	0.4579	-0.0741	-0.0446	0.1028	-0.2836
	0.3409	-0.0987	3.3668	-0.3918	-1.1599	0.9076	0.0515	-0.0070	0.3821	-0.0366	-0.1318	0.1047
	2.8444	-1.5892	-0.0009	3.8381	-1.5886	-0.3826	0.3519	-0.1847	0.0145	0.4608	-0.1885	-0.0338
	-1.6333	2.2956	-1.4084	-0.5926	4.3053	-2.9248	-0.1621	0.2672	-0.1596	-0.0601	0.4895	-0.3207
	-0.9519	-0.4722	1.0292	-0.8116	0.1810	3.4557	-0.1237	-0.0276	0.1443	-0.0832	0.0560	0.4052
$K_{15} =$	5.3581	-2.9046	-1.6244	0.9875	-1.8136	2.3514	0.6125	-0.3129	-0.1727	0.1259	-0.1978	0.2601
	-0.5913	4.6309	1.3736	-1.1983	0.3773	-1.3117	-0.0230	0.5427	0.1591	-0.1103	0.0660	-0.1287
	-0.9302	1.6716	5.9053	-2.8030	-2.2801	1.6161	-0.0875	0.1691	0.6333	-0.2898	-0.2456	0.1788
	1.7935	-0.9285	-0.9408	5.0289	1.2561	-1.2239	0.2282	-0.0761	-0.0575	0.5686	0.1350	-0.1090
	-1.9446	2.9904	-0.6439	1.2028	5.6773	-3.1396	-0.1968	0.3651	-0.0484	0.1268	0.6387	-0.3255
	0.5217	0.1212	2.1525	-1.4558	-0.5378	4.5090	0.0485	0.0567	0.2793	-0.1330	0.0017	0.5181

TAB. II.7 Gains K_e du contrôleur

$K_{e1} =$	3.3073	-0.1153	-0.5079	1.5693	-0.5076	-0.7496	0.0939	-0.0024	-0.0134	0.0450	-0.0134	-0.0202
	-0.1165	3.3132	-0.7523	-0.5098	1.5716	-0.5100	-0.0024	0.0939	-0.0202	-0.0134	0.0450	-0.0134
	-0.5061	-0.7480	3.3047	-0.1143	-0.5067	1.5681	-0.0134	-0.0202	0.0939	-0.0024	-0.0134	0.0450
	1.5659	-0.5044	-0.1132	3.2978	-0.7453	-0.5042	0.0450	-0.0134	-0.0024	0.0938	-0.0202	-0.0134
	-0.5080	1.5705	-0.5088	-0.7501	3.3097	-0.1166	-0.0134	0.0450	-0.0134	-0.0202	0.0939	-0.0024
	-0.7509	-0.5091	1.5707	-0.5089	-0.1167	3.3115	-0.0202	-0.0134	0.0450	-0.0134	-0.0024	0.0939
$K_{e2} =$	3.2063	-0.1598	-0.8733	1.4869	-0.4960	-1.0341	0.0908	-0.0037	-0.0239	0.0425	-0.0131	-0.0282
	-0.1492	3.8050	-1.5509	-0.5841	1.7495	-0.2932	-0.0036	0.1077	-0.0429	-0.0157	0.0500	-0.0074
	-0.8695	-1.5543	3.9495	-0.2587	-0.6715	1.6667	-0.0239	-0.0429	0.1118	-0.0066	-0.0180	0.0478
	1.4871	-0.5862	-0.2554	3.3375	-0.8454	-0.7220	0.0425	-0.0157	-0.0066	0.0948	-0.0230	-0.0195
	-0.4825	1.7505	-0.6620	-0.8407	3.2351	0.1343	-0.0129	0.0500	-0.0179	-0.0229	0.0918	0.0046
	-1.0240	-0.2923	1.6796	-0.7198	0.1323	3.2067	-0.0281	-0.0074	0.0478	-0.0195	0.0046	0.0909
$K_{e3} =$	3.2135	0.0654	-1.1150	1.6507	-0.0007	-1.0532	0.0911	0.0029	-0.0306	0.0471	0.0010	-0.0289
	0.0770	3.3065	-0.6658	-0.7232	1.7755	-0.1150	0.0030	0.0939	-0.0180	-0.0196	0.0509	-0.0025
	-1.1173	-0.6787	3.8064	-0.4980	-0.6623	1.5738	-0.0307	-0.0181	0.1078	-0.0134	-0.0176	0.0450
