

N° d'ordre : 1269

50376

1985

51

50376

1985

51

# THÈSE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le grade de

**DOCTEUR DE 3ème CYCLE**

Spécialité Génie Electrique

par

**S. DIMSTER**

**Maître es Sciences**



## **AIDE INFORMATIQUE A LA MODELISATION DE SYSTEMES NON LINEAIRES DANS L'ESPACE D'ETAT**

Soutenue le 10 Juin 1985 devant la Commission d'Examen

Membres du Jury :

G. SEGUIER, Président  
J.P. RICHARD, Rapporteur  
P. BORNE, Examineur  
G. DAUPHIN, Examineur  
F. ROTELLA, Examineur  
G. LEROY, Invité

*A mes parents*

AVANT-PROPOS

Le travail présenté dans ce mémoire a été effectué au Laboratoire d'Automatique et d'Informatique Industrielle de l'Institut Industriel du Nord.

Nous voulons tout d'abord saluer la mémoire du Professeur François LAURENT, Directeur du Laboratoire de Systématique de l'Université de Lille I, qui nous a quittés précocement en juin 1983. Il avait su nous accueillir au sein de son équipe et, par sa gentillesse et sa très grande compétence, nous faire partager son goût profond pour la recherche.

Nous tenons à remercier très vivement Monsieur le Professeur SEQUIER, Directeur du Laboratoire d'Application des Redresseurs de Puissance de l'Université de Lille I, pour le grand honneur qu'il nous a fait en acceptant de présider notre jury de thèse. Qu'il trouve ici le témoignage de notre respectueuse gratitude et de notre profond dévouement.

C'est avec empressement que nous remercions également Monsieur J.P. RICHARD, Docteur ès Sciences, qui a dirigé ce travail. Il a su nous faire profiter de son expérience par sa gentillesse et sa grande compétence. Ses conseils et remarques ont contribué à la réalisation de ce travail. Nous tenons à lui exprimer notre très vive reconnaissance.

Monsieur le Professeur BORNE, Directeur du Laboratoire d'Automatique et d'Informatique Industrielle de l'I.D.N., a suivi avec intérêt l'évolution de notre travail. Nous le remercions très vivement d'avoir bien voulu participer à notre jury de thèse.

Madame G. DAUPHIN-TANGUY, Docteur ès Sciences, a bien voulu juger notre travail et participer au jury de thèse, nous lui transmettons nos très sincères remerciements.

Monsieur F. ROTELLA s'est toujours intéressé à notre travail et nous a apporté de nombreux conseils. Nous le remercions très vivement pour sa collaboration et sa présence dans notre jury de thèse.

Nous sommes très flattée de la présence à notre jury de thèse de Monsieur G. LEROY, Directeur de l'Ecole des Hautes Etudes Industrielles, nous l'en remercions très vivement.

Madame TAILLY et Madame FERRAR se sont chargées de la présentation matérielle de ce mémoire, nous leur adressons nos très vifs remerciements.

Enfin, nous tenons à rendre hommage à tout le personnel du Laboratoire d'Automatique et d'Informatique Industrielle de l'I.D.N., pour l'aide précieuse qu'il nous a apportée tant sur le plan scientifique que sur le plan humain.

SOMMAIRE

# CHAPITRE I : MODÉLISATION ET STABILITÉ.

	<i>Pages</i>
INTRODUCTION	1
I - NOTION DE MODÈLE.	1
I.1 - Généralités	1
I.1.1 - <i>Notion d'état d'un système.</i>	1
I.1.2 - <i>Equations représentatives d'un processus.</i>	2
I.2 - Modélisation des processus étudiés	4
I.2.1 - <i>Problème de la modélisation.</i>	4
I.2.2 - <i>Définition de systèmes à non-linéarités de rang <math>k</math> type <math>L(k), C(k)</math></i>	6
I.3 - Représentations canoniques.	8
I.3.1 - <i>Représentations canoniques sous forme compagnon.</i>	8
I.3.1.1 - <i>Définitions</i>	8
I.3.1.2 - <i>Représentations canoniques associées.</i>	9
I.3.2 - <i>Représentations de Luenberger.</i>	10
I.3.3 - <i>Représentation sous forme en flèche.</i>	11
II - SYSTÈMES ÉTUDIÉS.	12
II.1 - Système de type Lur'e Postnikov mLP, MLP.	12
II.1.1 - <i>Définition</i>	12
II.1.2 - <i>Cas monovariable.</i>	14
II.2 - Représentations canoniques des systèmes étudiés.	16
II.2.1 - <i>Formes compagnons.</i>	16
II.2.2 - <i>Formes en flèche.</i>	17
II.3 - Généralisation - Systèmes à non linéarités parallèles et modélisations associées.	18

III - MÉTHODES D'ÉTUDE DE LA STABILITÉ.	20
III.1 - Définitions.	21
III.2 - Rappels de critères usuels d'étude de la stabilité.	21
III.2.1 - Méthode directe de Lyapunov.	22
III.2.2 - Fonctions quadratiques.	23
III.2.3 - Critère pratique de Borne et Gentina.	24
III.2.4 - Fonctions de type quadratique plus intégrale.	26

## CHAPITRE II : UTILISATION D'UN INVARIANT DE REPRÉSENTATION.

	<i>Pages</i>
INTRODUCTION.	27
I - NOTION D'INVARIANT DE REPRÉSENTATION.	27
I.1 - Définition du polynôme symbolique.	27
I.2 - Invariance du polynôme symbolique.	28
I.3 - Cas d'un système monovarié de type Lur'e Postnikov.	29
I.3.1 - <i>Calcul du polynôme symbolique.</i>	29
I.3.2 - <i>Application à la forme en flèche.</i>	30
II - CALCUL INFORMATIQUE DU POLYNÔME SYMBOLIQUE.	32
II.1 - Introduction.	32
II.2 - Algorithme de calcul.	33
II.2.1 - <i>Décomposition de la matrice.</i>	33
II.2.2 - <i>Calcul du polynôme symbolique.</i>	34
II.2.3 - <i>Exemple.</i>	36
II.3 - Développements informatiques.	37
II.3.1 - <i>Conditionnement et Stabilité.</i>	37
II.3.2 - <i>Méthode utilisée.</i>	38
II.3.3 - <i>Exemple.</i>	38
CONCLUSION.	42

# CHAPITRE III : MODÉLISATION DES SYSTÈMES À L'AIDE DE LA FORME EN FLÈCHE.

Pages

INTRODUCTION.	43
I - MODÉLISATION DES SYSTÈMES MONOVARIABLES DE TYPE LUR'E POSTNIKOV	43
I.1 - Définition du polynôme représentatif.	43
I.2 - Détermination des formes en flèche.	46
I.2.1 - $R(\lambda)$ admet des racines réelles distinctes.	46
I.2.1.1 - Forme matricielle.	46
I.2.1.2 - Cas plus général : le système comporte $n$ non-linéarités.	47
I.2.2 - $R(\lambda)$ admet des racines multiples et des racines complexes.	48
I.2.2.1 - Cas général.	48
I.2.2.1.1 - Forme matricielle	48
I.2.2.1.2 - Calcul des termes non constants.	50
I.2.2.2 - Forme en flèche mince à racines complexes	52
I.2.3 - Application informatique.	55
II - CHOIX DES PARAMÈTRES DE MODÉLISATION.	57
II.1 - Définition d'une problématique	57
II.2 - Résolution algébrique de la problématique P	59
II.2.1 - $\lambda_i$ racines de $D(\lambda)$	60
II.2.2 - $\lambda_i$ racines de $N(\lambda)$	60
II.2.3 - Cas général : $\lambda_i$ racines de $D(\lambda) + aN(\lambda) = 0$	61
II.2.3.1 - Théorème de Sturm.	61
II.2.3.2 - Tableau de Routh modifié.	63
II.2.3.2.1 - Racines du polynôme.	65
II.2.3.3 - Application à la résolution de $D(\lambda) + aN(\lambda) = 0$	65
II.2.3.4 - Exemples	69
II.2.3.4.1 - Tableau de Routh modifié.	69
II.2.3.4.2 - Application à la résolution de $D(\lambda) + aN(\lambda) = 0$ .	70
CONCLUSION	73

# CHAPITRE IV : ANALYSE DES SYSTÈMES À PARTIR DE LA MODÉLISATION EN FLÈCHE.

Pages

INTRODUCTION.	74
I - APPLICATION DE LA FORME EN FLÈCHE AUX SYSTÈMES À DEUX DYNAMIQUES.	75
I.1 - Méthode géométrique de séparation de dynamiques. Cas linéaire .	75
I.2 - Application aux systèmes non linéaires.	78
I.2.1 - Paramètres solutions de $D(\lambda) + aN(\lambda) = 0$	80
I.2.1.2 - Choix de a.	80
I.2.1.2 - Choix des $\lambda_i$ .	82
I.2.2 - Paramètres arbitraires	84
I.2.2.1 - Paramètres complexes.	84
II - APPLICATION DES CRITÈRES USUELS DE STABILITÉ AUX SYSTÈMES MODÉLISÉS PAR UNE FORME EN FLÈCHE.	87
II.1 - Traitement de la forme en flèche mince.	87
II.2 - Traitement de la forme en flèche épaisse.	93
II.2.1 - Blocs $C_j$	94
II.2.2 - Blocs $M_i$	96
II.2.3 - Remarque.	96
II.3 - Traitement informatique.	98
II.3.1 - Tableau récapitulatif.	98
II.3.2 - Réalisation informatique.	100
II.3.2.1 - Déterminant d'une forme en flèche mince.	101
II.3.2.2 - Déterminant d'une forme en flèche épaisse.	102
II.4 - Exemple.	105
CONCLUSION.	108
ANNEXE	109

CHAPITRE V : DESCRIPTION DU PROGRAMME D'AIDE À LA  
 MODÉLISATION ET A L'ANALYSE DES SYSTÈMES  
 NON-LINÉAIRES DE TYPE LUR'E POSTNIKOV.

	<i>Pages</i>
INTRODUCTION.	113
I - PRÉSENTATION DU PROGRAMME.	113
I.1 - La structure du programme.	113
I.2 - Aspect conversationnel du programme. Mode d'emploi.	119
I.2.1 - <i>Données et résultats.</i>	119
I.2.2 - <i>Messages d'erreur.</i>	132
I.2.3 - <i>Directives.</i>	132
II - DESCRIPTION DU PROGRAMME.	134
II.1 - Le programme principal.	134
II.2 - Les sous-programmes primaires.	138
II.2.1 - <i>SP de définition du système.</i>	138
II.2.2 - <i>SP de choix de paramètres de modélisation.</i>	140
II.2.3 - <i>SP de modélisation du système.</i>	142
II.2.4 - <i>SP d'étude de la stabilité du système.</i>	143
II.2.5 - <i>SP d'aide.</i>	144
II.3 - Les sous-programmes secondaires.	145
II.3.1 - <i>Les sous-programmes de calcul.</i>	145
II.3.1.1 - Définition du système.	145
II.3.1.2 - Choix des paramètres du système.	148
II.3.1.2.1 - Résolution d'une équation polynômiale	148
II.3.1.2.2 - Résolution de l'équation $D + aN = 0.$	149
II.3.1.2.3 - Sous-programmes appelés par le SPS ROUTH	153
II.3.1.3 - Modélisation du système.	157
II.3.1.4 - Etude de la stabilité.	162
II.3.1.4.1 - Critère de Borne et Gentina.	162
II.3.1.4.2 - Définition d'un modèle réduit.	167

II.3.2 - Les sous-programmes "édition des résultats".	168
II.3.2.1 - Définition du système.	168
II.3.2.2 - Paramètres du système.	172
II.3.2.3 - Modélisation du système.	173
II.4 - Récapitulatif.	174
III - EXEMPLES.	178
CONCLUSION.	191
ANNEXE 1	192
ANNEXE 2	209

INTRODUCTION GÉNÉRALE.

Dans ce mémoire, est présenté un programme d'aide à la modélisation et à l'analyse des systèmes non linéaires, et en particulier, des systèmes monovariables du type Lur'e Postnikov.

Les recherches menées sur ces systèmes ont contribué à la définition de différents modèles de représentation et à l'élaboration de nombreux critères d'étude de la stabilité ou de la séparation de dynamiques. Très souvent, ces critères sont essentiellement liés au choix du modèle.

En effet, l'étude des systèmes non linéaires pose, encore plus que dans le cas linéaire, le problème du choix de la représentation.

Si, dans le cas de systèmes linéaires, le conditionnement d'une matrice s'avère déjà très important pour la convergence des algorithmes d'analyse numérique, il vient se greffer dans le cas d'un système asservi à paramètres non constants, un problème supplémentaire : les résultats obtenus diffèrent selon qu'un même critère (de stabilité, par exemple) est appliqué à une modélisation plutôt qu'à une autre.

De plus, deux critères différents peuvent conduire à des conditions (suffisantes) de stabilité différentes. Il n'existe donc pas de méthode générale déterminant le meilleur modèle à choisir afin d'obtenir une synthèse optimale du système, d'où l'intérêt de définir plusieurs modèles du processus et d'essayer sur chacun, les critères d'étude de stabilité existants.

Pour des systèmes d'ordre élevé, une telle démarche est grandement facilitée par l'utilisation d'un programme fournissant les différents modèles de représentation et les conditions de stabilité associées.

Le premier chapitre présente les différents modèles de représentation des systèmes monovariabiles non linéaires de type Lur'e Postnikov ainsi que divers critères de stabilité.

Le deuxième chapitre présente un invariant de représentation utilisable dans la modélisation des systèmes étudiés. Le calcul informatique de cet invariant, appelé polynôme symbolique, y est détaillé.

Dans le troisième chapitre, on montre comment, à partir de cet invariant, il est possible d'obtenir une représentation matricielle particulière : la forme en flèche. Différentes méthodes de choix des coefficients diagonaux de cette matrice sont rappelées et programmées.

L'application de critères de stabilité sur la forme en flèche représentative d'un système est faite dans le quatrième chapitre. Une application du programme de modélisation en vue de l'étude de séparabilité des dynamiques y est également proposée.

Enfin, le dernier chapitre fournit une description du programme LIMA (Logiciel Interactif de Modélisation et d'Analyse) et de nombreux exemples d'utilisation.

o  
o o

CHAPITRE I

MODELISATION ET STABILITE

## INTRODUCTION.

Cette première partie définit la classe de systèmes étudiés.

Différents types de représentations matricielles dans l'espace d'état  $y$  sont présentés et en particulier une représentation canonique importante : la forme dite en flèche /BENREJEB 1980/, /GROUMPOS, SCOTT 1977/.

Le choix de la représentation d'un système non linéaire est déterminant lors de son analyse. En effet les conditions de stabilité obtenues dépendent de la modélisation choisie et du critère de stabilité utilisé.

En association avec les diverses formes matricielles présentées, les principaux critères relatifs à l'étude de la stabilité sont énoncés.

## I - NOTION DE MODELE.

### I.1 - Généralités.

#### I.1.1 - *Notion d'état d'un système.*

L'étude proposée concerne la représentation dans l'espace d'état de processus dynamiques continus et déterministes. Leur évolution considérée à partir d'un instant  $t_0$  dans l'intervalle de temps  $T$ ,  $T = [t_0, +\infty[$ , est décrite à partir de trois ensembles de variables :

- un ensemble  $u(t)$  de grandeurs d'entrée qui représentent les actions de l'extérieur sur le système.

- un ensemble  $y(t)$  de grandeurs de sortie accessibles.

La détermination mathématique de  $y(t)$  nécessite la connaissance d'un modèle du système, de ses entrées et d'un certain nombre d'informations à l'instant  $t_0$ . Ces informations constituent le troisième ensemble :

- l'état, ensemble de  $q$  informations dépendant de l'évolution du système antérieure à l'instant  $t$ .

Ces  $q$  variables, appelées variables d'état sont regroupées sous forme d'un vecteur état  $x(t)$  dont la dimension  $q$  détermine l'ordre de la représentation choisie pour le processus. Lorsque les grandeurs d'entrée  $u(t)$  sont connues pour  $t \geq t_0$ , la connaissance du vecteur état  $x(t)$  suffit à déterminer le comportement du système  $\forall t \geq t_0$ .

### 1.1.2. - Equations représentatives d'un processus.

L'application des lois de la physique conduit à différentes formulations mathématiques représentatives du système : équations différentielles, équations aux dérivées partielles ...

Nous supposerons dans ce mémoire que la modélisation des processus envisagés, peut être réalisée de façon à ce que, en réponse à une entrée  $u(t)$  dans un intervalle de temps  $T$ , leur comportement soit décrit par une équation vectorielle d'état (I.1).

$$(I.1) \quad \begin{cases} \dot{x}(t) = A(x, u(t), t) x(t) + B(x, u(t), t) u(t) \\ y(t) = C(x, u(t), t) x(t) + D(x, u(t), t) u(t) \end{cases}$$

où  $\forall t \in T$

$$\begin{array}{ll} x(t) & \text{vecteur état,} \quad x(t) \in \mathbb{R}^q, \quad x^T(t) = \{x_1(t), \dots, x_q(t)\} \\ u(t) & \text{vecteur des entrées,} \quad u(t) \in \mathbb{R}^n, \quad u^T(t) = \{u_1(t), \dots, u_n(t)\} \\ y(t) & \text{vecteur des sorties,} \quad y(t) \in \mathbb{R}^p, \quad y^T(t) = \{y_1(t), \dots, y_p(t)\} \end{array}$$

et

$A(x, u(t), t)$  matrice d'évolution du système

$$A \in \mathbb{R}^{q \times q}$$

$B(x, u(t), t)$  matrice de commande

$$B \in \mathbb{R}^{q \times n}$$

$C(x, u(t), t)$  matrice d'observation

$$C \in \mathbb{R}^{p \times q}$$

$D(x, u(t), t)$  matrice de transmission directe

$$D \in \mathbb{R}^{p \times n}$$

Lorsque aucune ambiguïté n'est possible, nous noterons quelquefois :

$$A(x, u, t) = A(\cdot)$$

$$B(x, u, t) = B(\cdot)$$

$$C(x, u, t) = C(\cdot)$$

$$D(x, u, t) = D(\cdot)$$

Différents cas sont alors envisageables : le système est

- non linéaire non stationnaire : A, B, C, D dépendent de x, u, t,
- linéaire non stationnaire : A, B, C, D dépendent de t,
- non linéaire non stationnaire : A, B, C, D dépendent de x, u,
- linéaire stationnaire : A, B, C, D sont des constantes.

Le système défini en (I.1) est représenté par le schéma de la figure I.1.

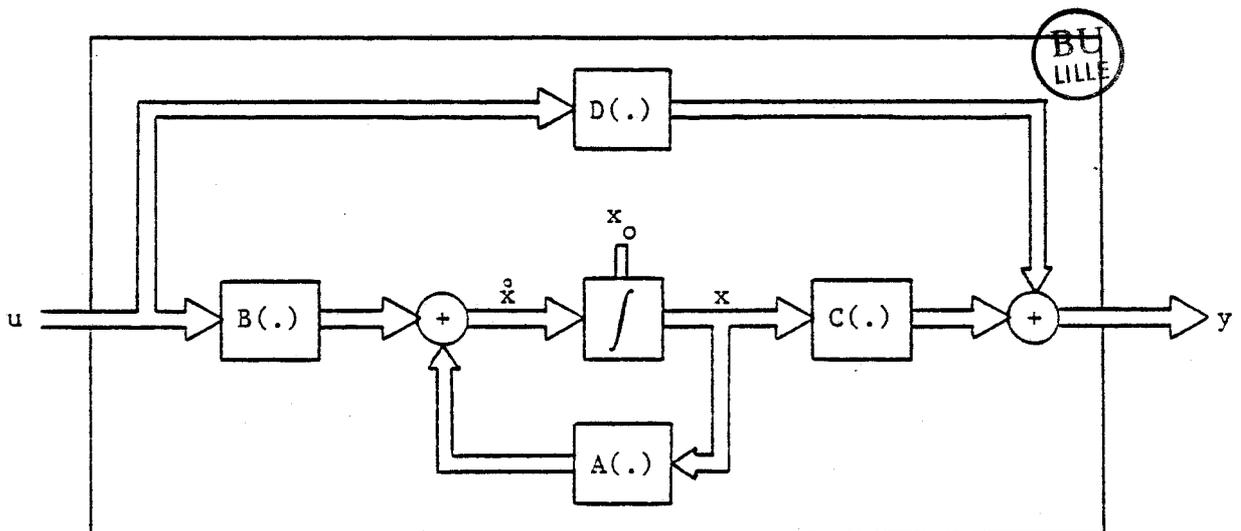


Fig. I.1.

La représentation (I.1) du système n'est pas unique. Le choix d'un autre vecteur état  $w(t)$  peut conduire à une représentation d'état différente qui simplifiera par exemple l'analyse du système.

Un cas intéressant est celui où le nouvel état  $w(t)$  est défini par :

$$(I.2) \quad w(t) = P x(t)$$

La matrice  $P$  peut être une matrice dépendant du temps et vérifiant les propriétés d'une transformation de Ljapunov /GANTMACHER 1966/, /MARKUS 1955/ :  
si  $P(t), \dot{P}(t)$  continues bornées,  $P \in \mathbb{R}^q \times q$ , il existe  $c \in \mathbb{R}^+$  tel que  
 $\forall t |\det P(t)| \geq c > 0$ .

Si la matrice  $P$  est constante et régulière, la transformation (I.2) correspond à un changement de base dans l'espace d'état. C'est ce dernier cas qui sera considéré dans cette étude.

De (I.2), il vient la nouvelle représentation d'état :

$$(I.3) \quad \begin{cases} \dot{w} = P A(P^{-1}w, u(t), t) P^{-1} w(t) + P B(P^{-1}w, u(t), t) u(t) \\ y = C(P^{-1}w, u(t), t) P^{-1} w(t) + D(P^{-1}w, u(t), t) u(t) \end{cases}$$

Cette relation est du même type que (I.1) mais fait intervenir des matrices différentes et un vecteur état différent.

## I.2 - Modélisation des processus étudiés.

### I.2.1. - Problème de la modélisation.

S'il existe une infinité de représentation  $(A, B, C, D)$  (I.1) pour un même système, le choix de la matrice  $P$  est souvent orienté de façon à obtenir une matrice  $P A P^{-1}$  présentée en (I.3) permettant une analyse performante des propriétés du système (stabilité, décomposition en sous-systèmes).

Ainsi, trois objectifs liés sont à considérer :

- obtenir un partitionnement de la matrice en blocs matriciels.

Ce partitionnement souvent réalisé en analyse numérique /GANTMACHER 1966/, /ROSENBROCK & STOREY 1970/, /DEIF 1982/ peut être utilisé lors d'une décomposition d'un système global en sous-systèmes interconnectés /LUENBERGER 1967/, /BENREJEB 1980/, /FOSSARD 1972/, /TITLI 1975/, ou bien sur les processus à perturbations singulières /FOSSARD 1982/, /DAUPHIN-TANGUY 1983/ ou encore sur l'étude de stabilité à partir de normes vectorielles /GENTINA, BORNE & LAURENT 1972/.

- Réduire le nombre de coefficients non nuls.

Ceci peut être intéressant pour la programmation de la matrice, l'identification du système. Ainsi a été introduite la notion de matrice creuse (comportant peu de termes non nuls) ou dense (comportant peu de termes nuls) /LAURENT 1968/, /BENREJEB 1980/, /DEIF 1982/.

- Réduire le nombre de termes non constants dans la matrice d'évolution afin d'obtenir une application plus aisée de critères de stabilité /ROSENBROCK 1965/, /BORNE, GENTINA, LAURENT 1972/, / LAURENT, ELMOUDNI, RICHARD BORNE 1979/ ou donner des schémas de simulation plus performants /BENREJEB, DAUPHIN, BORNE 1980/.