	1.6396	-0.7359	-0.4988	4.1158	-1.1026	-1.1803	0.0470	-0.0198	-0.0134	0.1167	-0.0301	-0.0326
	0.0104	1.7729	-0.6492	-1.0908	3.6132	-0.0954	0.0011	0.0508	-0.0175	-0.0300	0.1025	-0.0020
	-1.0612	-0.1333	1.5750	-1.1867	-0.1130	3.2212	-0.0290	-0.0027	0.0450	-0.0327	-0.0021	0.0911
$K_{e4} =$	4.4410	-0.3351	-0.1579	2.2010	-1.0692	-1.0093	0.1260	-0.0086	-0.0034	0.0628	-0.0295	-0.0277
	-0.3545	3.3372	-0.2441	-0.8004	1.5137	-0.6497	-0.0088	0.0945	-0.0058	-0.0215	0.0431	-0.0175
	-0.1636	-0.2314	2.8519	0.0994	-0.3281	1.2935	-0.0035	-0.0057	0.0812	0.0037	-0.0087	0.0371
	2.1855	-0.7867	0.0969	3.6785	-1.4716	-0.4106	0.0627	-0.0214	0.0037	0.1046	-0.0410	-0.0109
	-1.1005	1.4993	-0.3525	-1.4987	3.4621	-0.1624	-0.0298	0.0429	-0.0089	-0.0412	0.0978	-0.0039
	-1.0353	-0.6558	1.2789	-0.4279	-0.1562	3.3028	-0.0279	-0.0176	0.0369	-0.0111	-0.0038	0.0934
$K_{e5} =$	3.6946	-0.3768	-0.0480	1.4341	-1.1404	-0.7636	0.1047	-0.0101	-0.0005	0.0412	-0.0315	-0.0204
	-0.3960	3.8793	-1.6165	-0.6207	1.8014	-1.5103	-0.0103	0.1094	-0.0445	-0.0165	0.0512	-0.0413
	-0.0443	-1.5905	3.2434	0.2596	-0.8839	1.9792	-0.0005	-0.0443	0.0921	0.0082	-0.0243	0.0568
	1.4344	-0.5950	0.2609	2.7323	-0.8583	0.0633	0.0412	-0.0163	0.0082	0.0779	-0.0236	0.0028
	-1.1526	1.8061	-0.9001	-0.8776	3.6735	-0.5721	-0.0316	0.0512	-0.0244	-0.0238	0.1038	-0.0150
	-0.7513	-1.4775	1.9940	0.0711	-0.5474	4.4894	-0.0203	-0.0410	0.0569	0.0029	-0.0148	0.1271

TAB. II.8 Gains K_{ei} du contrôleur (...suite)

$K_{e6} =$	3.6033	-0.4406	-0.8832	1.3960	-1.0634	-1.9128	0.4253	-0.0468	-0.1021	0.1683	-0.1227	-0.2211
	0.2101	3.9126	-2.5298	-0.5605	1.4720	-1.4988	0.0304	0.4628	-0.2933	-0.0641	0.1766	-0.1732
	-0.7314	-1.8689	3.5628	-0.4071	-0.9033	1.5559	-0.0848	-0.2168	0.4193	-0.0417	-0.1029	0.1865
	1.7816	-1.0423	0.0526	3.2857	-1.7333	-0.9221	0.2162	-0.1213	0.0143	0.3881	-0.2027	-0.1019
	-0.7691	1.8180	-1.0851	-1.4407	3.4340	-0.7170	-0.0875	0.2185	-0.1237	-0.1682	0.4028	-0.0763
	-1.5799	-0.9036	1.7322	-0.6139	-0.0817	3.5431	-0.1842	-0.1028	0.2079	-0.0662	0.0003	0.4172
$K_{e7} =$	3.4331	-0.4142	-1.3136	1.6596	-0.5600	-1.9217	0.4052	-0.0429	-0.1519	0.1972	-0.0651	-0.2212
	0.2342	3.5442	-2.0447	-0.7650	1.4411	-0.7735	0.0340	0.4192	-0.2388	-0.0858	0.1751	-0.0902
	-1.1442	-1.7554	3.9871	-0.7220	-1.1176	1.5098	-0.1320	-0.2045	0.4642	-0.0766	-0.1261	0.1786
	1.4862	-1.0928	-0.0488	3.7642	-1.7004	-1.0965	0.1787	-0.1223	0.0050	0.4395	-0.1971	-0.1231