Se fixant ces trois objectifs, divers auteurs ont proposé un certain nombre de représentations types appelées formes canoniques /FOSSARD 1972/, /PRATZEL WOLTERS 1983/, /HINRICHSSEN 1984/.

La plupart de ces formes canoniques peuvent être obtenues par programme en utilisant les algorithmes présentés par exemple dans /BARRAUD 1975/ ou /HICKIN, SINHA 1977/.

Parmi ces formes canoniques et dans le cas de systèmes linéaires stationnaires citons la forme diagonale, les formes de Jordan et les représentations directement liées au polynôme caractéristique de la matrice : formes compagnons, formes en flèche.

Ces deux dernières formes matricielles seront présentées dans le paragraphe (I.3).

Nous allons tout d'abord définir une classe de systèmes non linéaires non stationnaires dits "*à non linéarités de rang k*".

I.2.2. - Définition de systèmes à non linéarités de rang  $k$  type  $L(k)$ ,  $C(k)$ .

Afin de réduire le nombre de termes constants, ou de regrouper les termes non constants, /RICHARD 1981, 1984/ et /ROTELLA 1983/ ont introduit la notion de non-linéarités de rang  $k$  présentant la définition suivante :

Définition.

Une matrice  $A(x,u,t)$  (notée  $A(\cdot)$ ) de (I.1) de  $\mathbb{R}^{q \times q}$  est à non linéarité de rang  $k$  si elle est semblable à une matrice  $F(x,u,t)$  ( $F(\cdot)$ ) dont les termes non constants sont regroupés dans  $k$  rangées de même nature (lignes ou colonnes).

Si le regroupement est en colonnes, la matrice est dite de type  $C(k)$  et s'il est en lignes de type  $L(k)$ .

Une matrice de type  $C(k)$  s'écrit donc :

$$(I.4) \quad \begin{aligned} A(\cdot) &= A + V(\cdot) H^T \\ A &\in \mathbb{R}^{q \times q} \\ V(\cdot) &\in \mathbb{R}^{q \times k} \\ H &\in \mathbb{R}^{q \times k} \end{aligned}$$

Lorsque la base de l'espace est judicieusement choisie,  $H$  peut s'écrire sous la forme :

$$H^T = \left[ \begin{array}{cc} \boxed{0} & \boxed{\text{diagonales}} \\ \longleftarrow q-k & \longleftarrow k \end{array} \right]$$

Les  $k$  dernières colonnes de  $A(\cdot)$  sont alors les seules à contenir des termes non constants.

Le rang de la matrice  $V(\cdot) H^T$  correspondant à un regroupement de tous les termes non constants de  $A(\cdot)$  est inférieur ou égal à  $k$  à chaque instant.

De même, une matrice de type L(k) s'écrit :

$$(I.5) \quad \begin{aligned} A(.) &= A + K V^T(.) \\ A &\in \mathbb{R}^{q \times q} \\ K &\in \mathbb{R}^{q \times k} \\ V(.) &\in \mathbb{R}^{q \times k} \end{aligned}$$

Cette écriture s'obtient par transposition de (I.4). Les k dernières lignes de A(.) peuvent alors être les seules à contenir des termes non constants.

Les définitions suivantes ont également été proposées /RICHARD 1984/.

### Définitions.

1°) - Le système décrit par les équations (I.6)

$$(I.6) \quad \begin{cases} \dot{x}(t) = A x(t) + V(.) \varepsilon(t) \\ \varepsilon(t) = u(t) + H^T x(t) \\ \text{avec } A, V(.), H \text{ définies en (I.4)} \\ x \in \mathbb{R}^q, u \text{ et } \varepsilon \in \mathbb{R}^k \end{cases}$$

est dit à non-linéarités de rang k, de type C(k).

2°) - Le système décrit par les équations (I.7).

$$(I.7) \quad \begin{cases} \dot{x}(t) = A x(t) + K \varepsilon(t) \\ \varepsilon(t) = u(t) + V^T(.) x(t) \\ \text{avec } A, V(.), K \text{ définies en (I.5)} \\ x \in \mathbb{R}^q, u \text{ et } \varepsilon \in \mathbb{R}^k \end{cases}$$

est dit à non-linéarités de rang k de type L(k).

On étudiera principalement les systèmes de type C(k) puisqu'il est facile d'en déduire la forme L(k). En régime autonome (u(t) entrée nulle) les équations (I.6) et (I.7) s'écrivent :

$$(I.8) \quad \begin{cases} \dot{x}(t) = (A + v(.) H^T) x(t) \\ \varepsilon(t) = H^T x(t) \end{cases}$$

et

$$(I.9) \quad \begin{cases} \dot{x}(t) = (A + K V^T(.)) x(t) \\ \varepsilon(t) = V^T(.) x(t) \end{cases}$$

Un cas particulier de systèmes est celui pour lequel le rang des non-linéarités est égal à 1. Les équations (I.6) et (I.7) deviennent alors respectivement (I.10) et (I.11)

$$(I.10) \quad \begin{cases} \dot{x}(t) = A x(t) + v(.) \varepsilon(t) \\ \varepsilon(t) = u(t) + h^T x(t) \end{cases} \quad \text{pour le type C(1)}$$

$A \in \mathbb{R}^q \times q$ ,  $v(.) \in \mathbb{R}^q$ ,  $h \in \mathbb{R}^q$  et  $A(.) = A + v(.) h^T$

$A(.)$  présente une seule colonne à termes non-constants si  $h = (0 \dots 0, 1)^T$ .

$$(I.11) \quad \begin{cases} \dot{x}(t) = A x(t) + k \varepsilon(t) \\ \varepsilon(t) = u(t) + v^T(.) x(t) \end{cases} \quad \text{pour le type L(1)}$$

$A \in \mathbb{R}^q \times q$ ,  $v(.) \in \mathbb{R}^q$ ,  $k \in \mathbb{R}^q$  et  $A(.) = A + k v^T(.)$

$A(.)$  présente une seule ligne à termes non constants si  $k = (0 \dots 0, 1)^T$ .

Dans le paragraphe suivant sont présentées différentes formes canoniques pouvant être regroupées dans la classe L(k) ou C(k).

### I.3 - Représentations canoniques.

#### I.3.1. - Représentations canoniques sous forme compagnon.

##### I.3.1.1. - Définitions.

Une matrice de forme compagnon /ROSENBRÖCK 1970/ ou Frobenius /HINRICHSEN 1984/ est une matrice définie par (I.12) ou (I.13) où  $X$  est un élément non constant.

(I.12)

$$\begin{bmatrix} 0 & 0 & \text{---} & 0 & x \\ 1 & 0 & \text{---} & 0 & x \\ 0 & 1 & \text{---} & 0 & x \\ \vdots & \vdots & \text{---} & \vdots & \vdots \\ 0 & 0 & \text{---} & 1 & x \end{bmatrix}$$

Type C(1).

(I.13)

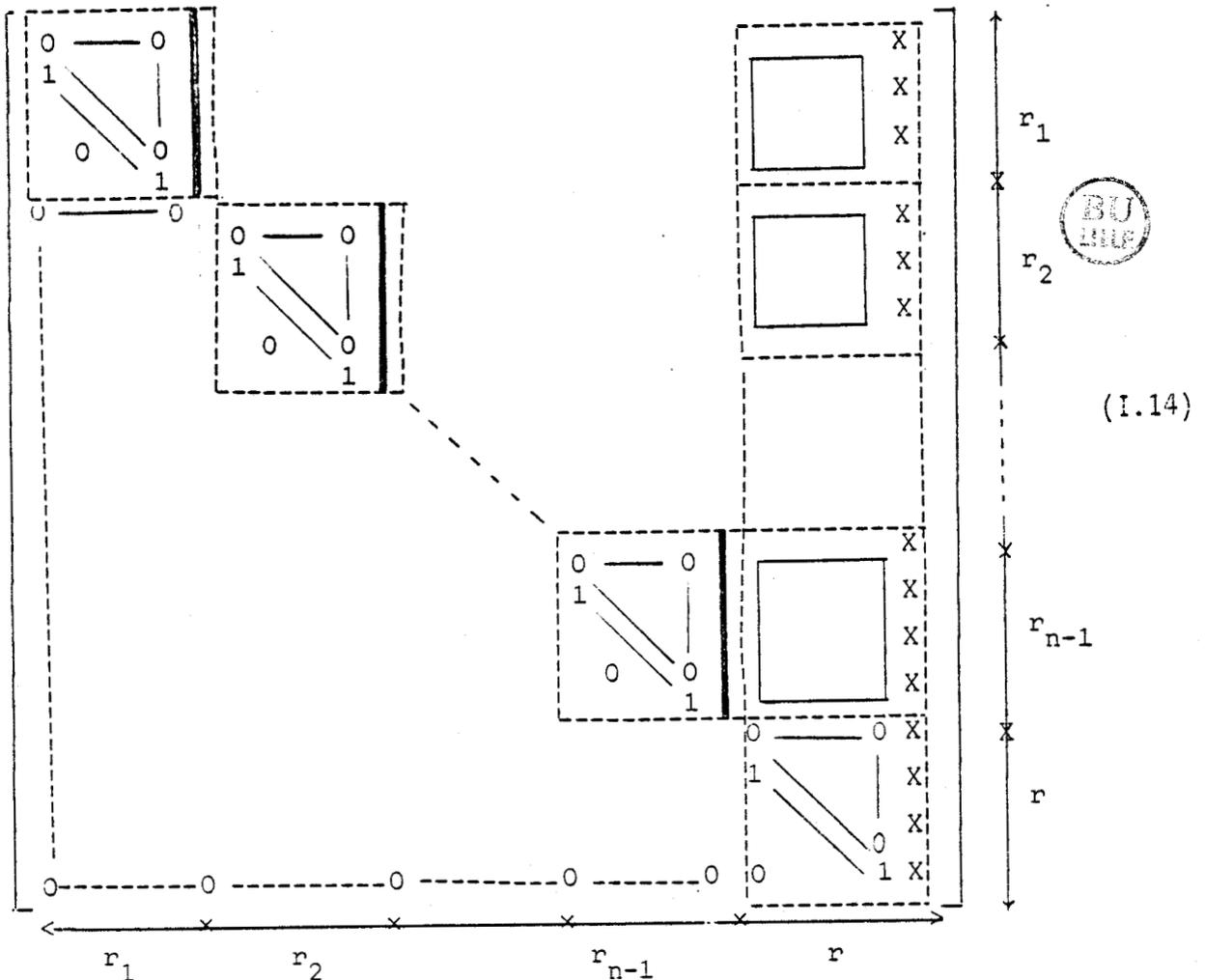
$$\begin{bmatrix} 0 & 1 & \text{---} & 0 \\ 0 & 0 & 1 & \text{---} & 0 \\ \vdots & \vdots & \vdots & \text{---} & \vdots \\ x & x & x & \text{---} & x \end{bmatrix}$$

Type L(1).

A partir de ces matrices, deux autres formes compagnons peuvent être obtenues en effectuant une symétrie suivant la diagonale secondaire.

1.3.1.2. - Représentations canoniques associées.

On se réfère dans ce paragraphe à /RICHARD 1981/. En régime autonome (I.8) ou (I.9), l'évolution du système peut être représentée par une matrice à non linéarités de rang 1 de type C(1) définie en (I.14).



avec  $r + \sum_{i=1}^{n-1} r_i = q.$

Les traits pleins représentent des coefficients constants et les croix des termes non constants qui sont tous situés dans la dernière colonne. Il est possible par changement de base d'obtenir la matrice  $L_1(.) = C_1^T(.)$ .

D'autres résultats concernant le régime non autonome ont été présentés dans /ROTELLA 1983/, /RICHARD 1984/ ainsi qu'une généralisation aux systèmes à non-linéarités de rang  $k$ .

Les matrices d'évolution de type  $C(k)$  et  $L(k)$  sont comme  $C_1(.)$  et  $L_1(.)$  des matrices triangulaires et comportent respectivement  $k$  colonnes ou  $k$  lignes non constantes.

1.3.2. - Représentations canoniques de Luenberger.

Divers auteurs ont montré que certaines formes de représentations sont bien adaptées à l'étude de la commandabilité et de l'observabilité des systèmes linéaires multivariés. Ainsi, /DESCUSSE 1981/ a souligné l'importance de la forme canonique dite de Luenberger-Brunovsky /LUENBERGER 1967/, /BRUNOVSKY 1970/ dans la détermination de la commandabilité du système.

Cette forme se présente comme suit :

The diagram illustrates the Luenberger-Brunovsky canonical form, consisting of three vertically stacked blocks. Each block is a square matrix with a specific structure of zeros, ones, and 'X' terms. The top block has a top row of 0, 1, 0, and a blank space followed by 0. Below this, there are three parallel diagonal lines sloping downwards from left to right. The bottom row of the top block contains 0, a blank space, 0, and 1. The middle block has a top row of 0, 1, 0, and a blank space followed by 0. Below this, there are two parallel diagonal lines sloping downwards from left to right. The bottom row of the middle block contains 0, a blank space, 0, and 1. The bottom block has a top row of 0, 1, 0, and a blank space followed by 0. Below this, there is one parallel diagonal line sloping downwards from left to right. The bottom row of the bottom block contains 0, a blank space, 0, and 1. Horizontal lines connect the 'X' terms in the bottom rows of the blocks. The first block has 'X X X' followed by a horizontal line to 'X'. The second block has 'X X X' followed by a horizontal line to 'X'. The third block has 'X X' followed by a horizontal line to 'X'. A long horizontal line at the bottom connects the 'X' of the first block to the 'X' of the third block. The entire structure is enclosed in large square brackets on the left and right sides. To the right of the brackets, the label (I.15) is present.



Différentes classifications de formes en flèche et de leurs propriétés sont énoncées dans /BENREJEB 1980/.

Dans le cas particulier où les  $A_{ij}$  sont réduites à des scalaires,  $F(.)$  (I.16) est une *forme en flèche mince*.

Dans le cas le plus général, la matrice  $F(.)$  est appelée *forme en flèche épaisse*.

Elle permet de représenter n'importe quel système défini par les équations (I.6) ou (I.7).

Elle est particulièrement adaptée à l'étude des systèmes monovariabiles de type Lur'e Postnikov dont nous rappelons la définition dans la partie suivante.

## II - SYSTEMES ETUDIES.

Ce travail concerne tout particulièrement les systèmes non linéaires du type Lur'e Postnikov /LUR'E & POSTNIKOV 1944/, /GRUJIC 1978/.

Ceci ne constitue cependant pas une limitation dans l'étude des systèmes non linéaires.

En effet, quelque soit le système non linéaire à étudier, on essaie généralement de se ramener à ce cas particulier de systèmes non linéaires dont les propriétés sont bien connues.

### II.1 - Systèmes de type Lur'e Postnikov mLP et MLP.

#### II.1.1. - Définition.

Les systèmes de type Lur'e Postnikov multivariabiles (*en abrégé MLP*) sont décrits par les équations générales suivantes :

$$(I.17) \quad \begin{cases} \dot{x}(t) = A(t) x(t) + B(t) f(y,t) \\ y(t) = C(t) x(t) + u(t) \end{cases}$$

Les dimensions sont les mêmes qu'en (I.6) et (I.7). La fonction  $f$  vérifie les propriétés suivantes :

$$(I.18) \quad \left\{ \begin{array}{l} f : \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \mathbb{R}^n \\ f(y,t) = (\phi_i(y,t))_{i=1,\dots,n} ; f_i(0,t) = 0 \quad \forall t \in \mathbb{R}_+ \\ F^*(y,t) = \text{diag} \left\{ \frac{\phi_i(y,t)}{\sigma_i} \right\} = \text{diag}(\phi_i^*(y,t)) \in \mathbb{R}^{n \times n} \\ \forall (y,t) \in \mathbb{R}^n \times \mathbb{R}_+, \exists F_n \in \mathbb{R}^{n \times n}, \exists F_M \in \mathbb{R}^{n \times n}, F_m \leq F^*(y,t) \leq F_M \end{array} \right.$$

Les termes  $\phi_i^*(.)$  sont appelés "gains équivalents" du système.

(I.17) devient alors :

$$(I.19) \quad \left\{ \begin{array}{l} \dot{x}(t) = (A(t) + B(t) F^*(y,t) C(t)) x(t) + B(t) F^*(y,t) u(t) \\ y(t) = C(t) x(t) + u(t). \end{array} \right.$$

et en régime libre, on obtient :

$$(I.20) \quad \dot{x}(t) = (A(t) + B(t) F^*(Cx,t) C(t)) x(t)$$

et la matrice d'évolution s'écrit :

$$(I.21) \quad A(x,t) = A(t) + B(t) F^*(Cx,t) C(t).$$

La structure correspondant au système (I.17) est décrite (fig. I.2).

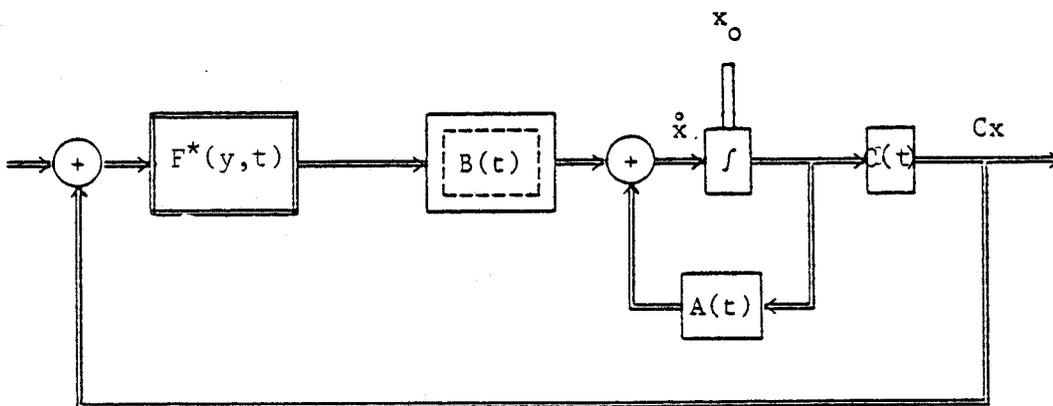


Fig. I.2. Système de type Lur'e Postnikov multivariable (MLP).

### II.1.2. - Cas monovariabile (mLP).

Lorsque la matrice  $F^*$  peut se réduire à un scalaire le système MLP est appelé système monovariabile, de type Lur'e Postnikov (en algèbre mLP) décrit par le système (I.22).

$$(I.22) \quad \begin{cases} \dot{x}(t) = A(t) x(t) + b(t) f(\varepsilon, t) \\ \varepsilon(t) = c^T x(t) + u(t) \end{cases}$$

où

$$A \in \mathbb{R}^{q \times q} \quad b, c \in \mathbb{R}^q \\ \varepsilon \in \mathbb{R}, u \in \mathbb{R}, x \in \mathbb{R}^q$$

On suppose que  $f$  vérifie (I.18) et qu'il existe donc une fonction bornée  $f^* : \mathbb{R} \rightarrow \mathbb{R}$  appelée gain équivalent telle que :

$$\forall \varepsilon \in \mathbb{R} \\ \forall x \in \mathbb{R}^q, \forall t \in T, f(\varepsilon, t) = f^*(\varepsilon, t) \cdot \varepsilon.$$

Le système se met alors en équation comme suit :

$$(I.23) \quad \begin{cases} \dot{x}(t) = (A(t) + b(t) \cdot f^*(\varepsilon, t) \cdot c^T) x(t) + b(t) f^*(\varepsilon, t) u(t) \\ \varepsilon(t) = c^T x(t) + u(t) \end{cases}$$

et en régime libre, la matrice d'évolution s'écrit :

$$(I.24) \quad A(x, t) = A(t) + b(t) f^*(\varepsilon, t) c^T(t)$$

La structure correspondant au système (I.22) est décrite fig (I.3)

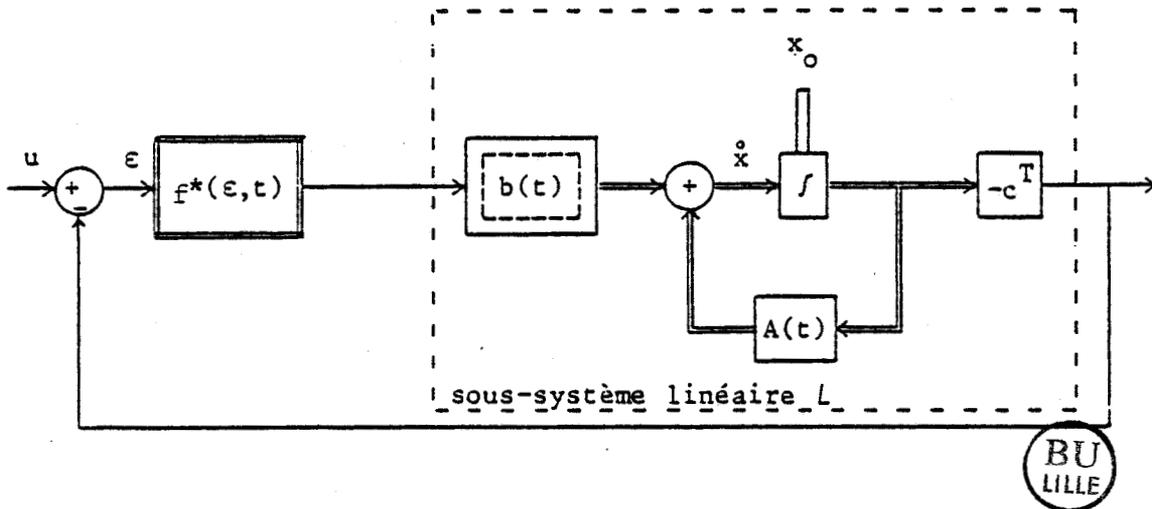


Fig. I.3. Système monovariable type Lur'e Postnikov (mLP).

Ce schéma fait apparaître un sous-système à entrée et sortie scalaires bouclé par une rétroaction non linéaire correspondant à la variable de commande. Une classe particulière de ces représentations correspond au cas où  $A(t)$ ,  $b(t)$ ,  $c(t)$  sont des constantes :

$$A(t) = A, \quad b(t) = b, \quad c(t) = c \quad \forall t \in \mathbb{R}$$

Le sous-système est alors linéaire stationnaire.

De nombreux processus peuvent être représentés à l'aide du schéma bloc suivant (figure I.4).

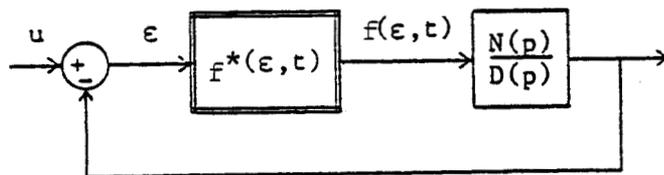


Fig. I.4.

où

$$\frac{N(p)}{D(p)} = -c^T (pI - A)^{-1} b$$

$\frac{N(p)}{D(p)}$  est appelée *fonction de transfert* et est telle que le degré de  $N(p)$  soit inférieur au degré de  $D(p)$ .

Elle représente un bloc linéaire commandé par une non linéarité scalaire asservi par un retour unitaire.

## II.2 - Représentations canoniques des systèmes étudiés.

Dans cette partie, nous nous proposons d'adapter des résultats concernant les représentations matricielles canoniques (§ I.3) aux systèmes monovariabiles de type Lur'e Postnikov (mLP) décrits par (I.22).

### II.2.1. - Formes compagnons.

Comme nous l'avons précisé dans le paragraphe I.3.1, les systèmes non linéaires à non linéarités de rang 1 peuvent être représentés par une matrice triangulaire comportant une seule colonne ou ligne non constante (I.14).

De précédents travaux /RICHARD 1981/ ont cependant étudié la possibilité de se ramener dans le cas des systèmes de type mLP à une matrice d'évolution réduite à une seule forme compagnon. Dans le cas où la fonction de transfert est non dégénérée /POPOV 1973/, les systèmes décrits par (I.22) peuvent être représentés par l'équation d'état :

$$\dot{x}(t) = A(.) x(t) + b(.) u(t)$$

où

$$(I.25) \left\{ \begin{array}{l} \forall t \in [t_0, +\infty[ \\ x(t) \in \mathbb{R}^q, \text{ vecteur état} \\ e(t) \in \mathbb{R}, \text{ entrée scalaire du système} \\ A(.) = \begin{bmatrix} 0 & \text{---} & 0 & -a_0 & -b_0 f^*(.) \\ 1 & \diagdown & & & \\ 0 & \diagdown & & & \\ \vdots & \diagdown & & & \\ 0 & \text{---} & 0 & 1 & -a_{q-1} & -b_{q-1} f^*(.) \end{bmatrix} \\ \text{matrice de type C(1)} \\ b^T(.) = (b f^*(.), \dots, b_{q-1} f^*(.)) \end{array} \right.$$

où

$$\left\{ \begin{array}{l} f^*(.) \text{ fonction réelle non constante dépendant de l'état et} \\ \text{du temps} \\ \forall i \in \{0, \dots, q-1\} (a_i, b_i) \in \mathbb{R}^2 \\ \text{coefficients réels du dénominateur } D(p) \text{ et } N(p) \text{ de la} \\ \text{fonction de transfert} \end{array} \right.$$

La modélisation proposée est de type C(1). Il est toutefois possible de mener une autre étude sur une matrice de type L(1) déduite de (I.25) par changement de base.

### II.2.2. - Formes en flèche.

Des travaux antérieurs /BENREJEB 1980/, /ROTELLA 1983/ ont montré que la matrice d'évolution de l'équation d'état (I.23) désignant un système (mLP) peut se mettre sous la forme d'une matrice en flèche (I.16) dont la forme la plus générale est la suivante :

$$\dot{x} = F(.) x(t)$$

où  $x(t)$  vecteur état

$$(I.26) \quad F(.) = \begin{bmatrix} \boxed{F_{11}} & & 0 & & \boxed{F_{1r}^*} \\ & & & & | \\ & & & \boxed{F_{r-1 \ r-1}} & \boxed{F_{r-1 \ r}^*} \\ & & & & | \\ 1 & \text{---} & 1 & \text{---} & 1 & \text{---} & 1 & & F_{rr}^* \end{bmatrix}$$

où

$$\left\{ \begin{array}{l} \forall i \in \{1, \dots, r-1\} \quad F_{ii} \in \mathbb{R}^{n_i \times n_i} \\ \\ F_{ir}^* \in \mathbb{R}^{n_i}, \quad F_{rr}^* \in \mathbb{R}. \\ \\ \text{avec } \sum_{i=1}^r n_i = q \end{array} \right.$$

Les matrices  $F_{ii}$  sont des matrices à éléments constants contrairement aux vecteurs  $F_{ir}^*$  qui comportent des éléments non constants dépendant de  $f^*$ .  $F(.)$  est une matrice de type  $C(1)$ .

Le principal avantage d'une telle forme est que les blocs constants  $F_{ii}$  de la diagonale peuvent être choisis arbitrairement.

Il est souvent avantageux de les déterminer en fonction des pôles et des zéros de la fonction de transfert  $\frac{N(p)}{D(p)}$  /BENREJEB 1976/, /ROTELLA 1983/, /RICHARD 1984/, /DAUPHIN-TANGUY 1983/.

Les formulations de  $F(.)$  qui peuvent ainsi être obtenues seront présentées ultérieurement dans le troisième chapitre.

### II.3 - Généralisation.

*Systèmes à non-linéarités parallèles et modélisations associées.*

Une étude de la modélisation d'un système monovariable constitué de différents systèmes non linéaires de type (mLP) mis en parallèle est également envisageable à partir de la forme en flèche.

Le système étudié est alors représenté par le schéma bloc de la figure I.5.

avec  $\forall i \in \{1, \dots, n\}$

$\frac{N_i(p)}{D_i(p)}$  fonction de transfert du bloc linéaire  $S_i$

$d^\circ N_i(p) < d^\circ D_i(p)$

$f_i$  : fonctions bornées de  $\mathbb{R} \rightarrow \mathbb{R}$  telles que  $f_i(\varepsilon) = f_i^*(\varepsilon) \cdot \varepsilon$

Les résultats présentés dans la partie II.2 concernant les systèmes du type mLP sont alors généralisés au système décrit par la figure I.5.

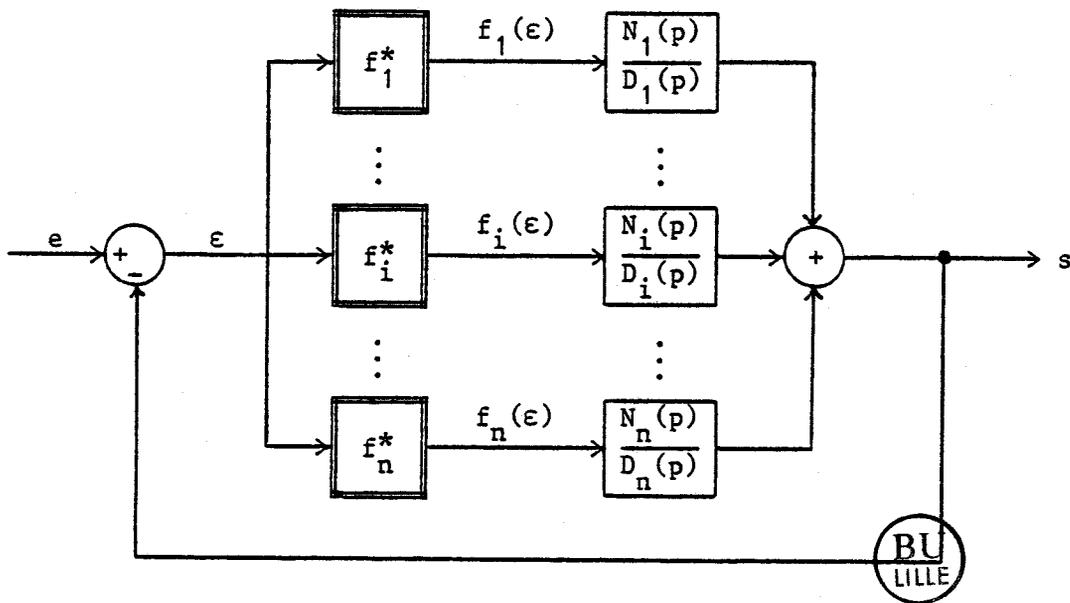


Fig. I.5.

On obtient alors le même type de représentations, où les termes non constants qui dépendent de  $f^*$  (I.25), (I.26) dans le cas d'un seul système non linéaire, sont remplacés par une combinaison linéaire  $g_i^*$  des fonctions  $f_i^*$ . Soit  $A(\cdot)$ , une matrice décrivant le système réalisé (fig. 5), en choisissant  $\varepsilon$  comme dernière composante du vecteur état, les non linéarités sont regroupés dans la dernière colonne et  $A(\cdot)$  est du type C(1). La décomposition canonique fait alors intervenir une matrice compagnon non constante définie par :

$$(I.27) \quad A(\cdot) = \begin{bmatrix} 0 & \text{---} & 0 & g_1^* \\ 1 & \text{---} & \text{---} & \text{---} \\ 0 & \text{---} & \text{---} & \text{---} \\ \vdots & \text{---} & \text{---} & \text{---} \\ 0 & \text{---} & 0 & 1 \\ & & & g_r^* \end{bmatrix}$$

où  $\forall i \in \{1, \dots, r\}, g_i^* = \delta_i + \sum_{k=1}^n \gamma_k f_k^*$

A(.) peut être mise sous forme en flèche et conduit à la matrice F(.) suivante.

$$(I.28) \quad F(.) = \begin{bmatrix} \boxed{F_{11}} & & \boxed{G_{1r}^*} \\ & & \downarrow \\ & \boxed{F_{r-1r-1}} & \boxed{G_{r-1r}^*} \\ \hline 1 \text{---} 1 & \text{---} 1 & \text{---} 1 & \quad \quad G_{rr}^* \end{bmatrix}$$

où  $\forall i \in \{1, \dots, r-1\}$   $F_{ii}$  matrice à éléments constants

$$F_{ii} \in \mathbb{R}^{n_i \times n_i}$$

$G_{ir}^*$  vecteur à éléments non constants décomposable comme suit :

$$G_{ir}^* = \Delta_{ir} + \sum_{k=1}^n f_k^* \Gamma_{ki}$$

avec  $\Delta_i, \Gamma_k$  vecteurs à éléments constants.

$$\Delta_i \in \mathbb{R}^{n_i}, \Delta_i = (d_{1i}, \dots, d_{n_i i})^T$$

$$\forall k \in \{1, \dots, n\}, \Gamma_{ki} \in \mathbb{R}^{n_i}, \Gamma_{ki} = (g_{1k_i}^*, \dots, g_{n_i k_i}^*)$$

Dans ce mémoire, la forme (I.28) sera étudiée dans le cas où les  $F_{ii}$  sont des scalaires. (A l'opposé de (I.26) les termes  $f_i^*$  peuvent être différents.) Une généralisation est cependant facilement envisageable.

### III - METHODES D'ETUDE DE LA STABILITE.

La stabilité est une notion fondamentale dans la synthèse des systèmes automatiques. Elle permet en effet d'évaluer l'influence de certaines perturbations sur la trajectoire nominale d'un processus.

Nous rappellerons ici certaines définitions issues de /HAHN 1963/ concernant la stabilité d'une position d'équilibre du vecteur état  $x(t)$  solution de  $\dot{x} = f(x, t)$  où  $x \in \mathbb{R}^q$  et  $t \in \mathbb{R}$ .

### III.1 - Définitions.

#### Définition 1. Stabilité au sens de Ljapunov (1892).

L'équilibre  $x = x_e$  est stable si :

(I.29)

$$\forall \varepsilon > 0, \forall t_0 \in \mathbb{R}, \exists \delta(\varepsilon, t_0) > 0, \|x_0 - x_e\| < \delta \Rightarrow \|x(t, x_0, t_0) - x_e\| < \varepsilon \quad \forall t > t_0.$$

#### Définition 2. Attractivité.

L'équilibre  $x = x_e$  est attractif si :

(I.30)

$$\exists \rho > 0, \forall \eta > 0, \exists T(\eta, x_0, t_0) > 0, \forall t \geq t_0 + T \quad (\|x_0 - x_e\| \leq \rho \Rightarrow \|x(t, x_0, t_0) - x_e\| \leq \eta)$$

#### Définition 3. Stabilité asymptotique.

Lorsque l'équilibre  $x_0$  est à la fois stable et attractif il est dit asymptotiquement stable.

Des définitions plus complètes, tenant compte de l'influence de l'instant initial  $t_0$  ainsi que des différents domaines de stabilité ont été définies /GRUJIC 1975/. Elles permettent de caractériser l'ensemble des conditions initiales assurant la convergence de l'état vers son équilibre. Cet ensemble constitue le domaine d'attraction. Lorsque le domaine d'attraction est tout l'espace  $\mathbb{R}^q$  la stabilité est asymptotiquement globale. On a alors  $\rho = +\infty$  dans la définition 2.

### III.2 - Rappels de critères usuels d'étude de la stabilité.

Pour étudier les systèmes présentés, nous envisageons l'utilisation de fonctions candidates à Ljapunov /LJAPUNOV 1892/ telles que :

- les fonctions quadratiques /KRASOVSKII 1963/
- les normes vectorielles aboutissant au critère de Borne et Gentina /GENTINA, BORNE 1972/
- les fonctions de type quadratique -plus- intégrale. /LEFSCHETZ 1965/

Dans cette partie, nous rappelons dans un premier temps la méthode directe de Ljapunov.

Nous présentons ensuite les différentes fonctions candidates et leur relation avec l'équation d'état du système ainsi que les différentes définitions qui y sont attachées.

### III.2.1. - Méthode directe de LJAPUNOV.

La mise en oeuvre de la *méthode directe de Ljapunov* (ou seconde méthode) implique le choix d'une fonction particulière  $V(x)$  (ou plus généralement  $V(x,t)$ , /HAHN 1963/) candidate à Ljapunov et qui vérifie les conditions du théorème suivant :

#### Théorème de Ljapunov /1892/

*S'il existe dans un voisinage de l'origine une fonction scalaire  $V(x)$  continue telle que*

$$1^{\circ) \quad V(x) = 0 \Leftrightarrow x = 0$$

$$2^{\circ) \quad \overset{\circ}{V}(0) = 0$$

$$3^{\circ) \quad \exists h \in \mathbb{R}, \forall x \in \mathbb{R}^q - \{0\}$$

$$\|x\| < h \Rightarrow \begin{cases} V(x) > 0 \\ \overset{\circ}{V}(x) \leq 0 \quad (\text{resp. } < 0) \end{cases}$$

*alors l'équilibre  $x=0$  est stable (resp. asymptotiquement stable).*

La stabilité est *asymptotique globale* lorsque le théorème précédent est vérifié et si en plus la fonction  $V(x)$  est telle que :

$$\lim V(x) = +\infty$$

$$\|x\| \rightarrow +\infty$$

Différents types de fonctions  $V(x)$  ont été proposées, en particulier par LUR'E (1951) qui a le premier utilisé des fonctions candidates de Ljapunov de type quadratique plus intégrale. De nombreux auteurs ont utilisé et perfectionné le choix de telles fonctions /YAKUBOVITCH 1962/, /KALMAN 1963/, /LEFSCHETZ 1965/, /POPOV 1973/, /GRUJIC 1978/.

Il a été montré dans /ZAMBETTAKIS 1983/ que ces fonctions s'appliquent très bien à la forme en flèche.

Divers résultats de cette application seront exposés dans le chapitre IV.

### III.2.2 - Fonctions quadratiques

Soit le système décrit par :

$$\dot{x} = A(x,t) x$$

et la fonction quadratique définie positive  $v(x)$  :

$$v(x) = x^T P x$$

où

$$\left\{ \begin{array}{l} P \in \mathbb{R}^{q \times q}, P = P^T \\ P = (p_{ij}) \\ P \text{ vérifie les inégalités de Kotelianski :} \\ p_{11} > 0, \begin{vmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{vmatrix} > 0 \dots, \det P > 0 \end{array} \right.$$

On a alors

$$(I.31) \quad \dot{v}(x) = x^T [A^T(x,t) P + PA(x,t)] x$$

Si la forme quadratique

$$FQ(x,t) = A^T(x,t) P + PA(x,t)$$

est définie négative pour tout  $(x,t)$ , alors l'équilibre est asymptotiquement stable.

Un cas particulier intéressant est celui où la matrice  $P$  est l'identité  $I_q$  et où la matrice  $A(x,t)$  est symétrique

$$A(x,t) = A^T(x,t)$$

La condition est que  $A(x,t)$  soit définie négative.

### III.2.3 - Critère pratique de Borne et Gentina.

La notion de norme vectorielle /GRUJIC, GENTINA, BORNE 1976/ et la définition d'une matrice  $M(A(.))$  pseudo-majorante /GENTINA 1976/, /BORNE 1976/ (matrice obtenue en remplaçant tous ses termes hors diagonaux par leur valeur absolue) ont conduit au critère suivant :

Critère : Lorsque la matrice d'évolution  $A(.)$  du système (I.22) a ses éléments non-constants regroupés dans une seule ligne ou une seule colonne l'équilibre  $x=0$  du système est asymptotiquement stable s'il existe  $\epsilon$  positif tel que

$$(I.32) \quad a_{11} < \epsilon < 0, \quad \begin{vmatrix} a_{11} & |a_{12}| \\ |a_{21}| & a_{22} \end{vmatrix} > -\epsilon > 0, \dots, (-1)^a \begin{vmatrix} a_{11} & |a_{12}| & \dots & |a_{1q}| \\ |a_{21}| & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ |a_{q1}| & \dots & \dots & a_{qq} \end{vmatrix} > -\epsilon > 0$$

Les déterminants intervenants dans cette formulation forment la suite des mineurs principaux successifs de la pseudo-majorante  $M(A(.))$ .

Dans le cas particulier où la matrice  $A(\cdot)$  a ses éléments hors diagonaux tous positifs, elle est égale à sa propre pseudo majorante.

Le critère de Borne et Gentina revient alors à appliquer sur la matrice  $A(\cdot)$  décrivant un système non linéaire les conditions du linéaire obtenues à partir du critère de Koteliansky /GANTMACHER 1966/. La conjecture du linéaire est alors vérifiée /BORNE, GENTINA 1972/, /RICHARD 1984/.

Dans le cas où la matrice  $A(\cdot)$  est mise sous forme en flèche, le critère de Borne et Gentina est équivalent au théorème suivant /BENREJEB 1980/.

Théorème.

-Si chaque bloc  $F_{ii}$  de la diagonale vérifie les conditions de Kotelianski

-Si tous les termes non constants sont dans la même ligne ou la même colonne

alors le système est asymptotiquement stable si la condition suivant est réalisée :

$$(I.33) \quad \exists c > 0 \quad , \quad (-1)^q \det M(A(\cdot)) \geq c > 0$$

Afin d'appliquer ce théorème sur une forme en flèche, il conviendra de choisir les blocs  $F_{ii}$  de l'expression (I.26) de façon à avoir tous leurs termes diagonaux à partie réelle négative.

Ce choix sera détaillé dans le chapitre III.

### III.2.4 - Fonctions de type quadratique plus intégrale.

Ces fonctions sont en général constituées d'une forme quadratique de l'état et des non-linéarités, plus une intégrale des non-linéarités. Par exemple, dans le cas d'un système de type Lur'e Postnikov mLP défini en (I.19) par :

$$\begin{cases} \dot{x} = Ax + bf(\varepsilon) \in \mathbb{R}^q \\ \varepsilon = Cx \in \mathbb{R}^n \end{cases}$$

On posera

$$v(x) = [x^T, f(\varepsilon)] Q \begin{bmatrix} x \\ f(\varepsilon) \end{bmatrix} + \int_0^\varepsilon f^T(u) D du.$$

avec  $Q \in \mathbb{R}^{(q+n) \times (q+n)}$  symétrique, définie positive  
 $D \in \mathbb{R}^{n \times n}$  diagonale  
 $u \in \mathbb{R}^n$  variable muette

Cette fonction  $v(x)$  généralise la fonction de Ljapunov quadratique (I.31)

#### CONCLUSION.

Après avoir défini les notions concernant l'état et la représentation des systèmes, nous avons présenté les différentes formes canoniques utilisées lors de l'étude de systèmes non linéaires et particulièrement la forme en flèche.

Ces représentations ont été appliquées à une classe importante de processus non linéaires du type Lur'e Postnikov monovariabiles étudiés séparément ou regroupés en parallèle.

Enfin, deux classes intéressantes de fonction de Ljapunov ont été rappelées en vue de l'étude de la stabilité.

La forme en flèche s'avère particulièrement adaptée à l'analyse et à la synthèse des systèmes.

C'est sur cette représentation que portera l'exposé dans les chapitres suivants.

CHAPITRE II

UTILISATION D'UN INVARIANT DE REPRESENTATION

## INTRODUCTION.

Dans ce chapitre sont rappelées les différentes définitions liées à la notion de polynôme symbolique /RICHARD 1981/.

Nous montrons comment il est possible d'utiliser cet invariant de représentation dans la modélisation des systèmes décrits dans le chapitre précédent.

Un algorithme de calcul est proposé et permet de calculer le polynôme symbolique d'une représentation matricielle par programme . La méthode de calcul employée est illustrée par un exemple.

### I - NOTION D'INVARIANT DE REPRESENTATION.

Dans cette partie, nous donnons la définition du polynôme symbolique et rappelons les propriétés qui en font un invariant de représentation.

#### I.1 - Définition du polynôme symbolique.

Considérons le système décrit en régime libre par l'équation suivante où  $A(x,t)$  est définie suivant (I.8) ou (I.9)

$$\left\{ \begin{array}{l} \dot{x}(t) = A(x,t) \quad x(t) \\ x(t) \in \mathbb{R}^q, A \in \mathbb{R}^q \times q, \varepsilon \in \mathbb{R}. \end{array} \right.$$

Nous nous proposons de calculer le polynôme caractéristique de la matrice  $A(x,t)$ .

Le calcul du déterminant des matrices à non linéarités de rang 1 dans la partie I.3 du premier chapitre, s'effectue en supposant que leurs coefficients non constants sont écrits sous forme littérale.

Les règles de calcul sont alors les mêmes que sur des matrices constantes. Le calcul du polynôme caractéristique de  $A(x,t)$  est donc réalisé comme s'il s'agissait du calcul d'un déterminant d'une matrice à termes constants.

Par définition, le polynôme symbolique  $P(\lambda, x, t)$  de  $A(\cdot)$  est égal au polynôme caractéristique instantané de la matrice :

$$(II.1) \quad \forall \lambda \in \mathbb{C}, \quad \forall x \in \mathbb{R}^q, \quad \det(\lambda I - A(.)) = P(\lambda, x, t)$$

## I.2 - Invariance du polynôme symbolique.

Si l'on définit une matrice constante régulière  $P$  d'ordre  $q$  caractérisant un changement de base, la matrice  $A_1$  définie en (II.2) est semblable à  $A(.)$

$$(II.2) \quad \begin{cases} P x = P A P^{-1} x \\ A_1 = P A P^{-1} \end{cases}$$

Le changement de base n'affecte pas l'expression du polynôme symbolique et l'on a :

$$(II.3) \quad \det(\lambda I - A_1) = \det(\lambda I - A) = P(\lambda, x, t)$$

Le polynôme symbolique est invariant à travers tout changement de base affectant la représentation d'état. Ainsi, les représentations canoniques présentées dans le chapitre précédent sont semblables si et seulement si :

1°) elles ont même polynôme symbolique

et si elles vérifient les conditions supplémentaires suivantes :

(II.4) 2°) - les matrices sont de même type L(1) ou C(1).

Elles s'écrivent donc sous la forme :

$$A(.) = A + v(.) h^T \quad (I.10)$$

ou

$$A(.) = A + k v^T(.) \quad (I.11)$$

- chaque matrice peut s'écrire sous forme compagnon non constante par changement de base. Il suffit par exemple de vérifier une des deux conditions suivantes :

a) le polynôme symbolique de la matrice n'a pas de zéro indépendant de  $(x, t)$

b) le rang de  $\{h, Ah, A^2h, \dots, A^k h, \dots\}$  est égal à  $q$ .

Le polynôme symbolique constitue donc un outil de calcul permettant de passer d'une forme matricielle à une autre.

Appliquons ces résultats au cas particulier de la représentation des systèmes monovariables du type Lur'e Postnikov.

### I.3 - Cas d'un système monovisible de type Lur'e Postnikov.

#### I.3.1 - Calcul du polynôme symbolique.

Considérons le système décrit (fig. II.1) en supposant vérifiées les hypothèses du paragraphe II.1.2. du chapitre I.

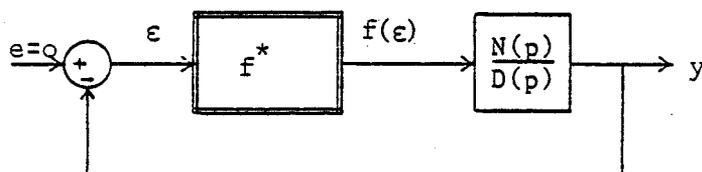


Fig. II.1.