	-0.7785	1.8883	-0.7175	-1.9889	3.6018	-0.1235	-0.0916	0.2303	-0.0780	-0.2341	0.4255	-0.0068
	-1.6119	-0.5100	1.7190	-1.2424	-0.0494	3.2377	-0.1846	-0.0569	0.2054	-0.1415	0.0036	0.3817
$K_{e8} =$	3.4790	-0.4474	-0.9945	1.6547	-0.6944	-1.6287	0.4077	-0.0452	-0.1128	0.1968	-0.0799	-0.1912
	0.3587	3.0194	-1.3098	-0.5587	1.0864	-1.0576	0.0521	0.3538	-0.1543	-0.0596	0.1324	-0.1240
	-0.4935	-1.3021	3.1477	-0.2481	-0.9130	1.0406	-0.0512	-0.1532	0.3681	-0.0214	-0.1058	0.1259
	1.7613	-0.9380	-0.1332	3.8518	-1.9371	-0.9444	0.2113	-0.1043	-0.0060	0.4533	-0.2243	-0.1090
	-0.7935	1.8258	-0.7876	-2.0256	3.6981	-0.4101	-0.0892	0.2206	-0.0901	-0.2331	0.4392	-0.0443
	-2.2464	-0.6226	1.7982	-1.5612	0.0579	3.6713	-0.2582	-0.0696	0.2174	-0.1808	0.0138	0.4359
$K_{e9} =$	3.8706	-0.6973	-0.4835	1.4704	-1.3236	-1.3993	0.4519	-0.0744	-0.0542	0.1764	-0.1517	-0.1650
	0.2518	3.5737	-1.7612	-0.4838	1.2942	-1.8348	0.0417	0.4165	-0.2035	-0.0533	0.1526	-0.2107
	-0.0649	-1.6151	2.9129	0.0551	-0.9540	1.1777	-0.0025	-0.1871	0.3458	0.0121	-0.1116	0.1430
	2.2402	-1.3491	0.2232	3.5749	-2.3429	-0.6192	0.2703	-0.1578	0.0323	0.4253	-0.2732	-0.0699
	-0.9756	2.0069	-1.1964	-1.6365	3.7692	-1.1320	-0.1076	0.2382	-0.1393	-0.1874	0.4409	-0.1253
	-2.0750	-0.7770	1.4370	-0.9298	0.2410	3.7796	-0.2424	-0.0840	0.1756	-0.1058	0.0389	0.4438
$K_{e10} =$	3.8663	-0.3723	-1.0133	1.7363	-1.0130	-2.2690	0.4552	-0.0384	-0.1175	0.2077	-0.1174	-0.2649
	0.3661	3.8210	-2.4751	-0.6619	1.5281	-1.4637	0.0505	0.4504	-0.2890	-0.0748	0.1833	-0.1708
	-0.7560	-2.1192	3.6899	-0.2924	-1.0086	1.5263	-0.0863	-0.2466	0.4344	-0.0286	-0.11660.1828	
	2.0464	-1.2283	0.0531	4.0007	-2.4985	-1.0630	0.2460	-0.1421	0.0137	0.4725	-0.2912	-0.1223
	-0.7742	2.0447	-1.1568	-2.1425	3.8680	-0.6048	-0.0879	0.2456	-0.1338	-0.2488	0.4564	-0.0654
	-2.3510	-0.8059	1.8378	-1.2258	0.1338	3.8263	-0.2732	-0.0912	0.2213	-0.1414	0.02360.4519	

TAB. II.9 Gains K_{ei} du contrôleur (...suite)

$K_{e11} =$	3.2365	-1.0600	-0.5386	0.6235	-1.3727	-0.2221	0.3876	-0.1192	-0.0596	0.0847	-0.1583	-0.0271
	0.0539	3.4916	-1.9351	-0.5698	0.7722	-2.0147	0.0184	0.4252	-0.2229	-0.0593	0.1045	-0.2310
	-0.3275	-1.1466	3.7215	-1.0654	-1.5789	1.2045	-0.0324	-0.1236	0.4157	-0.1127	-0.1728	0.1355
	1.5291	-0.5461	-0.3721	3.2550	-0.4586	-1.3235	0.1936	-0.0601	-0.0257	0.3810	-0.0586	-0.1378