La fonction de transfert du système est supposée non dégénérée,  $N(p)$  et  $D(p)$  sont définis comme suit :

$$(II.5) \quad \left\{ \begin{array}{l} D(p) = p^q + \sum_{i=0}^{q-1} a_i p^i \\ N(p) = \sum_{i=0}^{q-1} b_i p^i \end{array} \right.$$

En appliquant les résultats du premier chapitre, il est possible de mettre la matrice d'évolution  $A(x,t)$  sous la forme  $C(1)$  (ou  $L(1)$ ) décrite en (I.25) puisqu'elles ont même polynôme symbolique égal à

$$(II.6) \quad P(\lambda, \varepsilon) = \lambda^q + \sum_{i=0}^{q-1} (a_i + b_i f^*) \lambda^i$$

Ce polynôme se détermine aisément à partir du schéma bloc et il vient :

$$(II.7) \quad P(\lambda, \varepsilon) = D(\lambda) + f^*(\varepsilon) N(\lambda)$$

Il se calcule donc comme polynôme caractéristique d'un système linéaire bouclé par un gain  $k$  constant en remplaçant  $k$  par le gain équivalent variable  $f^*(\varepsilon)$ . La condition extraite de (II.4) portant sur les zéros constants devient la suivante :  $N(p)$  et  $D(p)$  n'ont pas de zéro commun.

### I.3.2 - Application à la forme en flèche.

A partir de la forme C(1) de (I.25), il est possible d'obtenir une forme en flèche mince par changement de base. La matrice  $P$  de passage est d'après /BENREJEB 1980/ la suivante :

$$(II.8) \quad P = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 \\ \lambda_1 & \lambda_2 & \dots & \lambda_{q-1} & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots \\ \lambda_1^{q-1} & \lambda_2^{q-1} & \dots & \lambda_{q-1}^{q-1} & 1 \end{bmatrix}$$

Les  $\lambda_i$  sont choisis arbitrairement tous différents et non nuls  
 $\forall i \in \{1, \dots, q-1\} \lambda_i \in \mathbb{R}^*$ .

En identifiant les polynômes suivants :

$$(II.9) \quad R(\lambda) = \prod_{i=1}^{q-1} (\lambda - \lambda_i) \text{ et } P(\lambda, \varepsilon) = \lambda^q + \sum_{i=0}^{q-1} (a_i + b_i f^*(\varepsilon)) \lambda^i$$

aux polynômes de la forme en flèche proposée dans /BENREJEB 1980/ il vient l'expression de la forme en flèche suivante :

$$\begin{aligned}
 F(x,t) = P^{-1} A(.) P = & \left[ \begin{array}{ccc}
 \lambda_1 & & \left\{ \frac{-P(\lambda, \epsilon)}{R(\lambda)} (\lambda - \lambda_1) \right\}_{\lambda = \lambda_1} \\
 & \ddots & \\
 & & \lambda_{q-1} \left\{ \frac{-P(\lambda, \epsilon)}{R(\lambda)} (\lambda - \lambda_{q-1}) \right\}_{\lambda = \lambda_{q-1}} \\
 1 & \text{-----} & 1 \quad -a_{q-1} - b_{q-1} f^* - \sum_{i=1}^{q-1} \lambda_i
 \end{array} \right]
 \end{aligned}
 \tag{II.10}$$

Le calcul du polynôme caractéristique de la matrice (II.10) conduit à l'expression du polynôme symbolique du système (II.7). Les matrices  $A(x,t)$  et  $F(x,t)$  sont donc semblables.

A partir de l'expression (II.7) du polynôme symbolique, il est facile de construire une forme en flèche semblable à toute autre représentation du système. Le calcul se fait uniquement sur des polynômes et ne fait intervenir aucun calcul de changement de base.

Par conséquent, pour obtenir l'expression des termes non constants de la forme en flèche mince, il suffit de calculer la valeur du dénominateur et du numérateur de la fonction de transfert en une valeur  $\lambda_i$  de  $\lambda$  puis la valeur de  $R(\lambda)$ . On place enfin les  $\lambda_i$  sur la diagonale et des 1 dans la dernière ligne.

A toutes valeurs de  $\lambda_i$  correspond une forme en flèche. Le choix des paramètres  $\lambda_i$  est fondamental dans l'expression de la matrice. Les  $\lambda_i$  peuvent être dans le cas des systèmes mLP les pôles ou les zéros de la fonction de transfert.  $P(\lambda, \epsilon)$  est alors réduit à  $f^*(.) N(\lambda)$  ou  $D(\lambda)$ .

Nous détaillerons dans le chapitre III les différentes façons de choisir les paramètres  $\lambda_i$ .

Auparavant, nous expliquons comment calculer par programme le polynôme symbolique.

## II - CALCUL INFORMATIQUE DU POLYNOME SYMBOLIQUE.

### II.1 - Introduction.

Dans la première partie de ce chapitre, nous avons montré l'importance du polynôme symbolique dans la définition du système.

A partir de cet invariant de représentation est établie une modélisation du système.

Deux cas sont alors envisageables :

1°) Le système est défini par son schéma bloc. L'expression du polynôme symbolique est alors immédiate d'après (II.7).

2°) Le système est défini par une matrice d'évolution. On calcule son polynôme caractéristique (II.1) qui conduit alors au schéma bloc.

Il existe différents algorithmes qui réalisent le calcul du polynôme caractéristique de matrices constantes.

Les matrices décrivant les systèmes étudiés sont toutes non constantes et nous ne pouvons utiliser ces algorithmes existants.

La plus grande difficulté du calcul est donc due à la présence de ces termes non constants.

Nous envisageons ici le traitement de matrices à coefficients non constants tous situés dans la dernière colonne et dont l'expression est tout à fait quelconque. Les termes non linéaires sont bien souvent des combinaisons linéaires de  $f_i$  comme dans le cas des matrices présentées en (I.27) et (I.28).

Pour effectuer le calcul du polynôme symbolique nous décomposons la matrice non constante en plusieurs matrices et appliquons les méthodes connues.

## II.2 - Algorithme de calcul.

### II.2.1 - Décomposition de la matrice.

La matrice  $A(x,t)$  a ses termes non constants dans la dernière colonne. Elle se présente donc comme suit :

$$(II.11) \quad A(x,t) = \left[ \begin{array}{c|c} \begin{array}{cc} \alpha_{11} & \alpha_{1q-1} \\ \vdots & \vdots \\ \alpha_{q1} & \alpha_{qq-1} \end{array} & \begin{array}{c} \alpha'_{1q} \\ \vdots \\ \alpha'_{qq} \end{array} \end{array} \right]$$

Le premier bloc formé des  $\alpha_{ij}$  avec  $i = 1, \dots, q$  et  $j = 1, \dots, q-1$ , est un bloc à termes constants. La dernière colonne formée des  $\alpha'_{iq}$  comporte les termes non constants, chaque  $\alpha'_{iq}$  étant combinaison linéaire des  $f_k^*$ . Le nombre de non-linéarités  $k$  varie de 1 à  $n$ .

$$\forall i \in \{1, \dots, q\}, \alpha'_{iq} = \alpha_{iq} + \sum_{k=1}^n a_{ik} \cdot f_k^*$$

Nous décomposons la matrice  $A(\cdot)$  comme suit :

$$(II.12) \quad A(\cdot) = C + (0, \dots, 0, \text{CNL} \cdot F)$$

$$\text{où } C = (\alpha_{ij}) \begin{cases} i = 1, \dots, q \\ j = 1, \dots, q \end{cases} \quad C \in \mathbb{R}^{q \times q}$$

$C$  : matrice des coefficients constants

$$\text{CNL} = (a_{ij}) \begin{cases} i = 1, \dots, q \\ j = 1, \dots, n \end{cases} \quad \text{CNL} \in \mathbb{R}^{q \times n}$$

$\text{CNL}$  : matrice des coefficients des non-linéarités

$$F^T(\cdot) = (f_1^*(\cdot), \dots, f_n^*(\cdot)) \quad F \in \mathbb{R}^n$$

$F$  : vecteur des non linéarités

II.2.2 - Calcul de polynôme symbolique.

Calculons le polynôme caractéristique de la matrice définie en (II.12)

(II.13)

$$\det(\lambda I - A) = \det \left[ \begin{array}{ccc|c} \lambda - \alpha_{11} & -\alpha_{12} & \dots & -\alpha_{1q-1} \\ -\alpha_{21} & \lambda - \alpha_{22} & & \\ \vdots & \vdots & \ddots & \vdots \\ -\alpha_{q1} & -\alpha_{q2} & \dots & -\alpha_{qq-1} \end{array} \right] - \left( \alpha_{1q} + \sum_{k=1}^n a_{1k} f_k^* \right) \left[ \begin{array}{c} \vdots \\ \vdots \\ \lambda - (\alpha_{qq} + \sum_{k=1}^n a_{qk} f_k^*) \end{array} \right]$$

ou encore

(II.14)

$$\det(\lambda I - A) = \det \left[ \begin{array}{ccc|c} \lambda - \alpha_{11} & -\alpha_{12} & \dots & -\alpha_{q1} \\ -\alpha_{21} & \lambda - \alpha_{22} & & \\ \vdots & \vdots & \ddots & \vdots \\ -\alpha_{q1} & -\alpha_{q2} & \dots & \lambda - \alpha_{qq} \end{array} \right] - \left[ \begin{array}{c} \sum_{k=1}^n a_{1k} \cdot f_k^* \\ \vdots \\ \sum_{k=1}^n a_{qk} \cdot f_k^* \end{array} \right]$$

En utilisant les propriétés des déterminants, on obtient :

(II.15)

$$\det(\lambda I - A) = \det \left( \begin{array}{c|c} \square & \begin{array}{c} -\alpha_{q1} \\ \vdots \\ \lambda - \alpha_{qq} \end{array} \end{array} \right) + \det \left( \begin{array}{c|c} \square & \begin{array}{c} -\sum_{k=1}^n a_{1k} \cdot f_k^* \\ \vdots \\ -\sum_{k=1}^n a_{qk} \cdot f_k^* \end{array} \end{array} \right)$$

Le premier terme de la somme est égal au polynôme caractéristique de la matrice C (II.12) que l'on calcule aisément. Décomposons le deuxième terme comme suit : on ajoute  $\lambda$  au dernier terme de la dernière colonne puis on retranche  $\lambda$ .

Ce deuxième terme peut alors s'écrire sous la forme d'une somme de polynômes caractéristiques :

(II.16)

$$\sum_{k=1}^n f_k \left\{ \det \begin{bmatrix} \square & -a_{1k} \\ \vdots & \vdots \\ \square & \lambda - a_{qk} \end{bmatrix} + \det \begin{bmatrix} \square & 0 \\ \vdots & \vdots \\ \square & -\lambda \end{bmatrix} \right\}$$

Le premier calcul de déterminant correspond au calcul du polynôme caractéristique  $P_{C_k}$  de la matrice  $C_k$  définie comme suit :

$C_k$  est obtenue en considérant la matrice  $C$  définie en (II.12) et en remplaçant la dernière colonne de  $C$  pour la  $k^{\text{ième}}$  colonne de CNL définie en (II.12).

Le deuxième déterminant apparaît comme étant le produit du polynôme caractéristique  $P_{CR}$  de la matrice  $CR$  par  $(-\lambda)$ .

La matrice  $CR$  est égale à la matrice  $C$  (II.12) en lui enlevant la dernière ligne et la dernière colonne. Elle est donc telle que :

$$(II.17) \quad CR = (\alpha_{ij}) \quad \begin{cases} i = 1, \dots, q-1 \\ j = 1, \dots, q-1 \end{cases}$$



Nous obtenons finalement en reprenant les expressions (II.15) et (II.16)

$$(II.18) \quad \det(\lambda I - A) = P_C(\lambda) + \sum_{k=1}^n f_k P_{C_k}(\lambda) - \lambda P_{CR}(\lambda) \left( \sum_{k=1}^n f_k \right)$$

par analogie avec l'expression de  $P(\lambda, \dots)$  (II.6) dans le cas d'un système de type Lur'e Postnikov à une seule non linéarité, nous posons :

$$(II.19) \quad \det(\lambda I - A) = D(\lambda) + \sum_{k=1}^n f_k N_k(\lambda).$$

en identifiant à (II.18), il vient :

$$(II.20) \quad \begin{cases} D(\lambda) = P_C(\lambda) \\ N(\lambda) = P_{C_K}(\lambda) = \lambda P_{CR}(\lambda) \end{cases}$$

### II.2.3 - Exemple.

Dans le cas particulier où la matrice d'évolution ne comporte qu'une seule non linéarité, la matrice A se met sous la forme suivante :

$$(II.21) \quad A = \begin{bmatrix} \alpha_{11} & \text{---} & \alpha_{1q-1} & \alpha_{1q} + a_{11} f^* \\ \vdots & & \vdots & \vdots \\ \alpha_{q1} & \text{---} & \alpha_{qq-1} & \alpha_{qq} + a_{q1} f^* \end{bmatrix}$$

Le calcul de son polynôme caractéristique se fait suivant l'algorithme décrit dans le paragraphe (II.1.3) et il vient :

$$(II.22) \quad \det(\lambda I - A) = P_C(\lambda) + f^* P_{C_1}(\lambda) - \lambda P_{CR}(\lambda) \cdot f^*$$

avec

$$(II.23) \quad \left\{ \begin{array}{l} C = (\alpha_{ij}) \text{ pour } \begin{cases} i = 1, \dots, q \\ j = 1, \dots, q \end{cases} \\ \\ CR = (\alpha_{ij}) \text{ pour } \begin{cases} i = 1, \dots, q-1 \\ j = 1, \dots, q-1 \end{cases} \\ \\ C_1 = (\alpha_{ij}) \quad \forall j < q \\ \\ C_1 = (a_{ij}) \text{ pour } j = q. \end{array} \right.$$

Ce cas particulier est souvent rencontré. La matrice d'évolution correspond en effet au cas le plus simple de système du type Lur'e Postnikov.

### II.3 - Développements informatiques.

Le calcul du polynôme symbolique d'une matrice à coefficients non constants nécessite comme le montre l'expression (II.18) le calcul de différents polynômes caractéristiques.

Il existe de nombreuses méthodes de calcul de valeurs propres et vecteurs propres qui déterminent dans un premier temps l'expression du polynôme caractéristique. Citons parmi ces méthodes celles de Krylov, d'Hessenberg, de Samuelson et deux méthodes plus efficaces: la méthode de Danilevskii et celle de Leverrier /FADEEV, FADEEVA 1963/.

L'efficacité d'une méthode peut être établie à partir de la notion de stabilité numérique ou suivant le nombre d'opérations qu'elle nécessite.

Précisons en première partie les notions de stabilité numérique et de conditionnement d'un problème.

#### II.3.1 - Conditionnement et stabilité.

On dit qu'un problème est bien (mal) conditionné si une petite variation des données entraîne une petite (grande) variation sur les résultats /BREZINSKI/.

La notion de conditionnement est liée au problème mathématique et est indépendante de la méthode utilisée pour le résoudre.

Une autre notion importante est celle de la stabilité numérique. Un problème peut être bien conditionné et la méthode utilisée pour le résoudre peut entraîner une propagation importante des erreurs.

Une méthode est dite stable numériquement si elle donne de bons résultats quelles que soient les données.

Si un problème est mal conditionné, quelle que soit la méthode utilisée, les résultats sont erronés. Il faut donc essayer de donner une formulation mathématique du problème différente.

Dans le cas de problèmes matriciels, un mauvais conditionnement de la matrice peut être résolu par un rééquilibrage de la matrice.

Ces différentes notions nous ont permis de choisir une méthode parmi celles citées plus haut.

### II.3.2 - Méthode utilisée.

Deux méthodes sont particulièrement intéressantes : la méthode de Danilevskii et celle de Leverrier

- pour la première le nombre d'opérations à effectuer est nettement inférieur à celui demandé par les autres méthodes.

Il est en effet égal à  $n^3 - n^2$  contre  $n^4 + 4n^3 + 2n^2 - n - 3$  dans le cas de la méthode de Krylov.

- pour la méthode de Leverrier, son avantage est qu'elle reste insensible aux particularités de la matrice et à la disproportion qu'elles peuvent entraîner dans les valeurs des déterminants intermédiaires.

Le nombre d'opérations demandé est toutefois important :

$$\frac{1}{2} (n-1) (2n^3 - 2n^2 + n + 2)$$

Plus la matrice est grande et plus la méthode est lente.

C'est cette dernière que nous utilisons, préférant une méthode stable numériquement à une méthode économique.

Dans la partie suivante, nous détaillons l'algorithme de Leverrier sur un exemple.

### II.3.3 - Exemple.

Soit la matrice d'évolution  $A(.)$  comportant deux non linéarités.

$$A(.) = \begin{pmatrix} 1 & 3 & 1 + f_1^* + f_2^* \\ 0 & 1 & 2 + f_1^* - f_2^* \\ 1 & 1 & 3 - f_1^* + 2f_2^* \end{pmatrix}$$

On entre en données les matrices C, CNL, F (cf. paragraphe II.2)

$$C = \begin{pmatrix} 1 & 3 & 1 \\ 0 & 1 & 2 \\ 1 & 1 & 3 \end{pmatrix} \quad \text{CNL} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 2 \end{pmatrix} \quad F^T = (f_1^*, f_2^*)$$

Le calcul du polynôme symbolique se fait conformément à l'algorithme présenté dans le paragraphe II.3

$$\det(\lambda I - A) = \det \begin{pmatrix} \lambda - 1 & -3 & -1 \\ 0 & \lambda - 1 & -2 \\ -1 & -1 & \lambda - 3 \end{pmatrix} + \det \begin{pmatrix} \square & -f_1^* & -f_2^* \\ \square & -f_1^* + f_2^* & \square \\ \square & +f_1^* & -2f_2^* \end{pmatrix}$$

= polynôme caractéristique de la matrice C. + déterminant D



Le second déterminant se calcule comme suit

$$D = f_1 \left\{ \det \begin{pmatrix} \square & -1 \\ \square & -1 \\ \square & \lambda + 1 \end{pmatrix} + \det \begin{pmatrix} \square & 0 \\ \square & 0 \\ \square & -\lambda \end{pmatrix} \right\}$$

$$+ f_2^* \left\{ \det \begin{pmatrix} \square & -1 \\ \square & +1 \\ \square & \lambda - 2 \end{pmatrix} + \det \begin{pmatrix} \square & 0 \\ \square & 0 \\ \square & -\lambda \end{pmatrix} \right\}$$

d'où finalement

$$\det(\lambda I - A) = P_C(\lambda) + f_1^* P_{C_1}(\lambda) + f_2^* P_{C_2}(\lambda) - (f_1^* + f_2^*) P_{CR}(\lambda)$$

avec

$$C_1 = \begin{pmatrix} 1 & 3 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \quad C_2 = \begin{pmatrix} 1 & 3 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 2 \end{pmatrix} \quad CR = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}$$

Chacun des polynômes caractéristiques  $P_C, P_{C_1}, P_{C_2}, P_{CR}$  est calculé suivant l'algorithme de Leverrier que nous détaillons et illustrons sur un exemple.

Soit une matrice  $A$  d'ordre  $n$ , dont le polynôme caractéristique est donné par :

$$\det(\lambda I - A) = \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_0.$$

Le calcul des coefficients  $a_i$  s'effectue comme suit :

On définit les matrices  $B_i$  suivantes :

$$\begin{aligned} B_{n-1} &= I_n \\ B_{n-2} &= a_{n-1} I_n + AB_{n-1} \\ B_{n-3} &= a_{n-2} I_n + AB_{n-2} \\ &\vdots \\ B_1 &= a_2 I_n + AB_2 \\ B_0 &= a_1 I_n + AB_1 \end{aligned}$$

Les coefficients  $a_i$  sont donnés par :

$$\begin{aligned} a_{n-1} &= -\operatorname{tr}(A \cdot B_{n-1}) \\ a_{n-2} &= -\frac{1}{2} \operatorname{tr}(A \cdot B_{n-2}) \\ a_{n-3} &= -\frac{1}{3} \operatorname{tr}(A \cdot B_{n-3}) \\ &\vdots \\ a_1 &= -\frac{1}{n-1} \operatorname{tr}(A \cdot B_1) \\ a_0 &= -\frac{1}{n} \operatorname{tr}(A \cdot B_0) \end{aligned}$$

La démonstration de ces formules est donnée dans /DEIF 1982/.

Calculons suivant cet algorithme le polynôme caractéristique  $P_C(\lambda)$  de  $C$

$$C = \begin{pmatrix} 1 & 3 & 1 \\ 0 & 1 & 2 \\ 1 & 1 & 3 \end{pmatrix}$$

$$P_C(\lambda) = \lambda^3 + a_2 \lambda^2 + a_1 \lambda + a_0$$

avec  $a_2 = -\text{tr } C_1 = -5$

$$B_1 = -5I + C_1$$

$$B_1 = \begin{pmatrix} -4 & 3 & 1 \\ 0 & -4 & 2 \\ 1 & 1 & -2 \end{pmatrix} \quad \text{et} \quad C_1 B_1 = \begin{pmatrix} -3 & -8 & 5 \\ 2 & -2 & -2 \\ -1 & 2 & -3 \end{pmatrix}$$

$$a_1 = -\frac{1}{2} \text{tr}(C_1 B_1) = 4$$

$$B_0 = 4I + C_1 B_1$$

$$B_0 = \begin{pmatrix} 1 & -8 & 5 \\ 2 & 2 & -2 \\ -1 & 2 & 1 \end{pmatrix} \quad \text{et} \quad C_1 B_0 = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{pmatrix}$$

$$a_0 = -\frac{1}{3} \text{tr}(C_1 B_0) = -6$$

d'où

$$P_C(\lambda) = \lambda^3 - 5\lambda^2 + 4\lambda - 6$$

De la même façon, il vient :

$$P_{C_1}(\lambda) = \lambda^3 - \lambda^2 - 3\lambda$$

$$P_{C_2}(\lambda) = \lambda^3 - 4\lambda^2 + 5\lambda + 1$$

$$P_{CR}(\lambda) = \lambda^2 - 2\lambda + 1$$



d'où l'expression globale de  $P_A(\lambda)$

$$P_A(\lambda) = (\lambda^3 - 5\lambda^2 + 4\lambda - 6) + f_1(\lambda^2 - 4\lambda) + f_2(-2\lambda^2 + 4\lambda + 1)$$

### CONCLUSION.

Dans ce chapitre, nous avons rappelé la définition de l'invariant de représentation qu'est le polynôme symbolique : deux matrices sont semblables ssi elles ont même polynôme symbolique.

Un algorithme permettant de le calculer par programme a été donné. Une fois le calcul effectué, le polynôme symbolique permet de déterminer une représentation en flèche mince du système étudié.

Ainsi à partir d'une matrice d'évolution ou d'un schéma bloc on peut établir une nouvelle représentation matricielle du système. Cette représentation peut être en flèche et dépend alors de paramètres  $\lambda_i$  choisis arbitrairement par l'utilisateur.

Dans le chapitre suivant, nous expliquerons comment il est possible de choisir ces paramètres  $\lambda_i$ .

CHAPITRE III  
MODELISATION DES SYSTEMES  
A L'AIDE DE LA FORME EN FLECHE.

## INTRODUCTION.

Le chapitre précédent a permis de définir et de calculer le polynôme symbolique d'une matrice à non linéarités de rang 1.

Dans ce chapitre nous montrons comment, à partir de la notion d'invariant de représentation, il est possible d'obtenir une représentation matricielle spécifique : la forme en flèche, qui s'avère particulièrement adaptée à l'étude de la stabilité des systèmes /BENREJEB 1980/, /ROTELLA 1983/.

A partir de la forme en flèche la plus générale, un nouveau polynôme est défini, appelé polynôme représentatif et noté  $R(\lambda, .)$ .

Par opposition avec le polynôme symbolique qui détermine la structure du système de façon invariante quelle que soit la représentation, le polynôme représentatif définit la représentation adoptée.