	-0.8251	2.1118	-1.6581	0.0387	3.9824	-2.2905	-0.0819	0.2529	-0.1875	0.0050	0.4579	-0.2505
	-0.3245	-0.7193	1.5014	-0.4677	-0.1178	3.5758	-0.0397	-0.0782	0.1911	-0.0367	0.0142	0.4343
$K_{e12} =$	3.3012	-1.1671	-1.6170	1.3129	-0.4410	-0.6552	0.3969	-0.1308	-0.1849	0.1600	-0.0554	-0.0743
	0.2561	3.4393	-0.8762	-0.9622	0.8544	-0.4856	0.0442	0.4218	-0.1036	-0.0998	0.1236	-0.0552
	-1.4646	-0.6561	4.8380	-2.0672	-2.0400	1.3680	-0.1584	-0.0731	0.5292	-0.2177	-0.2180	0.1489
	1.1412	-0.9410	-0.5761	3.9561	-0.4325	-1.6803	0.1387	-0.0881	-0.0383	0.4486	-0.0515	-0.1788
	-0.6297	2.2244	-0.5416	-0.6131	4.0069	-0.9613	-0.0699	0.2809	-0.0487	-0.0783	0.4692	-0.0963
	-0.5766	-0.1100	1.6562	-1.6674	-0.3309	3.2916	-0.0620	-0.0024	0.2034	-0.1816	-0.0095	0.3960
$K_{e13} =$	3.7496	-1.5523	-1.6923	1.5681	-0.8075	-0.2364	0.4495	-0.1705	-0.1886	0.1943	-0.0950	-0.0313
	0.2080	3.7588	0.5835	-0.7873	0.6778	-0.8430	0.0502	0.4532	0.0716	-0.0795	0.0944	-0.0939
	-0.9447	1.3465	4.6732	-1.4482	-0.7568	0.9706	-0.0920	0.1479	0.5137	-0.1493	-0.0808	0.1080
	1.8757	-1.2295	-0.5651	4.1857	-1.1339	-1.0741	0.2291	-0.1241	-0.0394	0.4908	-0.1290	-0.1172
	-0.9462	1.9947	0.0194	-1.0456	3.4831	-1.1323	-0.0966	0.2528	0.0186	-0.1164	0.4113	-0.1218
	-1.1670	0.1411	1.6843	-1.7508	-0.1623	3.1072	-0.1346	0.0398	0.2225	-0.1984	0.0039	0.3758
$K_{e14} =$	4.5342	-1.9061	-0.3875	1.1013	-1.9765	0.8880	0.5317	-0.2095	-0.0374	0.1415	-0.2250	0.0931
	-0.0402	3.7922	-0.6856	-0.4391	0.8446	-2.4955	0.0276	0.4484	-0.0725	-0.0436	0.1007	-0.2777
	0.3338	-0.0966	3.2967	-0.3837	-1.1357	0.8887	0.0504	-0.0069	0.3742	-0.0359	-0.1290	0.1025
	2.7851	-1.5561	-0.0009	3.7582	-1.5555	-0.3746	0.3446	-0.1809	0.0142	0.4512	-0.1845	-0.0331
	-1.5992	2.2478	-1.3791	-0.5803	4.2156	-2.8639	-0.1587	0.2616	-0.1563	-0.0588	0.4793	-0.3140
	-0.9321	-0.4623	1.0077	-0.7947	0.1772	3.3837	-0.1212	-0.0271	0.1413	-0.0814	0.0548	0.3967
$K_{e15} =$	5.2465	-2.8441	-1.5906	0.9669	-1.7758	2.3024	0.5997	-0.3064	-0.1691	0.1233	-0.1937	0.2547
	-0.5790	4.5344	1.3450	-1.1733	0.3695	-1.2844	-0.0225	0.5313	0.1557	-0.1080	0.0646	-0.1260
	-0.9108	1.6367	5.7822	-2.7446	-2.2326	1.5825	-0.0857	0.1656	0.6201	-0.2838	-0.2405	0.1750
	1.7561	-0.9091	-0.9212	4.9242	1.2299	-1.1984	0.2235	-0.0745	-0.0563	0.5568	0.1322	-0.1067
	-1.9041	2.9281	-0.6305	1.1777	5.5591	-3.0742	-0.1927	0.3575	-0.0474	0.1241	0.6254	-0.3187
	0.5109	0.1187	2.1077	-1.4255	-0.5266	4.4150	0.0475	0.0555	0.2735	-0.1303	0.0016	0.5073