A toute forme en flèche correspond un polynôme représentatif. Le choix de ce polynôme est équivalent au choix des paramètres  $\lambda_i$  diagonaux intervenant dans la modélisation, chaque  $\lambda_i$  étant un zéro de  $R(\lambda, .)$ .

Différentes méthodes de choix des  $\lambda_i$  sont rappelées et programmées. Ainsi il est possible de trouver des paramètres répondant du mieux possible aux exigences que nous fixons pour étudier le système.

## I - MODELISATION DES SYSTEMES MONOVARIABLES DE TYPE LUR'E POSTNIKOV.

Dans cette partie sont exposées différentes formes de matrices en flèche susceptibles de représenter les systèmes décrits dans le chapitre I.

Nous rappelons tout d'abord la forme générale de la matrice en flèche et définissons son *polynôme représentatif* /DIMSTER, ROTELLA, RICHARD 1984/.

### I.1 - Définition du polynôme représentatif.

Soit la matrice en flèche déjà définie dans le premier chapitre en (I.16) :

$$(III.1) \quad F(.) = \begin{bmatrix} \boxed{A_{11}(.)} & & \boxed{A_{1r}(.)} \\ & 0 & \downarrow \\ \boxed{A_{r1}(.)} & \xrightarrow{\quad} & \boxed{A_{rr}(.)} \end{bmatrix}$$

où  $A_{ii} \in C^{n_i \times n_i}$ .

Dans /BENREJEB 1980/ a été proposé le théorème suivant :

Théorème.

Lorsque les matrices  $A_{ii}(.)$ ,  $\forall i \in \{1, \dots, r-1\}$ , sont régulières, le déterminant de  $F(.)$  définie en (III.1) est égal à :

$$(III.2) \quad \det(F(.)) = \det\left(A_{rr} - \sum_{i=1}^{r-1} A_{ri} A_{ii}^{-1} A_{ir}\right) \prod_{i=1}^{r-1} \det(A_{ii})$$

La matrice  $(\lambda I_q - F(.))$  est également une forme en flèche il vient donc :

$$(III.3) \quad P(\lambda, .) = \det\left((\lambda I_{n_r} - A_{rr}) - \sum_{i=1}^{r-1} A_{ri} (\lambda I_{n_i} - A_{ii})^{-1} A_{ir}\right) \prod_{i=1}^{r-1} \det(\lambda I_{n_i} - A_{ii})$$

où  $I_{n_i}$  désigne la matrice identité d'ordre  $n_i$ .

Définition.

Le polynôme à coefficients réels défini par :

$$(III.4) \quad R(., .) = \prod_{i=1}^{r-1} \det(\lambda I_{n_i} - A_{ii})$$

est appelé polynôme représentatif de la forme en flèche.

De précédents travaux /DIMSTER, ROTELLA, RICHARD 1984/ font intervenir des formes en flèche dont le polynôme représentatif présente au plus un zéro non constant.

Ce cas particulier, bien qu'étant également programmable n'est pas envisagé dans notre réalisation informatique. Celle-ci ne concerne donc que le cas où  $R(\lambda, .)$  admet des zéros constants,  $R(\lambda, .)$  est alors noté  $R(\lambda)$ .

Considérant une matrice  $F(.)$  (III.1) à non-linéarités de rang 1, les coefficients des  $A_{ij}$  sont déterminés à partir de deux types de données :

- la connaissance du polynôme symbolique
- les coefficients des matrices  $A_{ii}$  de la diagonale. Le choix de ces coefficients est équivalent au choix des paramètres  $\lambda_i$ .

En effet chacun des coefficients est une racine de  $R(\lambda)$ . C'est pourquoi  $R(\lambda)$  est appelé polynôme représentatif par opposition au polynôme symbolique qui ne dépend pas de  $\lambda$ . Ce dernier détermine la structure du système tandis que le polynôme représentatif caractérise la modélisation.

Ainsi suivant la nature de ses racines, qui peuvent être réelles, simples multiples ou complexes, les blocs diagonaux ( $A_{ii}$ ) de  $F(.)$  sont alors conventionnellement choisis sous formes canoniques diagonales de Jordan ou de rotation c'est à dire :

- si  $R(\lambda)$  admet des racines réelles simples  $\lambda_i$ .  
 $A_{ii}$  est réduite à un scalaire  $\lambda_i$ .
- Si  $R(\lambda)$  admet des racines multiples  $\lambda_i$   
 $A_{ii}$  est une forme de Jordan

$$A_{ii} = M_i = \begin{pmatrix} \lambda_i & & 0 \\ & 1 & \\ & & 1 \\ 0 & & \lambda_i \end{pmatrix}$$

l'ordre de la matrice est égal à la multiplicité de la racine.

- Si  $R(\lambda)$  admet des racines complexes  $\lambda_j = \mu_j + i\nu_j$

$$A_{ii} = C_i = \begin{pmatrix} \mu_j & \nu_j \\ -\nu_j & \mu_j \end{pmatrix}$$

## I.2 - Détermination des formes en flèche.

La détermination des formes en flèche est réalisée à partir des polynômes symbolique et représentatif.

Différentes représentations sont envisageables selon que  $R(\lambda)$  admet des racines réelles, complexes, distinctes ou multiples. Ce sont les formulations présentées dans /ROTELLA 1983/ que nous allons rappeler.

### I.2.1. - $R(\lambda)$ admet des racines réelles distinctes.

Nous supposons que  $R(\lambda)$  admet  $(q-1)$  racines réelles distinctes  
 $R(\lambda) = \prod_{i=1}^{q-1} (\lambda - \lambda_i)$ ,  $q$  étant l'ordre du système.

La forme de la matrice est alors celle présentée dans le paragraphe (I.3.2) du chapitre II.

#### I.2.1.1 - Forme matricielle.

Dans le cas de systèmes du type Lur'e Postnikov, les termes de la dernière colonne se mettent sous la forme  $\delta_i + \gamma_i \cdot f^*$  avec  $\delta_i$  et  $\gamma_i$  réels.

Le polynôme symbolique s'écrit  $P(\lambda, \cdot) = D(\lambda) + f^* \cdot N(\lambda)$ . La représentation de  $F(\cdot)$  devient avec cette notation :

$$(III.5) \quad F(\cdot) = \begin{bmatrix} \lambda_1 & & & \delta_1 + \gamma_1 \cdot f^* \\ & \lambda_{q-1} & & \delta_{q-1} + \gamma_{q-1} \cdot f^* \\ & & 1 & \\ & & & -a_{q-1} - b_{q-1} f^* - \sum_{i=1}^{q-1} \lambda_i \end{bmatrix}$$

Par identification avec la forme présentée en (II.10) les coefficients des termes non constants sont définis par :

$$(III.6) \quad \delta_i + \gamma_i \cdot f^* = \left[ \frac{-P(\lambda, \cdot)}{R(\lambda)} \right]_{\lambda = \lambda_i}$$

ou encore en posant :

$$R_i(\lambda) = \left( \frac{R(\lambda)}{(\lambda - \lambda_i)} \right)_{\lambda = \lambda_i}$$

$$\delta_i = - \frac{D(\lambda_i)}{R_i(\lambda_i)}$$

$$\gamma_i = - \frac{N(\lambda_i)}{R_i(\lambda_i)}$$

I.2.1.2 - Cas plus général : le système comporte n non linéarités.

Un cas plus général est illustré par un ensemble de systèmes de type Lur'e Postnikov mis en parallèle, représenté (fig. I.5) au chapitre I.

Si on choisit  $R(\lambda)$  de façon à ce que ses  $q-1$  racines soient réelles simples, la forme en flèche mince correspondante est définie par :

$$(III.7) \quad F(\cdot) = \begin{bmatrix} \lambda_1 & & \delta_1 + \sum_{k=1}^n \gamma_{1k} \cdot f_k^* & & \\ & \ddots & & & \\ & & \lambda_{q-1} & & \delta_{q-1} + \sum_{k=1}^n \gamma_{q-1k} \cdot f_k^* \\ & & & & \\ 1 & \text{---} & 1 & & \gamma(\cdot) \end{bmatrix}$$

où

$$\delta_i + \sum_{k=1}^n \gamma_{ik} \cdot f_k^* = \frac{-P(\lambda_i, \cdot)}{R_i(\lambda_i)}$$

et

$$(III.8) \quad \left\{ \begin{array}{l} P(\lambda_i, \cdot) = D(\lambda_i) + \sum_{k=1}^n f_k^* \cdot N_k(\lambda_i) \\ R_i(\lambda_i) = \left( \frac{R(\lambda)}{\lambda - \lambda_i} \right)_{\lambda=\lambda_i} \end{array} \right.$$

d'où plus précisément :

$$\left\{ \begin{array}{l} \delta_i = \frac{-D(\lambda_i)}{R_i(\lambda_i)} \quad \forall i \in \{1, \dots, q-1\} \\ \gamma_{ik} = \frac{-N_k(\lambda_i)}{R_i(\lambda_i)} \quad \begin{array}{l} \forall k \in \{1, \dots, n\} \\ \forall i \in \{1, \dots, q-1\} \end{array} \end{array} \right.$$

et

$$\gamma(\cdot) = -a_{q-1} - \sum_{k=1}^n b_{kq-1} f_k^* - \sum_{i=1}^{q-1} \lambda_i$$

avec  $b_{kq-1}$  coefficients d'ordre  $q-1$  des polynômes  $N_k(\lambda)$ .

I.2.2 -  $R(\lambda)$  admet des racines multiples et des racines complexes.

I.2.2.1 - Cas général.

I.2.2.1.1 - Forme matricielle.

Si  $R(\lambda)$  a pour expression :

$$(III.9) \quad R(\lambda) = \prod_{i=1}^s (\lambda - \lambda_i)^{n_i} \prod_{j=1}^t ((\lambda - \mu_j)^2 + \nu_j^2)^{m_j}$$

La matrice d'évolution se présente sous forme de matrice en flèche épaisse comme suit :

$$(III.10) \quad F(\cdot) = \left[ \begin{array}{cccc|c} M_1 & & & & \alpha_1(\cdot) \\ & \searrow & & & | \\ & & M_s & & \alpha_s(\cdot) \\ & & & & | \\ & & & c_1 & \beta_1(\cdot) \\ & & 0 & \searrow & | \\ & & & & \beta_t(\cdot) \\ & & & & | \\ 1 & \text{---} & 1 & & 1 & \text{---} & 1 & \gamma(\cdot) \end{array} \right]$$

où

$$- \forall (i,k) \in \{1, \dots, s\}^2$$

- $n_i$  : multiplicité de la racine  $\lambda_i$   
 $n_i \in \mathbb{N}, \lambda_i \in \mathbb{R}$

- $M_i \in \mathbb{R}^{n_i \times n_i}$ , matrice sous forme de Jordan

$$(III.11) \quad M_i = \left[ \begin{array}{cccc|c} \lambda_i & 1 & 0 & \text{---} & 0 \\ & \searrow & & & | \\ & & & & 0 \\ & & & & | \\ & & & & 1 \\ & & 0 & \searrow & \\ & & & & \lambda_i \end{array} \right]$$

- $\alpha_i(\cdot) \in \mathbb{R}^{n_i}, \alpha_i^T(\cdot) = (\alpha_i^{n_i}(\cdot), \dots, \alpha_i^1(\cdot))$

$$- \forall (j,k) \in \{1, \dots, t\}^2$$

•  $m_j$  : multiplicité de la racine complexe

$\mu_j$  : partie réelle de la racine

$\nu_j$  : partie complexe de la racine

$$(\mu_j, \nu_j) \in \mathbb{R}^2$$

•  $C_j \in \mathbb{R}^{2m_j \times 2m_j}$

•  $j \neq k \Leftrightarrow \mu_j \neq \mu_k$  ou  $|\nu_j| \neq |\nu_k|$

(III.12)

$$C_j = \begin{bmatrix} \begin{array}{cc|cc|c} \mu_j & \nu_j & 1 & 0 & 0 \\ -\nu_j & \mu_j & 0 & 1 & 0 \\ \hline 0 & & \mu_j & \nu_j & 0 \\ & & -\nu_j & \mu_j & 1 \\ \hline 0 & & & & \mu_j & \nu_j \\ & & & & -\nu_j & \mu_j \end{array} \end{bmatrix}$$

•  $\beta_j(\cdot) \in \mathbb{R}^{2m_j}$

$$\beta_j^T(\cdot) = (\beta_{j2}^{m_j}(\cdot), \beta_{j1}^{m_j}(\cdot), \dots, \beta_{j2}^1(\cdot), \beta_{j1}^1(\cdot))$$

$$- \sum_{i=1}^s n_i + 2 \sum_{j=1}^t m_j = q-1.$$

#### I.2.2.1.2 - Calculs des termes non constants.

Les termes non constants contenus dans la dernière colonne de la matrice  $F(\cdot)$  s'écrivent comme suit :

$$\alpha_i(\cdot) = \delta_i + f^* \cdot \gamma_i \quad \left\{ \begin{array}{l} \gamma_i \in \mathbb{R}^{n_i} \\ \delta_i \in \mathbb{R}^{n_i} \end{array} \right.$$

$$\bullet \beta_j(.) = \delta_j' + f^* \cdot \gamma_j' \quad \left\{ \begin{array}{l} \gamma_j' \in \mathbb{R}^{2m_j} \\ \delta_j' \in \mathbb{R}^{2m_j} \end{array} \right.$$

$$\bullet \gamma = -a_{q-1} - b_{q-1} \cdot f^* - \sum_{i=1}^s n_i \lambda_i - 2 \sum_{j=1}^t \mu_j m_j.$$

Le calcul des termes  $\gamma_i, \delta_i$  et  $\gamma_j, \delta_j$  est effectué dans /ROTELLA 1983/. Nous rappelons ici les résultats concernant le cas où les blocs  $C_j$  (III.12) de  $F(.)$  sont tous d'ordre 2 c'est à dire  $m_j = 1$ .

### Notations.

De façon à exprimer plus simplement les termes  $\alpha_i$  et  $\beta_i$ , nous noterons :

$$R_i(\lambda) = \prod_{j=1}^t \{(\lambda - \mu_j)^2 + v_j^2\} \prod_{\substack{\ell=1 \\ \ell \neq i}}^s (\lambda - \lambda_\ell)^{n_\ell}$$

$$R_j(\lambda) = \prod_{\substack{\ell=1 \\ \ell \neq j}}^t \{(\lambda - \mu_\ell)^2 + v_\ell^2\} \prod_{i=1}^s (\lambda - \lambda_i)^{n_i}$$

(III.13)

$$(A)^* = \left( \frac{P(\lambda, .)}{R_j(.)} \right)_{\lambda = \mu_j + i v_j}$$

(III.14)

$$(B)^* = P^{j-1}(\lambda, .) + \sum_{p=0}^{j-2} C_{j-1}^p R_i^{(j-p-1)}(\lambda) p! \sum_{\ell=1}^{p+1} \alpha_i^\ell(.)$$

Dans ces relations,

$C_{j-1}^p$  représente les coefficients du binôme.

$P(\lambda, .)$  représente le polynôme symbolique défini dans le chapitre II et  $P^{(i)}(\lambda, .)$  la dérivation  $i^{\text{ème}}$  de  $P(\lambda, .)$  par rapport à  $\lambda$



$$\forall i \in \mathbb{N}, P^i(\lambda, \dots) = \frac{d^i D(\lambda)}{d\lambda^i} + f^* \frac{d^i N(\lambda)}{d\lambda^i}$$

Proposition.

-  $F(\cdot)$  définie en (III.11) est telle que :

a)  $\forall i \in \{1, \dots, s\} \quad \forall k \in \{1, \dots, s\}$

$$k \neq i \Leftrightarrow \lambda_i \neq \lambda_k.$$

$$\forall j \in \{1, \dots, n_i\}$$

les  $\alpha_i^k(\cdot)$  sont définis de proche en proche

$$(III.15) \quad \sum_{k=1}^j \alpha_i^k(\cdot) = - \frac{1}{(j-1)!} \cdot \left\{ \frac{(B)^*}{R_i(\lambda)} \right\}_{\lambda=\lambda_i}$$

b)  $\forall j \in \{1, \dots, t\} \quad m_j = 1 \quad v_j \neq 0$

$$\forall k \in \{1, \dots, t\} \quad j \neq k \Leftrightarrow \mu_j \neq \mu_k \text{ ou } |v_j| \neq |v_k|$$

$$(III.16) \quad \left\{ \begin{array}{l} \beta_{j1}^1(\cdot) = - \frac{1}{2v_j} [\text{Réel}(A)^* + \text{Im}(A)^*] \\ \beta_{j2}^1(\cdot) = \frac{1}{2v_j} [\text{Réel}(A)^* - \text{Im}(A)^*] \end{array} \right.$$

I.2.2.2 - Forme en flèche mince à racines complexes.

Nous avons présenté dans cette première partie le cas général de la forme de la matrice  $F(\cdot)$ . Lorsqu'une racine simple de  $R(\lambda)$  est complexe, deux représentations différentes sont envisageables pour les blocs de la diagonale associés à cette racine et à son conjugué :

- le type rotation à coefficients réels

$$\begin{pmatrix} \mu_j & v_j \\ -v_j & \mu_j \end{pmatrix}$$

- le type diagonal à coefficients complexes

$$\begin{pmatrix} \mu_j + i\nu_j & 0 \\ 0 & \mu_j - i\nu_j \end{pmatrix} \quad \text{où } i = \sqrt{-1}$$

La première possibilité revient à remplacer dans la matrice  $F(\cdot)$  (III.10) les blocs  $M_i$  par des scalaires, les blocs  $C_i$  restant inchangés.

Le calcul des termes non constants  $\beta_i(\cdot)$  s'effectue comme en (III.13), les termes  $\alpha_i(\cdot)$  sont des scalaires lorsque tous les coefficients complexes sont conjugués.

(III.17)

$$\alpha_i(\cdot) = - \left( \frac{P(\lambda, \cdot)}{R_i(\lambda)} \right)_{\lambda=\lambda_i}$$

Dans le second cas, la représentation est en flèche mince complexe.

Il vient :

(III.18)

$$F(\cdot) = \begin{bmatrix} \lambda_1 & & & & & & & & \alpha_1 \\ & \lambda_s & & & & & & & \alpha_s \\ & & \mu_1 + i\nu_1 & & & & & & \beta_{12} \\ & & & \mu_1 - i\nu_1 & & & & & \beta_{11} \\ & & & & \mu_t + i\nu_t & & & & \beta_{t2} \\ & & & & & \mu_t - i\nu_t & & & \beta_{t1} \\ & & & & & & & & \gamma(\cdot) \\ 1 & \text{---} & 1 & & 1 & & 1 & \text{---} & 1 & & 1 & & 1 & & 1 & & 1 & & 1 \end{bmatrix}$$

BU  
LILLE

Dans ce cas, on place sur la diagonale de la matrice les valeurs des racines réelles et complexes. Les termes non constants sont alors de la forme :

$$\alpha_i(.) = \delta_i + f^* \gamma_i \quad \left\{ \begin{array}{l} \gamma_i \in \mathbb{R} \text{ ou } \mathbb{C} \\ \delta_i \in \mathbb{R} \text{ ou } \mathbb{C} \end{array} \right.$$

$$\beta_j(.) = \delta'_j + f^* \gamma'_j \quad \left\{ \begin{array}{l} \gamma'_j \in \mathbb{C} \\ \delta'_j \in \mathbb{C} \end{array} \right.$$

Les  $\alpha_i(.)$  se calculent comme en (III.17).

Les  $\beta_i(.)$  sont dans ce cas complexes.

Les formules de  $\beta_j$  sont identiques à celles utilisées dans le cas de la flèche mince (III.6). Nous les avons toutefois développées de façon à faire apparaître la formulation employée dans les programmes.



$$\beta_j(.) = \begin{cases} \beta_{j2}(.) \\ \beta_{j1}(.) \end{cases}$$

$$(III.19) \quad \left\{ \begin{array}{l} \beta_{j2}(.) = \frac{-P(\mu_j + i\nu_j, .)}{R_j(\mu_j + i\nu_j)} \\ \beta_{j1}(.) = \frac{-P(\mu_j - i\nu_j, .)}{R_j(\mu_j - i\nu_j)} \end{array} \right.$$

avec

$$(III.20) \quad \left\{ \begin{array}{l} R_j(\mu_j + i\nu_j) = \prod_{i=1}^s (\lambda - \lambda_i) \cdot \prod_{\substack{\ell=1 \\ \ell \neq j}}^t ((\lambda - \mu_\ell)^2 + \nu_\ell^2) \cdot 2i\nu_j \\ R_j(\mu_j - i\nu_j) = \prod_{i=1}^s (\lambda - \lambda_i) \cdot \prod_{\substack{\ell=1 \\ \ell \neq j}}^t ((\lambda - \mu_\ell)^2 + \nu_\ell^2) \cdot (-2i\nu_j) \end{array} \right.$$

$$\gamma(.) = - a_{q-1} - b_{q-1} f^* - \sum_{i=1}^s \lambda_i - \sum_{i=1}^t \mu_i.$$

### I.2.3 - Application informatique.

Les formules de calcul présentées dans la partie (I.2) sont assez longues et fastidieuses à mettre en oeuvre manuellement en particulier pour un système d'ordre élevé ou bien même simplement dans le cas de matrice en flèche épaisse.

C'est pourquoi nous avons écrit les programmes correspondants, qui permettent de passer facilement du polynôme symbolique à diverses formes en flèche. Les arbres programmatiques seront présentés dans le dernier chapitre.

Les programmes donnent en sortie la valeur des termes de la dernière colonne et de la diagonale des matrices en flèche décrites dans le tableau (fig. III.1).

Sur ce tableau figurent :

1°) en trait plein ce qui est réalisé c'est à dire la détermination des formes en flèche :

- . mince            - réelle à une ou plusieurs non-linéarités
- complexe à une non-linéarité.
  
- . épaisse        - réelle à une seule non-linéarité.

2°) en traits pointillés les extensions futures possibles qui correspondent cependant à des cas exceptionnels de choix de  $R(\lambda)$ .

Le choix des racines  $\lambda_i$  peut être fait arbitrairement ou si l'utilisateur le désire de façon à ce que les racines  $\lambda_i$  vérifient une équation conditionnant les valeurs des termes non constants.

Il est à noter que le cas des  $\lambda_i$  complexes multiples n'a pas été envisagé dans cette étude.

Cette idée est développée dans la partie suivante.

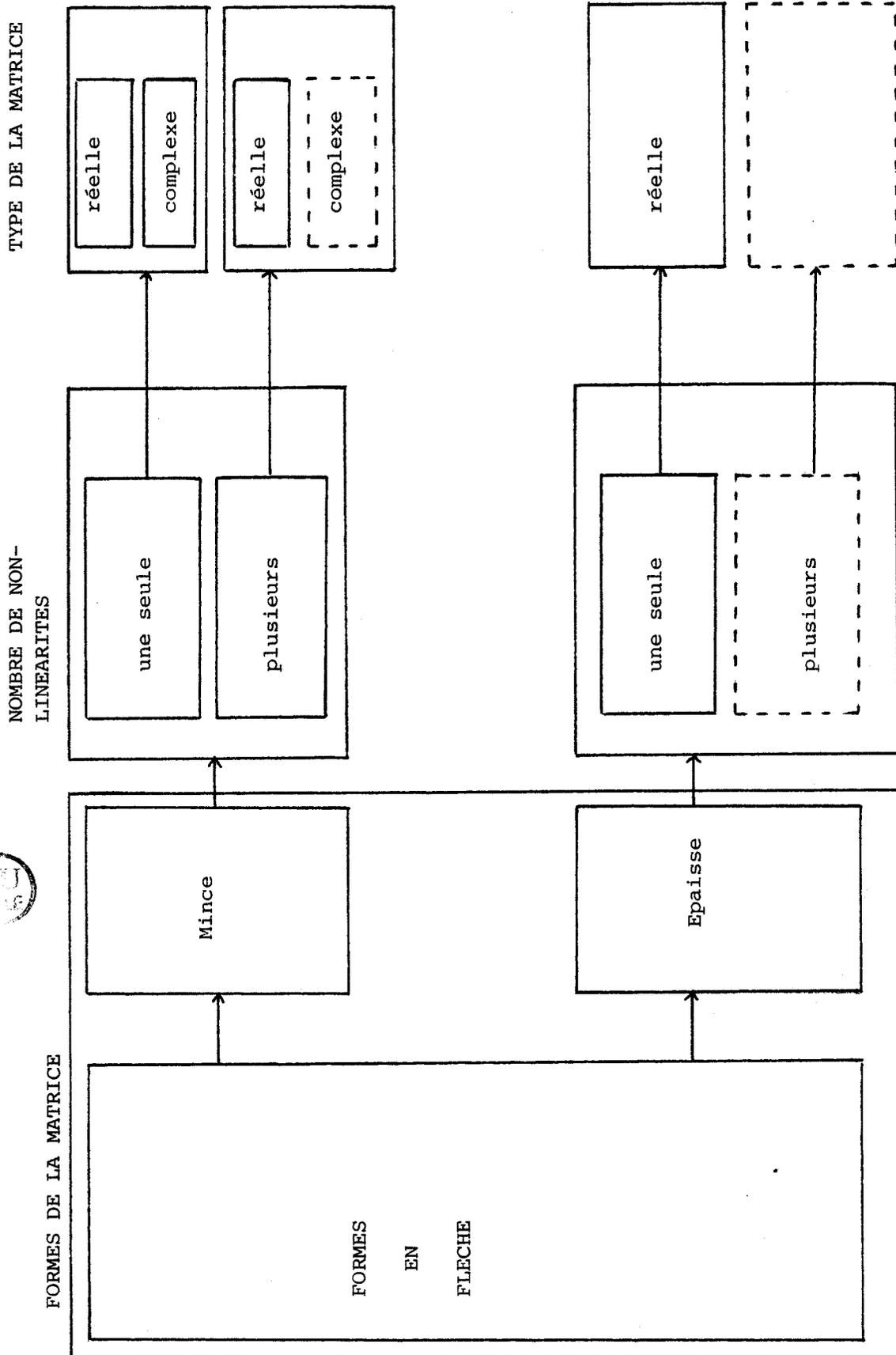


FIGURE III.1.

## II - CHOIX DES PARAMETRES DE MODELISATION.

Divers auteurs, en utilisant la forme en flèche mince, ont proposé de prendre des valeurs particulières pour  $\lambda_i$  de façon à obtenir les résultats suivants :

- /BENREJEB 1976/ : les  $\lambda_i$  sont choisis comme racines de  $D(\lambda)$  ou  $N(\lambda)$ . Dans ce dernier cas, le but principal recherché est d'obtenir la conjecture du linéaire lors de l'étude de stabilité.

- /DAUPHIN-TANGUY 1983/ : les  $\lambda_i$  sont solution de l'équation  $\bar{P}(\lambda) = 0$ ,  $\bar{P}$  étant le polynôme obtenu en fixant  $f^*$  à une valeur moyenne comprise entre 2 bornes  $\bar{f}$  et  $\underline{f}$ . Cette étude est proposée en relation avec l'application du critère de Gersgorine que nous détaillerons dans le chapitre suivant.

- /ROTELLA, ZAMBETTAKIS, RICHARD 1982/ : les  $\lambda_i$  sont choisis comme racines de  $D(\lambda) + a N(\lambda) = 0$ . De cette façon, les  $q-1$  premiers termes de la dernière colonne de la matrice sont constants ou proportionnels à un même terme non constant. Les  $\lambda_i$  sont les  $q-1$  racines d'un polynôme de degré  $q$  ou  $q-1$ . Ceci permet l'utilisation de fonctions de Ljapunov de type quadratique plus intégrale.

Ce résultat établi tout d'abord pour le cas des matrices en forme en flèche a été généralisé au cas de la forme en flèche épaisse.

### II.1 - Définition d'une problématique.

Rappelons tout d'abord les résultats démontrés dans /ROTELLA, ZAMBETTAKIS, RICHARD 1982/ concernant le choix des  $\lambda_i$  solutions d'un polynôme de degré  $q$  ou  $q-1$ .

Il vient la proposition suivante :

#### Proposition.

Soit  $a$ ,  $a \in \mathbb{R}$ , tel que le polynôme  $Q(\lambda) = D(\lambda) + a N(\lambda)$  ait au moins  $q-1$  racines réelles distinctes :

$$\mu_1, \dots, \mu_{q-1}$$

alors le système décrit précédemment (III.7) peut être représenté en régime autonome par

$$\dot{y} = A(.) y$$

avec

$$(III.21) \quad A(.) = \begin{bmatrix} \mu_1 & & & -k_1 g^* \\ & 0 & & -k_i g^* \\ & & \ddots & -k_{q-1} g^* \\ & 0 & & \mu_{q-1} \\ 1 & \dots & 1 & \mu_q - b_{q-1} g^* \end{bmatrix}$$

où

$$g^* = f^* - a$$

$$\forall i \in \{1, \dots, q-1\} \quad k_i = \left[ \frac{N(\mu_i) (\lambda - \mu_i)}{R(\lambda)} \right]_{\lambda = \mu_i}$$

$\mu_q = q^{\text{ième}}$  racine de  $Q(\lambda)$ .

La matrice  $A(.)$  ainsi obtenue est une matrice en flèche mince dont les éléments diagonaux  $\mu_i$  sont les zéros du polynôme  $D(\lambda) + a N(\lambda)$ , et dont les éléments de la dernière colonne sont tous proportionnels au terme  $g^*$  défini ci-dessus.

Les  $\mu_i$  dépendent directement de la valeur du paramètre  $a$ . Le choix des coefficients  $\lambda_i$  revient donc maintenant au choix du réel  $a$  (fini ou infini).

Du point de vue de l'étude de stabilité (cf. chapitre IV) il est intéressant que les  $\lambda_i$  aient des parties réelles strictement négatives.

Si on cherche une représentation en flèche mince, on a alors à résoudre la problématique P suivante :

(P.) - Quelles sont les valeurs de  $a$  telles que  $D(\lambda) + a N(\lambda) = 0$  ait au moins  $q-1$  racines  $\lambda_i$  réelles négatives distinctes ?

Dans le cas où (P) n'a pas de solution, on peut chercher de façon plus générale à résoudre  $P'$  :

(P'.) - Quelles sont les valeurs de  $a$  telles que  $D(\lambda) + a N(\lambda)$  ait au moins  $q-1$  racines à parties réelles négatives ?

Cette problématique (P') plus simple peut être traitée par l'utilisation du tableau de Routh traditionnel. Résoudre (P') peut conduire à une forme en flèche mince intéressante mais à coefficients complexes contrairement à (P).

Différentes méthodes de détermination de  $a$  ont été mises en oeuvre pour résoudre (P).

La première immédiate consiste à donner une valeur arbitraire à  $a$  et à chercher les racines de  $D(\lambda) + a N(\lambda) = 0$ . S'il n'existe pas au moins  $q-1$  racines réelles distinctes, on attribue à  $a$  une autre valeur et ainsi de suite.

Cette méthode risque d'être longue et il est préférable d'utiliser le lieu des racines ou lieu d'Evans /EVANS 1948/ afin de déterminer un secteur de validité du paramètre  $a$  vérifiant (P). /ROTELLA, ZAMBETTAKIS, RICHARD 1982/.

Pour ce faire, il est possible d'utiliser des programmes écrits dans des modules d'aide à l'analyse des systèmes et qui tracent rapidement les lieux d'Evans /PAILLET 1981/, /PAN, CHAO 1978/.

Un des inconvénients de cette méthode graphique est qu'elle manque parfois de précision, notamment pour les valeurs de  $a$  correspondant aux points de séparation des racines du lieu d'Evans.

/ZAMBETTAKIS, RICHARD, ROTELLA 1984/ ont envisagé l'utilisation d'une méthode algébrique, que nous emploierons par la suite et qui va être détaillée dans la partie suivante.

## II.2 - Résolution algébrique de la problématique P.

Deux solutions particulières de P et P' ont été proposées auparavant par des auteurs différents. Nous les rappelons pour commencer cette partie.



$$(III.23) \quad A(.) = \left[ \begin{array}{cccc} z_1 & & & \delta_1 \\ & z_{q-1} & & \delta_{q-1} \\ & & & \\ 1 & \text{---} & 1 & -a_{q-1} - b_{q-1} f^* - \sum_{i=1}^{q-1} z_i \end{array} \right]$$

Ce cas particulier a été présenté dans /BENREJEB 1976/. Il permet de modéliser un système à l'aide d'une flèche mince dont les  $q-1$  premiers termes de la dernière colonne sont constants.

Si le numérateur  $N(\lambda)$  admet des racines complexes ou multiples, on réalise une représentation en flèche épaisse à termes hors diagonaux constants dans la dernière colonne.

### II.2.3 - Cas général : $\lambda_i$ racines de $D(\lambda) + a N(\lambda) = 0$ .

Dans /ZAMBETTAKIS, RICHARD, ROTELLA 1984/ est exposée une méthode algébrique permettant de résoudre le problème (P).

Quelles sont les valeurs de  $a$  tellesqu'il soit possible de trouver  $q$  ou  $q-1$  réels  $\lambda_i$  distincts et négatifs solution de  $D(\lambda) + a N(\lambda) = 0$  ?

Nous énonçons dans le paragraphe suivant le théorème nécessaire à la réalisation de cette méthode.

#### II.2.3.1 - Théorème de Sturm.

Notre étude est basée sur le théorème de Sturm /DURAND 1960/.

Soit un polynôme  $S_0(\lambda)$  de degré  $n$  n'ayant que des racines distinctes et  $S_1(\lambda)$  sa dérivée. En divisant  $S_0$  par  $S_1$  et en prenant le reste changé de signe on obtient un polynôme  $S_2(\lambda)$ . Une opération analogue divisant  $S_1$  par  $S_2$  et prenant le reste changé de signe conduit à un polynôme  $S_3(\lambda)$ ... etc. On établit ainsi une relation de récurrence :

$$S_{j-1} = S_j q_j - S_{j+1} \text{ portant sur une suite } S_0, \dots, S_n.$$

Les polynômes  $S_j$  ont les propriétés suivantes.

1°)  $S_n$  est constant, non nul, si  $S_0$  et  $S_1$  n'ont pas de racines communes.

2°) Deux polynômes consécutifs  $S_j(\lambda)$  et  $S_{j+1}(\lambda)$  ne peuvent s'annuler simultanément.

3°) Si on a  $S_j(\alpha) = 0$  pour une valeur  $\alpha$  de la variable  $\lambda$  alors on a  $S_{j-1} \cdot S_{j+1} < 0$ .

Énonçons maintenant le théorème :

Théorème. (III.24)

Si  $N(\alpha)$  et  $N(\beta)$  sont les nombres de changements de signes de la suite pour  $\lambda = \alpha$  et  $\lambda = \beta$ , le nombre des racines réelles du polynôme  $S_0(\lambda) = 0$  comprises entre  $\lambda = \alpha$  et  $\lambda = \beta$  est  $N(\alpha) - N(\beta)$ .

On utilisera ce théorème pour déterminer la position et la nature des racines d'un polynôme.

Trois cas particuliers sont envisageables.

1°) Si la suite est complète et si le coefficient du terme de plus haut degré de chaque  $S_j$  est positif, toutes les racines sont réelles et distinctes.

$$N(-\infty) - N(+\infty) = n - 0 = n$$

et il n'y a pas de racines multiples puisque  $S_n \neq 0$ .

2°) Si, en plus des conditions précédentes, les termes indépendants de  $\lambda$  dans les  $S_j$  sont alternativement (+) et (-) toutes les racines sont positives.

$$N(0) - N(+\infty) = n - 0$$

3°) La condition pour que toutes les racines soient négatives est que ces termes indépendants de  $\lambda$  soient de même signe.

Dans les calculs qui mènent aux fonctions  $S_j$  on est amené à diviser un polynôme de degré  $p$  par un polynôme de degré  $p-1$ , le quotient est de degré 1 et le reste de degré  $p-2$  au plus.

E.J. Routh a mis au point un mode opératoire très simple et une disposition commode des calculs. Nous proposons de généraliser cette disposition bien connue en vue de résoudre le problème (P) énoncé dans le paragraphe (II.1).

### II.2.3.2 - Tableau de Routh modifié.

On dispose sur la première ligne les coefficients  $a_i$  du polynôme étudié  $P_0(\lambda)$  et ceux  $b_i$  de sa dérivée  $P_1(\lambda)$  sur une ligne au-dessous.

$$P_0(\lambda) = \sum_{i=0}^n a_i \lambda^i, P_1(\lambda) = \sum_{i=0}^{n-1} b_i \lambda^i \text{ où } b_i = (i+1) \cdot a_{i+1}$$

$P_0$	$a_n$	$a_{n-1}$	$a_{n-2}$	...	$a_i$	...	$a_1$	$a_0$
$P_1$	$b_{n-1}$	$b_{n-2}$	$b_{n-3}$	...	$b_{i-1}$	...	$b_0$	

A partir de ces deux lignes, on en calcule une troisième ayant  $n$  termes  $c'_i$  par les produits croisés des lignes précédentes suivant la formule ci-dessous.

$$(III.25) \quad c'_i = \frac{b_{n-1} a_i - a_n b_{i-1}}{b_{n-1}}$$

qui définit le polynôme  $P_2$ , lorsque  $i$  varie de 0 à  $n-1$ .

On obtient ensuite une quatrième ligne  $c_i$  ( $n-1$  termes) en réalisant le produit en croix des coefficients de  $P_2$  et  $P_1$ , soit

$$(III.26) \quad c_i = \frac{c'_{n-1} b_i - b_{n-1} c'_i}{b_{n-1}}$$

qui définit  $P_3$ ,  $i$  variant de 0 à  $n-2$ .

Par analogie au tableau de Routh, le tableau formé est appelé tableau de Routh modifié /ZAMBETTAKIS, RICHARD, ROTELLA 1984/ ou encore dans /DURAND 1961/ schéma de Routh du premier type. Il se présente comme suit :

1ère ligne : coefficients  $a_i$  de  $P_0$ .

2ème ligne : coefficients  $b_i$  de  $P_1$ .

3ème ligne : coefficients  $c_i$  de  $P_3$ .

4ème ligne : coefficients  $d_i$  de  $P_5$ , déduits de  $c_i$  et  $b_i$  en passant par  $d_i$  suivant les règles (III.25) et (III.26).

Notons que les lignes  $c_i$ ,  $d_i$ ...etc correspondant aux polynômes  $P_k$ ,  $k$  pair, sont des lignes de calculs intermédiaires et ne figurent pas dans le tableau final.

Les calculs se font jusqu'à ce que l'on obtienne un polynôme  $P_i$  à coefficients nuls. La dernière ligne correspond donc à la première ligne précédent un polynôme à coefficients tous nuls.

Le tableau de Routh ainsi constitué possède  $n$  lignes et se présente sous une forme triangulaire comme suit :

$l_1$	$a_n$	$a_{n-1}$ ...	$a_0$
$l_2$	$b_{n-1}$	$b_{n-2}$ ...	$b_0$
$l_3$	$c_{n-2}$	$c_{n-3}$ ...	$c_0$
$l_m$	$y_{n-m+1}$	...	$y_0$

la série  $(a_n, b_{n-1}, c_{n-2} \dots y_{n-m+1})$  constitue la première ligne du tableau et la série  $(a_0, b_0 \dots y_0)$  sa dernière diagonale.

Ce sont ces deux séries que l'on utilisera pour déterminer le nombre de racines négatives distinctes de polynôme  $P_0(\lambda)$  étudié.

#### II.2.3.2.1 - Racines du polynôme.

La construction du tableau de Routh modifié à partir de polynôme étudié  $P_0$  et de sa dérivée  $P_1$  nous donne ligne après ligne les coefficients de polynômes constituant une série de Sturm présentée dans le paragraphe III.2.3.1  $P_0, P_1 \dots P_{n-1}$ .

On utilise le théorème de Sturm et ses conséquences

$$(III.27) \quad N(-\infty) - N(+\infty) = R : \text{nombre de racines réelles distinctes}$$

$$N(-\infty) - N(0) = N : \text{nombre de racines réelles négatives}$$

De plus, si  $P_{m-1}$  est le dernier polynôme à coefficients non nuls, nous avons également la propriété suivante :

$$(III.28) \quad N(-\infty) = m - 1 - N(+\infty).$$

En remplaçant  $N(0)$  par  $L$  nombre de changements de signe dans la diagonale et  $N(+\infty)$  par  $F$  nombre de changements de signe dans la première colonne, à partir de (III.27) et (III.28) il vient

$$(III.29) \quad \begin{aligned} N &= m - 1 - F - L \\ R &= m - 1 - 2F \end{aligned}$$

#### II.2.3.3. - Application à la résolution de $D(\lambda) + a N(\lambda) = 0$ .

Dans les paragraphes précédents, nous avons montré comment il est possible à partir d'un tableau de Routh modifié de déterminer le nombre de racines réelles distinctes négatives d'un polynôme donné.

Le polynôme étudié est un polynôme à coefficients réels constants. Nous allons dans ce paragraphe appliquer les résultats précédents au polynôme  $D(\lambda) + a N(\lambda)$  avec  $a$  paramètre réel et :

$$N(\lambda) = \sum_{i=0}^{q-1} b_i \lambda^i$$

$$D(\lambda) = q + \sum_{i=0}^{q-1} a_i \lambda^i$$

$$D(\lambda) + a N(\lambda) = \lambda^q + \sum_{i=0}^{q-1} (a_i + ab_i) \lambda^i$$

Le calcul de la dérivée de  $D(\lambda) + a N(\lambda)$  donne :

$$\frac{d(D(\lambda) + a N(\lambda))}{d\lambda} = q \lambda^{q-1} + \sum_{i=1}^{q-1} i (a_i + ab_i) \lambda^{i-1}$$

Les deux premières lignes du tableau de Routh sont les suivantes.

$\ell_1$	1	$a_{q-1} + ab_{q-1}$	.....	$a_1 + ab_1$	$a_0 + ab_0$
$\ell_2$	q	$(q-1)(a_{q-1} + ab_{q-1})$	.....	$a_1 + ab_1$	

Les coefficients de ces deux premières lignes ne sont plus des scalaires comme dans le cas d'un polynôme à coefficients constants mais fonctions affines du paramètre  $a$ .

En effectuant les produits croisés des deux premières lignes suivant les formules (III.25) et (III.26) on obtient pour les lignes suivantes des coefficients qui sont des quotients de polynômes en  $a$ . Nous allons montrer comment le dénominateur de ces quotients n'intervient pas dans la recherche du nombre de  $\lambda_i$  et exposerons le tableau permettant de calculer le degré de chacun des polynômes en  $a$ .

La formule (III.25) conduit à l'expression des  $c'_i$  qui est la suivante :

$$(III.30) \quad c'_i = i (a_{q-i} + a b_{q-i}) / q \quad i \text{ varie de } 1 \text{ à } q$$

L'expression de la troisième ligne est alors :

$$(III.31) \quad c_i = \frac{c'_1(q-i) (a_{q-i} + ab_{q-i}) - c'_{i+1} \cdot q}{q} \quad i \text{ varie de } 1 \text{ à } q-1$$

$c'_i$  est un polynôme en  $a$  du premier degré.

$c_i$  est un polynôme en  $a$  du second degré divisé par  $q^2$ .

Les degrés respectifs des polynômes sont donnés dans le tableau suivant. Nous faisons également figurer le degré des polynômes dans les lignes de calcul intermédiaires.

Lignes définitives

1	0	1	.....	1
2	0	1	.....	1
3	2	2	.....	2
4	5	5	.....	5
5	12	12	.....	12
6	29	.....	29	
	et ainsi de suite			

Lignes intermédiaires.

	2'	1	.....	1
	3'	3	.....	3
	4'	7	.....	7
	5'	17	.....	17

la formule de récurrence est la suivante :

$$(III.32) \quad \begin{cases} d_i = d_{(i-1)'} + d_{i-1} \\ d_{i'} = d_i + d_{i-1} \end{cases}$$

où  $d_i$  représente le degré du polynôme de la ligne  $i$  et  $d_{i'}$  le degré du polynôme de la ligne intermédiaire  $i'$ . La relation (III.32) est vérifiée pour  $i \geq 3$  et  $i' \geq 3$  en posant  $d_2 = 1$ .

Les calculs des différentes lignes, dont les coefficients sont des polynômes en  $a$ , sont effectués suivant les règles (III.25) et (III.26). Le degré de chacun des polynômes est donné par (III.32).

Nous remarquons d'autre part que pour les lignes à retenir, le polynôme de degré  $a$  est divisé par un autre polynôme de degré  $a$  qui est toujours un carré, et donc positif.

Ainsi l'expression de ce dénominateur n'intervient pas dans la recherche du nombre de changements de signe dans la première colonne et la diagonale.

Pour connaître le nombre de changements de signe  $F$  dans la première colonne et le nombre de changements de signe  $L$  dans la diagonale, il faut étudier le signe de chacun des coefficients contenus dans cette première colonne et dans la diagonale. Puisque chaque coefficient est un polynôme en  $a$ , nous étudions le signe de ce polynôme suivant la valeur de  $a$ , ce qui risque d'être fastidieux.

Nous devons toutefois tenir compte de l'ordre du système étudié. En effet, si le système est d'ordre  $q$  le tableau de Routh possède  $q+1$  lignes ou  $q$  lignes dans le cas d'une racine double.

La formule (III.31) devient :

- dans le premier cas

$$N = q - F - L$$

- dans le second cas

$$N = q - 1 - F - L$$

Pour obtenir  $q$  ou  $q-1$  valeurs de  $\lambda_i$  trois solutions sont possibles :

$$(m, F, L) \in \{(q+1, 0, 0), (q+1, 0, 1), (q, 0, 0)\}$$

Finalement, la recherche du nombre de changements de signe L et F est inutile. Le problème revient à déterminer la valeur de a telle que  $F = 0$  et  $L = 0$  ou 1.

Cette étude a été réalisée par moyens informatiques que nous détaillerons dans le dernier chapitre et donne les intervalles possibles pour a de façon à obtenir q ou q-1 paramètres  $\lambda_i$  distincts et négatifs.

#### II.2.3.4 - Exemples.

##### II.2.3.4.1 - Tableau de Routh modifié.

Soit le polynôme  $P(\lambda) = \lambda^3 + 8\lambda^2 + 20\lambda + 17$  d'après le paragraphe III.2.3.2, la formulation du tableau de Routh modifié est la suivante :

ligne				
1	1	8	20	17
2	3	16	20	
2'	8/3	40/3	17	
3	8/9	7/9		
3'	107/8	20		
4	$\frac{531}{64}$			

En utilisant les relations III.29, on obtient :

$$N = 4 - 1 - 1 - 1 = 1$$

$$R = 4 - 1 - 2 = 1$$

On en déduit que le polynôme  $P(\lambda)$  admet une seule racine réelle négative.

### II.2.3.4.2 - Application à la résolution de

$$\underline{D(\lambda) + a N(\lambda) = 0 .}$$

Considérons le système décrit fig. (III.2)

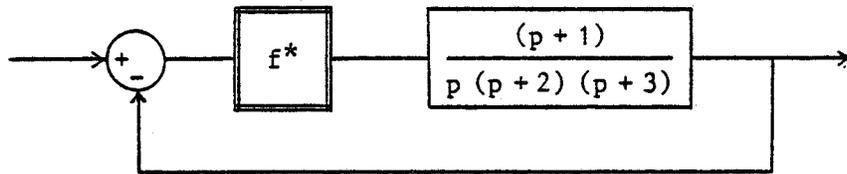


Fig. (III.2)

$$D(p) = p(p+2)(p+3)$$

$$N(p) = (p+1)$$

$$D(p) + a N(p) = p^3 + 5p^2 + p(6+a) + a$$

Construisons le tableau de Routh modifié à partir de cette expression .

ligne				
1	1	5	6+a	a
2	3	10	6+a	
2'	$\frac{5}{3}$	$\frac{2(6+a)}{3}$	a	
3	$\frac{14-6a}{5}$	$\frac{30-4a}{5}$		
3'	$\frac{50-48a}{14-6a}$	6+a		
4	$\frac{(-36a^3 - 144a^2 - 828a + 324)}{(14-6a)^2}$			

$$m = q+1 = 4.$$

Pour que le polynôme  $D + aN$  admette  $q$  ou  $q-1$  racines distinctes négatives,  $(F, L)$  doit être égal à  $(0, 0)$  ou  $(0, 1)$ .

Le nombre de changements de signe dans la première colonne doit être nul. On étudie le signe de chacun des coefficients de la première colonne. Ces coefficients sont des polynômes en  $a$  et il faut déterminer leur variation suivant  $a$ .

première colonne	$\infty$	0.418	2.333	$-\infty$
1	+	+	+	+
3	+	+	+	+
$(14 - 6a)/5$	+	+	0	-
$- 36 a^3 - 144 a^2 - 828 a + 324$	+	0	-	-
F : nombre de changements de signe	0	1	1	



Le nombre de changements de signe  $F$  est nul si  $a$  est inférieur à 0.418.

Etudions maintenant le nombre de changements de signe dans la diagonale.

diagonale	$\infty$	-6	0	0.418	7.5	$-\infty$
a	-	-	0	+	+	+
6+a	-	0	+	+	+	+
30-4a	+	+	+	+	0	-
$-36a^3 - 144a^2 - 828a + 324$	+	+	+	0	-	-
L : nombre de changements de signes	1	1	0	1	1	

Quelle que soit la valeur de a, il y a 0 ou 1 changement de signe dans la diagonale.

Seule la condition sur F est à vérifier.

Il vient finalement :

$D(\lambda) + a N(\lambda)$  admet

- q racines  $\lambda_i$  réelles négatives

si  $F = L = 0$  c'est à dire  $0 < a < 0.418$

- q-1 racines  $\lambda_i$  réelles négatives

si  $F = 0$  et  $L = 1$  c'est à dire  $a < 0$ .

pour  $a = 0,25$  il vient :

$$\lambda_1 = -0,04$$

$$\lambda_2 = -2,16$$

$$\lambda_3 = -2,79.$$

La forme en flèche correspondante est: (en choisissant comme racines  $\lambda_1, \lambda_2$  et  $\lambda_3$ )

$$\begin{pmatrix} -2.16 & 0 & -0.461 & +1,841 f^* \\ 0 & -2,79 & -0,7347 & -2,841 f^* \\ 1 & 1 & -0,05 & \end{pmatrix}$$

## CONCLUSION.

Dans ce chapitre, nous avons présenté les formes en flèche principales utilisées dans l'étude des systèmes du type Lur'e Postnikov.

Nous avons rappelé l'importance du choix des racines de  $R(\lambda)$ , paramètres de la forme en flèche et expliqué comment il est possible de mener ce choix en relation avec la résolution de l'équation  $D(\lambda) + a N(\lambda) = 0$ ,  $a$  étant un réel fini ou infini.

Pour cela nous avons eu recours au tableau de Routh modifié qui fournit la valeur du paramètre  $a$  permettant d'obtenir  $q$  ou  $q-1$  valeurs  $\lambda_i$  de  $\lambda$  afin de représenter le système sous forme de flèche mince.

Calculés à partir des données qui sont les coefficients du polynôme  $D(\lambda) + a N(\lambda)$ , les résultats suivants sont fournis :

- intervalles de choix admissibles de  $a$ ,
- expression des paramètres  $\lambda_i$  pour  $a$  fixé,
- forme en flèche correspondante.

La réalisation informatique sera détaillée dans le chapitre V.

CHAPITRE IV

ANALYSE DES SYSTEMES  
A PARTIR DE LA MODELISATION  
EN FLECHE.

## INTRODUCTION.

Comme nous l'avons déjà souligné la forme en flèche est particulièrement bien adaptée à l'analyse des systèmes, notamment ceux de type Lur'e Postnikov monovariante (mLP).

Il est en effet possible d'obtenir par application de critères usuels sur la forme en flèche des résultats supérieurs à ceux obtenus sur une forme matricielle quelconque.

Ainsi, par exemple, la séparation des dynamiques d'un système non linéaire dont la matrice est au départ mal conditionnée peut être réalisée si, par changement de base, on transforme cette matrice en une forme en flèche /DAUPHIN-TANGUY 1983/.

La séparation des dynamiques du système est alors menée sur une forme en flèche à partir de la méthode des cercles de Gershgorine /DEIF 1982/. C'est par cette application que débute le chapitre. La détermination de ces cercles est faite sur divers exemples. On montre alors que la valeur du rayon de chaque cercle dépend des paramètres  $\lambda_i$  de représentation et de la non linéarité  $f^*$ . Il est d'une part possible de choisir des paramètres  $\lambda_i$  réels négatifs ou complexes et d'autre part de déterminer une valeur du paramètre  $a(D+aN \neq 0)$  permettant d'obtenir la meilleure séparation des dynamiques du système, la variation de  $f^*$  étant fixée.

Dans une seconde partie les méthodes d'étude de la stabilité, exposées dans le premier chapitre, sont mises en oeuvre sur les formes en flèche décrites dans le chapitre précédent.

Ces résultats sont ensuite appliqués à l'élaboration d'un critère définissant un modèle réduit /ZAMBETTAKIS, RICHARD, ROTELLA 1984/

Enfin dans une dernière partie sont exposées les règles de calcul nécessaires à une réalisation informatique de l'analyse des systèmes de type Lur'e Postnikov. Les programmes créés seront détaillés dans le dernier chapitre.

## I - APPLICATION DE LA FORME EN FLECHE AUX SYSTEMES A DEUX DYNAMIQUES.

### I.1 - Méthode géométrique de séparation de dynamiques. Cas linéaire.

Nous énonçons, tout d'abord, les résultats concernant les systèmes linéaires. Ces résultats sont principalement issus de /DAUPHIN-TANGUY 1983/. Il est intéressant de pouvoir mettre en évidence deux dynamiques différentes pour un même système.

Ceci est réalisé en effectuant un découplage temporel partie lente/partie rapide par *la méthode des perturbations singulières*.

La principale difficulté de cette méthode réside dans le choix des composantes du vecteur état, qui forment la partie lente et la partie rapide du système. Ce choix repose sur la détermination approchée des valeurs propres du système.

Différentes techniques ont permis de localiser les valeurs propres dans le plan complexe.

Nous en rappelons une, géométrique basée sur la détermination des *cercles de Gershgorine* /DEIF 1982/.

Ainsi, dans le cas de systèmes linéaires il vient la définition suivante :

#### Définition.

Si  $A$  est la matrice caractérisant un système, d'éléments  $a_{ij}$  ;  $i, j = \{1, \dots, q\}$  alors les valeurs propres de  $A$  appartiennent au domaine formé par l'union des intérieurs des cercles de rayons respectifs  $r_i$  est de centres  $a_{ii}$  définis par :

$$(IV.1) \quad |\lambda - a_{ii}| = \sum_{\substack{j=1 \\ j \neq i}}^q |a_{ij}| = r_i$$

Ces cercles appelés cercles de Gershgorine (d'après le théorème de Gershgorine) sont définis ici en ligne mais peuvent également l'être en colonne par :

$$(IV.2) \quad |\lambda - a_{ii}| = \sum_{\substack{j=1 \\ j \neq i}}^q |a_{ji}|$$

La définition de ces cercles permet notamment une analyse de la séparation des dynamiques d'un système :

Propriété.

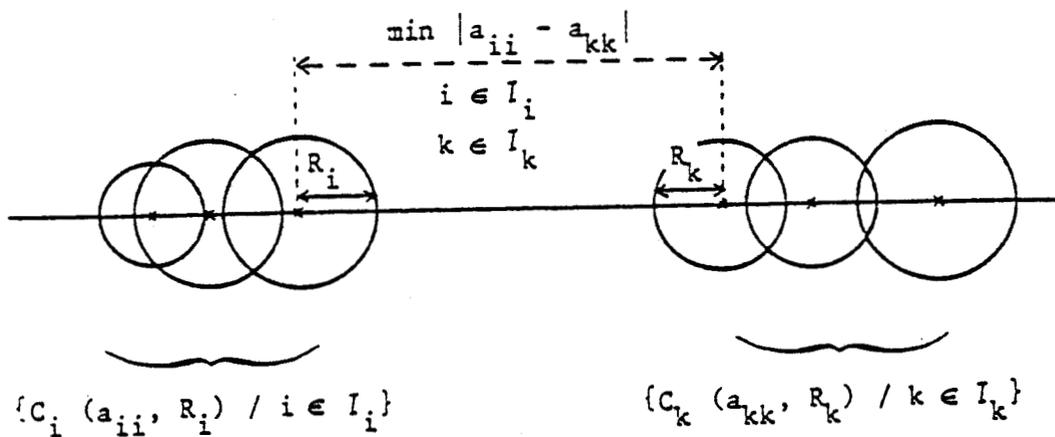
Si les cercles  $C_i(a_{ii}, r_i)$  et  $C_k(a_{kk}, r_k)$  sont tels que  $|a_{ii} - a_{kk}| \gg (r_i + r_k)$   $\forall i \in I_i$  et  $\forall k \in I_k$ , avec  $I_i \cap I_k = \emptyset$ , alors la matrice possède deux ensembles séparables de valeurs propres, définissant deux dynamiques.

La précision du résultat dépend du rapport  $\epsilon$ :

$$(IV.3) \quad \epsilon = \max_{\substack{i \in I_i \\ k \in I_k}} \frac{r_i + r_k}{|a_{ii} - a_{kk}|}$$

avec  $\epsilon$  inférieure à 1.

Cette propriété est illustrée par la fig. (IV.1)



Les cercles de Gershgorine peuvent être regroupés en deux domaines disjoints, associés à deux sous-ensembles disjoints de valeurs propres, d'où l'existence de deux dynamiques distinctes.

Une matrice en flèche mince représentative d'un système linéaire à ses termes tous constants et se définit de la même façon qu'en non linéaire :

(IV.4)

$$A_1 = \begin{bmatrix} \lambda_1 & & & -\frac{P(\lambda_1)}{R_1(\lambda_1)} \\ & \ddots & & \vdots \\ & & \lambda_{q-1} & -\frac{P(\lambda_{q-1})}{R_{q-1}(\lambda_{q-1})} \\ 1 & \text{---} & 1 & \text{---} & 1 & -\sum_{i=1}^{q-1} \lambda_i - a_{q-1} \end{bmatrix}$$

Il est possible par changement de base diagonale de matrice  $R = \text{diag}(r_1, r_2, \dots, r_{q-1}, 1)$  de mettre la matrice  $A_1$  sous une forme semblable :  $A_2 = R^{-1} A R$ .

(IV.5)

$$A_2 = \begin{bmatrix} \lambda_1 & & & -\frac{1}{r_1} \frac{P(\lambda_1)}{R_1(\lambda_1)} \\ & \ddots & & \vdots \\ & & \lambda_{q-1} & -\frac{1}{r_{q-1}} \frac{P(\lambda_{q-1})}{R_{q-1}(\lambda_{q-1})} \\ r_1 & \text{---} & r_{q-1} & -\sum_{i=1}^{q-1} \lambda_i - a_{q-1} \end{bmatrix}$$

De façon à optimiser les rayons des cercles de Gershgorine, c'est à dire séparer le mieux possible les dynamiques du système /DAUPHIN-TANGUY 1983/ on définit la matrice  $R$  de changement de base comme suit :

(IV.6)

$$r_i = \sqrt{\frac{|P(\lambda_i)|}{|R_i(\lambda_i)|}} \quad i = 1, \dots, q-1$$

La matrice  $A_2$  prend alors la forme suivante  $A_3$  :

$$(IV.7) \quad A_3 = \begin{bmatrix} \lambda_1 & & & \text{sgn}(-P(\lambda_1))r_1 \\ & \lambda_{q-1} & & \text{sgn}(-P(\lambda_{q-1}))r_{q-1} \\ & & & \sum_{i=1}^{q-1} \lambda_i - a_{q-1} \\ \text{sgn}(R_1(\lambda_1))r_1 \dots \text{sgn}(R_{q-1}(\lambda_{q-1}))r_{q-1} & & & \end{bmatrix}$$

Les rayons des cercles de Gershgorine sont donnés par  $r_i$  pour  $i=1, \dots, q-1$  dans le cas du linéaire

## I.2 - Applications aux systèmes non linéaires.

Pour un système non linéaire, l'étude est menée suivant le déroulement exposé dans le cas d'un système linéaire avec l'objectif supplémentaire suivant : déterminer le plus grand domaine de variation des valeurs propres obtenues lorsqu'il y a variation des termes non constants.

Dans le cas de matrices bien conditionnées la méthode définie précédemment s'applique sur une matrice majorante du système non linéaire.

Les cercles de Gershgorine constituent dans ce cas un faisceau de cercles à centres variables ( $-a_{ii}(\cdot)$ ) et de rayons majorés fixes  $r_i = \sum_{\substack{j=1 \\ j \neq i}}^n \max |a_{ij}(\cdot)|$

Dans le cas d'un système mLP dont la matrice est malconditionnée (de type compagnon par exemple), l'application de la méthode des cercles de Gershgorine s'effectue en trois étapes.

- 1°) - On cherche la valeur de  $a$  telle que l'équation  $D(\lambda) + aN(\lambda)$  admette  $q-1$  racines distinctes  $\lambda_i$  à partie réelle négative (problématique (P) cf § II.2 chap. III). Cette recherche est effectuée suivant la procédure décrite dans le paragraphe (II.2.3) du chapitre III.
- Si on ne peut déterminer  $a$ , les paramètres de modélisation  $\lambda_i$  sont choisis arbitrairement.

2°) - On détermine une représentation sous forme en flèche mince réelle (III.5) ou en flèche mince complexe (III.18).

3°) - On détermine les cercles de Gershgorine. Les éléments non constants de la matrice sont isolés dans la dernière colonne sous la forme d'une fonction linéaire de  $f^*$  :  $\delta_i + \gamma_i f^*$ . Leurs valeurs extrêmes sont donc obtenues aux bornes de l'intervalle de variation de  $f^*$ . L'ensemble des cercles est alors formé de :

- (q-1) cercles à centre et rayon fixes

- un faisceau de cercles à centre variable et rayon fixe

Suivant la valeur des paramètres  $\lambda_i$  et par conséquent le type de la matrice en flèche, les centres des cercles peuvent être situés :

- sur l'axe réel dans le cas d'une flèche mince réelle

- dans le plan complexe dans le cas d'une flèche mince complexe.

La valeur des rayons  $r_i$  maximaux est définie en analogie avec l'étude menée sur le cas linéaire par :

$$(IV.8) \quad r_i = \max_{f^*} \sqrt{\left| \frac{P(\lambda_i, \dots)}{R_i(\lambda_i)} \right|} \quad \text{pour } i = 1, \dots, q-1$$

$$(IV.9) \quad \varepsilon = \max \frac{r_i + r_k}{|\lambda_i - \lambda_k|}$$

Cette méthode, conduisant aux cercles de Gershgorine dans le cas de systèmes non linéaires de type mLP, est détaillée dans la partie qui suit.

On suppose dans un premier temps qu'il est possible de trouver  $a$  tel que  $D(\lambda) + aN(\lambda) = 0$  admette  $q$  ou  $q-1$  racines réelles négatives, dans un second temps, on choisit arbitrairement les paramètres de modélisation  $\lambda_i$ .

### I.2.1 - Paramètres solutions de $D(\lambda) + aN(\lambda) = 0$

Comme nous l'avons expliqué dans le chapitre III, on détermine non pas une solution unique pour  $a$  mais un intervalle de valeurs solutions.

D'autre part, pour une valeur de  $a$  précise, il est possible d'obtenir  $q$  valeurs de  $\lambda_i$  négatives pour un système d'ordre  $q$ . Il est alors nécessaire pour modéliser de choisir  $q-1$  valeurs parmi ces  $q$  termes.

A partir d'un exemple, nous montrons comment on sélectionne la valeur de  $a$  et celles des  $q-1$  racines  $\lambda_i$  de façon à obtenir la meilleur séparation des dynamiques du système pour une variation de  $f^*$  donnée.

#### I.2.1.1. - Choix de $a$ .

Les paramètres  $\lambda_i$  sont solution de  $D(\lambda) + aN(\lambda) = 0$ . Les termes non constants situés dans la dernière colonne sont de la forme : (cf § II.1 chap III)

$$\begin{aligned} & \bullet - k_i g^* \text{ avec } \begin{cases} g^* = f^* - a \text{ pour } i = 1, \dots, q-1 \\ k_i = \frac{N(\lambda_i)}{R_i(\lambda_i)} \end{cases} \\ & \bullet - \sum_{i=1}^{q-1} \lambda_i - b_{q-1} g^* \text{ pour } i = 1, \dots, q. \end{aligned}$$

L'expression du rayon des cercles est alors :

$$(IV.10) \quad r_i = \max_{f^*} \sqrt{|-k_i (f^* - a)|} \quad \text{pour } i = 1, \dots, q-1$$

et

$$r_q = \max_{f^*} \sqrt{\left| - \sum_{i=1}^{q-1} \lambda_i - b_{q-1} (f^* - a) \right|} \quad \text{pour } i = q.$$

Les rayons minimaux  $r_i$  sont obtenus pour une valeur de  $f^*$  proche de  $a$ ,  $-k_i g^*$  tend en effet dans ce cas vers 0.

La fonction définie par  $\sqrt{|-k_i (f^* - a)|}$  est croissante en  $f^*$  pour  $f^* > a$  et décroissante pour  $f^* < a$ . Si  $f$  varie dans l'intervalle  $[f^* \min, f^* \max]$ , la fonction sera maximale :

- pour  $f$  max si  $f^* > a$

- pour  $f$  min si  $f^* < a$ .

La valeur du rayon maximal est donc donnée par

$$(IV.11) \quad r_i = \max \left( \sqrt{|k_i(f_{\min}^* - a)|}, \sqrt{|-k_i(f_{\max}^* - a)|} \right)$$

Cette expression dépend de  $a$  et il apparaît nécessaire de déterminer la valeur de  $a$  dans l'intervalle solution de façon à obtenir la meilleure séparation des dynamiques, c'est à dire la plus petite valeur de  $\epsilon$  correspondant à des rayons  $r_i$  définis par (IV.11). Selon /DAUPHIN-TANGUY 1983/, il est intéressant de choisir

$$a = \frac{f_{\min}^* + f_{\max}^*}{2}.$$

L'expression du rayon  $r_i$  est dans ce cas la suivante :

$$r_i = \sqrt{|k_i| \cdot \left( \frac{f_{\max}^* - f_{\min}^*}{2} \right)}$$

L'exemple suivant a permis de justifier un tel choix, qui s'avère ici optimal

#### Exemple.

Soit le système caractérisé par sa fonction de transfert

$$\frac{N(p)}{D(p)} = \frac{p^2 + 6p + 5}{p^3 + 11.3p^2 + 13.12p + 1.2}$$

Quelle que soit la valeur de  $a$ , l'équation  $D+aN = 0$  admet toujours  $q-1$  racines réelles négatives (cf. §II.2.3, chap. III).

On se fixe un intervalle de variation pour  $f^*$  :  $[0,2]$  et l'on étudie la valeur de  $\epsilon$  suivant celle de  $a$  choisie dans cet intervalle.

a	0.5			0.75			0.95			1		
$\lambda_i$	- 0.289 et -10.24			-0.373 et -10.40			-0.434 et -10.51			-0.449 et -10.54		
$f^*$	0	1	2	0	1	2	0	1	2	0	1	2
$r_1$	1.486	1.627	2.74	0.465	0.2688	0.6012	0.493	0.1135	0.518	0.497	0	0.497
$r_2$	0.409	0.409	0.701	1.948	1.1251	2.5159	2.215	0.543	2.344	2.28	0	2.28
$r_3$	1.895	2.036	3.441	2.413	1.3939	3.1171	2.708	0.655	2.863	2.77	0	2.77
$\epsilon$	0.62			0.561			0.517			0.501		

Les valeurs des  $r_i$  encadrées correspondent à des rayons maximaux.

D'après (IV.11), on a en effet : par exemple pour  $a = 0.5$

$$r_1 = \max (1.486 ; 2.74)$$

$$r_2 = \max (0.409 ; 0.701)$$

$$r_3 = \max (1.895 ; 3.441)$$

La même définition est utilisée pour  $a = 0.75$  ou  $a = 0.95$  ou  $a = 1$ .

On se place ainsi dans les conditions les plus mauvaises pour la séparation des dynamiques du système suivant la variation de  $f^*$ .

Le coefficient  $\varepsilon$  est donné par :

$$= \max \left( \frac{r_1+r_2}{\lambda_1-\lambda_2}, \frac{r_1+r_3}{\lambda_1-\lambda_2}, \frac{r_2+r_3}{\lambda_1-\lambda_2} \right)$$

Les résultats obtenus dans le tableau montrent qu'il est possible de sélectionner une valeur de  $a$  qui donne à  $\varepsilon$  une valeur minimale et permet par conséquent une meilleure séparation des dynamiques.

Ainsi sur l'intervalle  $[0,2]$ , il est préférable de choisir  $a=1$  qui correspond à  $a = \frac{f_{\min}^* + f_{\max}^*}{2}$  d'où  $\varepsilon = 0,501$

#### I.2.1.2 - Choix des $\lambda_i$ .

Une fois la valeur de  $a$  fixée, on doit résoudre l'équation de  $D+aN = 0$  qui peut avoir  $q$  ou  $q-1$  racines réelles négatives pour un système d'ordre  $q$ . Il apparaît dans certains cas nécessaire de choisir  $q-1$  racines parmi  $q$  pour la modélisation.

Ainsi dans l'exemple, la résolution de l'équation  $D+0.5N = 0$  conduit à 3 racines :

$$\lambda_1 = -0.289, \lambda_2 = -1.24 \text{ et } \lambda_3 = -10.24$$

On se pose alors le problème suivant :

Quelles sont les 2 valeurs de  $\lambda_i$  donnant la meilleure séparation des dynamiques pour une valeur de  $f^*$  fixée?

Pour répondre à cette question, différentes valeurs ont été attribuées à  $a$  et en conséquence aux paramètres  $\lambda_i$  possibles. Le calcul des rayons  $r_i$  et du terme  $\varepsilon$  a donné les résultats suivants.

f	1		2		6	
a	0,5		1		5	
$\lambda_i$	- 0.289 - 10.24	- 1.24 - 10.24	- 0.449 - 10.54	- 1.308 - 10.54	- 0.89 - 13.17	- 2.23 - 13.17
$r_1$	1.627	0.236	0.497	0.35	0.199	0.562
$r_2$	0.409	1.714	2.280	2.395	2.85	3.024
$r_3$	2.036	1.950	2.777	2.745	3.05	3.586
$\varepsilon$	0.368	0.407	0.501	0.556	0.478	0.604

Le cas des couples solutions  $(-0.289, -1.24)$ ,  $(-0.449, -1.308)$  et  $(-0.89, -2.23)$  n'a pas été envisagé, car la différence  $|\lambda_i - \lambda_k|$  est pour ces 3 cas très faible et contribue à augmenter la valeur de  $\varepsilon$ .

Une interprétation des résultats montre que la valeur de  $\varepsilon$  dépend des coefficients  $\lambda_i$ .

Il n'existe pas de critère général permettant de sélectionner les valeurs de  $\lambda_i$  donnant un coefficient  $\varepsilon$  satisfaisant.

La solution actuelle consiste à effectuer la modélisation pour un ensemble de solutions  $\lambda_i$ , de calculer les rayons, le coefficient  $\varepsilon$  et de recommencer l'opération pour d'autres ensembles de solutions  $\lambda_i$ .

On choisit finalement l'ensemble qui donne le meilleur résultat. Une telle démarche est intéressante mais nécessite de nombreux calculs.

La conception des programmes concernant le calcul de  $a$  et des formes en flèche associées se trouve alors tout à fait justifiée et conduit rapidement à un choix de modèle quasi-optimal.

### 1.2.2. - Paramètres arbitraires.

Il peut arriver, pour un système donné, que la valeur de  $a$  égale à  $\frac{f^*_{\min} + f^*_{\max}}{2}$  ne permette pas de résoudre la problématique (P).

Différentes solutions sont alors possibles :

- 1°) - On choisit une valeur de  $a$  différente de  $\frac{f^*_{\min} + f^*_{\max}}{2}$  telle que (P) soit résolu. Les termes non constants sont alors de la forme  $-k_i g^*$  et on procède comme précédemment pour calculer les rayons  $r_i$ .
- 2°) On utilise  $q-1$  paramètres  $\lambda_i$  arbitraires. Les rayons  $r_i$  maximaux sont calculés suivant la formule (IV.8). Les termes non constants sont dans ce cas de la forme  $\delta_i + \gamma_i f^*$  et il est alors plus difficile de déterminer la valeur  $f^*$  donnant le rayon maximal.

La formule (IV.8) s'écrit :

$$(IV.12) \quad r_i = \max_{f^*} \sqrt{|\delta_i + \gamma_i f^*|}$$

Les  $q-1$  paramètres  $\lambda_i$  choisis sont à partie réelle négative. Ils peuvent être réels ou complexes.

Un exemple d'utilisation de paramètres complexes est décrit dans le paragraphe suivant.

#### 1.2.2.1. - Paramètres complexes.

Dans ce cas, les cercles de Gershgorine ne sont pas tous situés sur l'axe réel dans le plan complexe.

La forme en flèche est celle présentée en (III.18) avec des termes non constants réels  $\alpha_i$  pour les  $\lambda_i$  réels et complexes  $\beta_i$  pour les  $\lambda_i$  complexes.

Pour les coefficients  $\lambda_i$  réels, l'expression des rayons maximaux est  $r_i = \max_{f^*} \sqrt{|\alpha_i|}$  ou  $\alpha_i = \frac{P(\lambda_i, \cdot)}{R_i(\lambda_i)}$  et les centres  $\lambda_i$  des cercles sont réels.

Dans le cas complexe, les rayons  $r_j$  sont donnés par  $r_j = \max_f \sqrt{|\beta_j|}$

$$\text{où } \beta_j = \frac{P(\mu_j + i\nu_j)}{R_j(\mu_j + i\nu_j)} \quad \text{et } \beta_j \in \mathbb{C}$$

Les centres des cercles sont alors complexes à partie réelle  $\mu_j$  et à partie imaginaire  $\nu_j$ . Les rayons maximaux sont déterminés en attribuant à  $f^*$  différentes valeurs numériques et non plus les valeurs  $f_{\min}^*$  et  $f_{\max}^*$ .

Rappelons que ces dernières peuvent être utilisées dans le cas d'un paramètre  $a$  égal à  $\frac{f_{\min}^* + f_{\max}^*}{2}$  permettant de résoudre (P).

Pour des paramètres  $\lambda_i$  complexes, il suffirait de déterminer une méthode réalisant la problématique (P') déjà énoncée (§ II.1 chap III) :

(P') :  $D + aN$  admet  $q-1$  solutions complexes à partie réelle négative, avec de plus  $a = \frac{f_{\min}^* + f_{\max}^*}{2}$ . Les termes non constants seraient alors de la forme

$-k_i g^*$  et les rayons maximaux pour  $f_{\min}^*$  ou  $f_{\max}^*$ .

On pourrait alors se demander si les résultats obtenus pour ne sont pas meilleurs dans le cas de paramètres complexes.

Afin d'illustrer cette idée, nous avons repris l'exemple précédent en modélisant le système sous une forme en flèche mince complexe non pas à partir de paramètres solutions de  $D + aN = 0$  mais de paramètres arbitraires.

#### Exemple.

$$\frac{N(p)}{D(p)} = \frac{p^2 + 6p + 5}{p^3 + 11.3p^2 + 13.12p + 1.2} \quad \text{et } f^* \in [0,1]$$

Choisissons comme paramètres de modélisation  $\lambda_1$  complexe et  $\lambda_2$  réel. La forme en flèche correspondante est alors une forme en flèche mince complexe (III.18).

Les paramètres  $\lambda_i$  sont :

$$\lambda_1 = -1 - \frac{i}{4}$$

$$\lambda_2 = -10,24$$

Les rayons  $r_i$  sont les suivants :

$$r_1 = \max_{f^*} \sqrt{|\beta_1|} = \max_{f^*} \left( \left| \frac{-2,138 - 0,062 f^* + i(1,635 - f^*)}{-10,24 + 1 + \frac{i}{4}} \right| \right)$$

$$r_2 = \max_{f^*} \sqrt{|\alpha_1|} = \max_{f^*} (|-2,44 + 5,379 f^*|)$$

$$r_1 = 0,291 \quad \text{pour } f^* = 0$$

$$r_2 = 1,714 \quad \text{pour } f^* = 1$$

$$r_3 = 2,005$$

et

$$\varepsilon = \left| \frac{2,005 + 1,714}{-10,24 + 1 + \frac{i}{4}} \right|$$

$$\varepsilon = 0,402.$$

Comparons ce résultat à celui obtenu dans le cas de paramètres  $\lambda_i$  réels.

$$r_1 = 0,236 \quad \text{pour } f^* = 1$$

$$r_2 = 1,714 \quad \text{pour } f^* = 1$$

$$r_3 = 1,950$$

et

$$\varepsilon = \left| \frac{1,950 + 1,714}{-10,24 + 1,24} \right|$$

$$\varepsilon = 0,407$$

Sur cet exemple, la séparation des dynamiques du système est meilleure dans le cas où le système est représenté par une forme en flèche mince complexe.

Il apparaît ainsi que la recherche d'une meilleure séparation de dynamique peut être traitée en considérant des paramètres  $\lambda_i$  complexes, c'est à dire des centres de cercles de Gershorine complexes.

Pour déterminer les paramètres  $\lambda_i$  complexes à choisir, de nombreux essais successifs de valeurs de  $\lambda_i$  sur la représentation en flèche mince complexe sont nécessaires.

Cette opération, qui demande de longs calculs est réalisée aisément par utilisation du programme de modélisation en flèche mince complexe.

## II - APPLICATION DES CRITERES USUELS DE STABILITE AUX SYSTEMES MODELISES PAR UNE FORME EN FLECHE.

Dans le premier chapitre ont été présentés la méthode directe de Ljapunov et deux critères connus : le critère pratique de Borne-Gentina et celui concernant les fonctions quadratiques.

Dans cette partie, nous appliquons ces différentes méthodes aux systèmes de type MLP modélisés par une forme en flèche.

Sur une forme en flèche mince dont les  $\lambda_i$  sont tous réels négatifs solutions de  $D(\lambda) + a N(\lambda)$ , l'interprétation de la méthode directe de Ljapunov à partir du critère pratique de Borne et Gentina et d'une fonction de type quadratique plus intégrale conduit à la mise en oeuvre d'un autre critère définissant un modèle réduit /ZAMBETTAKIS, RICHARD, ROTELLA 1984/.

Pour une forme en flèche  $F(.)$  quelconque définie en III.1, les conditions de stabilité sont données par le critère de Borne et Gentina ou celui des fonctions quadratiques.

### II.1 - Traitement de la forme en flèche mince.

S'il existe  $a$  tel que  $D(\lambda) + a N(\lambda)$  présente  $q$  ou  $q-1$  racines réelles négatives  $\lambda_i$ , le système non linéaire admet une représentation en flèche définie dans le chapitre précédent au paragraphe II.1 par (III.21).

Le système est représenté en régime libre par :

$$\overset{\circ}{x} = \begin{bmatrix} \lambda_1 & & & -k_1 g^* \\ & 0 & & \vdots \\ & & \ddots & \vdots \\ & & & \lambda_{q-1} & -k_{q-1} g^* \\ 1 & & & & \gamma^* \end{bmatrix} x$$

où

$$x^T = (x_1, \dots, x_{q-1}, \varepsilon)$$

$$\gamma^* = \lambda_q - b_{q-1} g^*$$

$$g^* = f^* - a$$



Il est possible de regrouper dans la dernière colonne de cette forme matricielle les facteurs positifs et négatifs de  $g^*$  en effectuant une permutation adéquate des termes  $\lambda_i$  de la diagonale.

La représentation en régime libre du système (S) est la suivante :

$$\overset{\circ}{x} = F(.)x$$

(IV.13)

avec  $F(.) =$

$$\begin{bmatrix} \lambda_1 & & & -k_1^+ g^* \\ & \lambda_\ell & & \vdots \\ & & \lambda_{\ell+1} & -k_\ell^+ g^* \\ & & & \vdots \\ & & & \lambda_{q-1} & -k_{\ell+1}^- g^* \\ & & & & \vdots \\ 1 & & & & -k_{q-1}^- g^* \\ & & & & \gamma^* \end{bmatrix}$$

où

$$x^T = (x_+^T, x_-^T, \varepsilon)$$

$$x_+^T = (x_1, \dots, x_\ell)$$

$$x_-^T = (x_{\ell+1}, \dots, x_{q-1})$$

$$\forall i \in \{1, \dots, \ell\}, k_i^+ > 0$$

$$\forall i \in \{\ell+1, \dots, q-1\}, k_i^- < 0$$

$$\forall \varepsilon \in E, g^*(\varepsilon) \cdot \xi = g(\varepsilon) \in \mathbb{R}$$

$$\gamma^*(\varepsilon) \cdot \varepsilon = \gamma(\varepsilon) \in \mathbb{R}.$$

$g^*(\varepsilon)$  peut être constant ou non suivant le choix de  $a$ .

Nous considérons que le signe de  $g^*(\varepsilon)$  reste constant dans un voisinage de  $\varepsilon = 0$ . Ce signe est, sans que cela soit restrictif, supposé positif dans un premier temps.

$$\exists h > 0, \forall \varepsilon \in E, |\varepsilon| < h \Rightarrow g^* > 0$$

De façon à schématiser l'expression de la matrice  $F(\varepsilon)$  présentée en (IV.13), nous posons :

$$\begin{cases} \Lambda_+ = \text{diag} \{ \lambda_1, \lambda_2, \dots, \lambda_\ell \}, \Lambda_+ \in \mathbb{R}^{\ell \times \ell} \\ \Lambda_- = \text{diag} \{ \lambda_{\ell+1}, \lambda_{\ell+2}, \dots, \lambda_{q-1} \}, \Lambda_- \in \mathbb{R}^{(q-\ell-1) \times (q-\ell-1)} \end{cases}$$

$$\begin{cases} k_+ = (k_1^+, k_2^+, \dots, k_\ell^+)^T, k_+ \in \mathbb{R}^\ell \\ k_- = (k_{\ell+1}^-, k_{\ell+2}^-, \dots, k_{q-1}^-)^T, k_- \in \mathbb{R}^{q-\ell-1} \end{cases}$$

La matrice  $F(\cdot)$  s'écrit alors :

(IV.14)

$$F(\cdot) = \begin{bmatrix} \Lambda_+ & 0 & -k_+ & g^* \\ 0 & \Lambda_- & -k_- & g^* \\ 1 & -1 & \gamma^* & \end{bmatrix}$$

C'est à partir de cette modélisation que nous étudions la stabilité du système.

Il a été démontré dans /ZAMBETTAKIS, RICHARD, ROTELLA 1984/, /RICHARD 1984/ le théorème suivant :

Théorème.

Soient les deux sous systèmes interconnectés définis à partir de (IV.13) et (IV.14) par :

$$(S_+) : \overset{\circ}{z}_+ = \Lambda_+ z_+$$

$$(S_-) : \overset{\circ}{z}_- = S_-(.)z_-$$

(IV.15)

$$\text{avec } S_-(.) = \begin{bmatrix} & & & | & \\ & & & | & \\ & \Lambda_- & & | & -k_- g^* \\ & & & | & \\ 1 & \text{-----} & 1 & | & \gamma^* \end{bmatrix}$$

matrice en flèche mince définissant le système réduit  $(S_-)$  où  $g^* > 0$

Si  $(S_+)$  est asymptotiquement stable ( $\Lambda_+ < 0$ ) et si  $(S_-)$  vérifie les conditions de stabilité du linéaire :

$$(IV.16) \quad 1^\circ) \Lambda_- < 0$$

$$2^\circ) \exists h > 0, \forall \epsilon \in \mathbb{R}, |\epsilon| < h \Rightarrow (-1)^{q-l} \det S_-(.) > 0$$

alors l'équilibre  $x=0$  du système complet  $(S)$  est asymptotiquement stable.

Il est à noter que le signe de  $g^*$  constant dans un voisinage de  $\epsilon = 0$  peut être choisi négatif.

On définit de la même façon que pour  $g^* > 0$  deux sous systèmes interconnectés à partir de  $(S)$  définis par

$$(S_-) : \overset{\circ}{z}_- = \Lambda_- z_-$$

$$(S_+) : \overset{\circ}{z}_+ = S_+(.)z_+$$



3°) Un autre cas où l'on ne peut utiliser les résultats précédents est celui d'un système à terme  $f^*$  non linéaire dépendant d'autres variables que  $\varepsilon$ , de  $(x,t)$  par exemple.

On applique alors sur sa matrice représentative le critère de Borne et Gentina pour déterminer le domaine de stabilité.

### Cas particuliers.

Nous avons déjà souligné que les termes de la dernière colonne peuvent être constants ou non suivant la valeur de  $a$  (cf § II.2.1 et II.2.2 du chap. III)

Ainsi pour  $a$  nul les termes de la dernière colonne de  $F(\cdot)$  sont tous proportionnels à  $f$ .

Le domaine de stabilité est déterminé en appliquant le théorème aux systèmes  $(S_+)$  et  $(S_-)$  définis en (IV.15) ou (IV.17). Les conditions suivantes doivent être vérifiées :

- pour  $f^* > 0$  1°)  $\Lambda_- < 0$

2°)  $\exists h > 0, \forall \varepsilon \in \mathbb{R}, |\varepsilon| < h \Rightarrow (-1)^{q-l} \det S_-(\cdot) > 0$   
avec  $(S_-)$  défini par la matrice suivante

$$S_-(\cdot) = \begin{bmatrix} & & & | & \\ & & & | & \\ & & \Lambda_- & | & -k_- f^* \\ 1 & \text{-----} & 1 & | & \gamma_-^* \\ & & & | & \end{bmatrix} \quad \text{et } -k_i^- = \gamma_i$$

- pour  $f^* > 0$  1°)  $\Lambda_+ < 0$

2°)  $\exists h > 0, \forall \varepsilon \in \mathbb{R}, |\varepsilon| < h \Rightarrow (-1)^{\ell+1} \det S_+(\cdot) > 0$   
avec  $(S_+)$  défini par la matrice suivante :

$$S_+(\cdot) = \begin{bmatrix} & & & | & \\ & & & | & \\ & & \Lambda_+ & | & -k_+ f^* \\ 1 & \text{-----} & 1 & | & \gamma_+^* \\ & & & | & \end{bmatrix} \quad \text{et } -k_i^+ = \gamma_i$$

Pour  $a$  infini, les termes de la dernière colonne de la matrice sont tous constants.

Les conditions à vérifier sont alors les suivantes :

- 1°)  $\Lambda_- < 0$

2°)  $\exists h > 0, \forall \varepsilon \in \mathbb{R}, |\varepsilon| < h \Rightarrow (-1)^{q-l} \det S_-(.) > 0$   
avec  $(S_-)$  défini par la matrice suivante :

$$S_-(.) = \begin{bmatrix} & & & \delta_{\ell+1} \\ & & & | \\ & & \Lambda_- & \delta_{q-1} \\ 1 \text{ --- } & & 1 & \gamma^* \end{bmatrix}$$

et  $\forall i \in \{\ell+1, \dots, q-1\} \delta_i > 0$

ou

- 1°)  $\Lambda_+ < 0$

2°)  $\exists h > 0, \forall \varepsilon \in \mathbb{R}, |\varepsilon| < g \Rightarrow (-1)^{\ell+1} \det S_+(.) > 0$   
avec  $S_+(.)$  défini par la matrice suivante :

$$S_+(.) = \begin{bmatrix} & & & |\delta_1| \\ & & & | \\ & & \Lambda_+ & |\delta_\ell| \\ 1 \text{ --- } & & 1 & \gamma^* \end{bmatrix}$$

et  $\forall i \in \{1, \dots, \ell\} \delta_i < 0$

## II.2 - Traitement de la forme en flèche épaisse.

Les résultats précédents définissant un modèle réduit ne sont pas directement valables dans le cas de racines multiples ou complexes. La recherche de conditions de stabilité s'effectue alors par l'utilisation du critère de Borne et Gentina ou de formes quadratiques.

L'application de ces critères sur une forme en flèche épaisse, identique à celle définie en (III.10) au chapitre III, nous amène à formuler certaines remarques concernant les blocs  $C_i$  et  $M_i$  de la matrice. On énonce ainsi des conditions nécessaires à l'application des critères.

### II.2.1 - Blocs $C_j$ .

L'application du critère de Borne et Gentina sur une forme en flèche épaisse  $F(\cdot)$  nécessite la recherche de la matrice pseudo-majorante  $M(F(\cdot))$ . Ainsi, si la matrice comporte des blocs  $C_j$ , on détermine des blocs  $M(C_j)$  pseudo-majorants.

$$(IV.19) \quad M(C_j) = \begin{bmatrix} \mu_j & |v_j| & 1 & 0 & 0 \\ |v_j| & \mu_j & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \mu_j & |v_j| \\ 0 & 0 & 0 & |v_j| & \mu_j \end{bmatrix}$$

La condition portant sur les mineurs successifs de la matrice pseudo-majorante se traduit dans un premier temps par une condition sur les blocs  $M(C_j)$  : le critère de Borne et Gentina est utilisable à la condition (nécessaire) que les mineurs successifs de  $-M(C_j)$  soient positifs, ce qui est équivalent à :

$$(IV.20) \quad \mu_j^2 - v_j^2 > 0, \mu_j > 0, \forall j \in \{1, \dots, t\}$$

Le critère de Borne et Gentina peut donc être appliqué à une forme en flèche épaisse, dont le polynôme représentatif admet des racines complexes  $\mu_j + i v_j$ ,  $\forall j \in \{1, \dots, t\}$ , si ces racines sont contenues dans le plan défini (fig IV.1) avec les droites limites exclues.

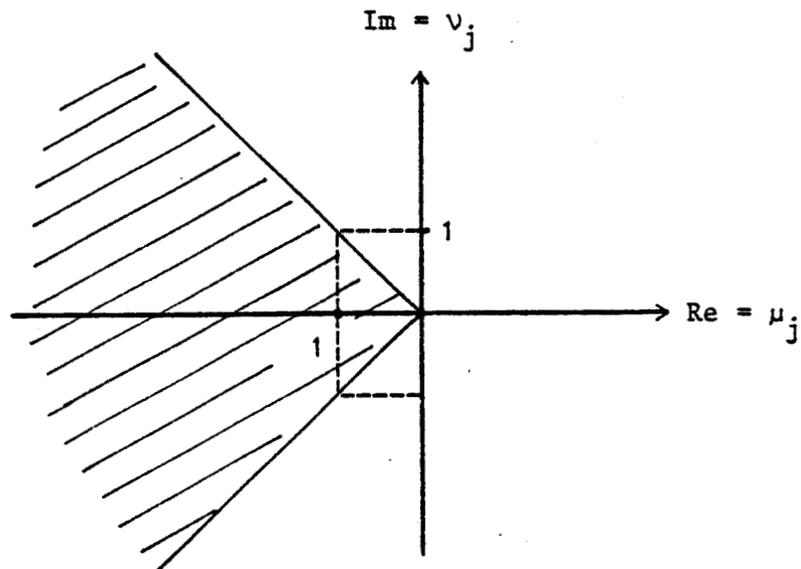


fig. IV.1



D'autre part, l'application de la méthode directe de Lyapunov à partir de fonctions de type quadratique plus intégrale fait intervenir le signe de la forme quadratique  $\frac{F(.) + F^T(.)}{2}$  où  $F(.)$  est une matrice en flèche épaisse.

Ce calcul se traduit sur les blocs  $C_j$  par une étude du signe de la forme suivante :

$$(IV.21) \quad C_j + C_j^T = \begin{bmatrix} 2\mu_j & 0 & 1 & 0 & 0 & 0 \\ 0 & 2\mu_j & & & & \\ 1 & & & & & \\ & & & & & \\ | & & & & & \\ 0 & 0 & 1 & 0 & 2\mu_j & \end{bmatrix}$$

On constate que dans ce cas, il n'existe pas de conditions sur la partie imaginaire  $\nu_j$  de la racine complexe mais uniquement sur la partie réelle  $\mu_i$ , contrairement à l'application du critère de Borne et Gentina.

### II.2.2 - Blocs $M_i$ .

L'étude précédente menée sur les blocs  $C_j$  peut être faite sur les blocs  $M_i$ .

L'application du critère de Borne et Gentina n'entraîne pas de conditions particulières sur les blocs  $M_i$ . En effet ces blocs sont leurs propres pseudo-majorants.

$$M_i = M(M_i) = \begin{pmatrix} \lambda_i & 1 & 0 & 0 \\ 0 & & & \\ 0 & & & \\ 0 & & 0 & \lambda_i \end{pmatrix}$$

La seule condition est donc  $\lambda_i < 0$ , condition réalisée a priori.

En appliquant la méthode directe de Ljapunov, l'étude du signe de la forme quadratique suivante est nécessaire :

$$(IV.22) \quad M_i + M_i^T = \begin{pmatrix} 2\lambda_i & 1 & 0 \\ 1 & & \\ 0 & & 1 \\ & & 1 & 2\lambda_i \end{pmatrix}$$

### II.2.3 - Remarque.

La recherche du signe des formes quadratiques (IV.21) et (IV.22) se fait par calcul des mineurs successifs de chacune des formes. Ce calcul peut être menée en utilisant les équations récurrentes suivantes :

- pour la forme quadratique (IV.21) :

On pose  $n = 2 \times m_j$ ,  $m_j$  étant la multiplicité de la racine complexe. Il vient la relation de récurrence suivante :



## II.3 - Traitement informatique.

### II.3.1 - Tableau récapitulatif.

Dans les paragraphes précédents ont été présentées les méthodes permettant de déterminer les conditions de stabilité dans le cas d'une flèche mince ou épaisse.

Il n'existe pas de méthode générale permettant d'assurer qu'un critère donné conduira aux meilleures conditions de stabilité pour un système donné.

On procède par essais successifs des critères existants sur une matrice représentative du système.

Afin de faciliter ce travail un programme interactif d'aide à l'analyse a été réalisé.

Certains modules ne sont pas actuellement programmés, seuls les principaux ont été réalisés.

Le tableau figure IV.2 décrit :

1°) en trait plein : ce qui est réalisé c'est à dire le traitement d'une forme en flèche

- mince à paramètres  $\lambda_i$  solutions de  $D(\lambda) + a N(\lambda) = 0$ 
  - par définition d'un modèle réduit
  - par application du critère de Borne et Gentina
- épaisse à paramètres  $\lambda_i$  solution de  $D(\lambda) = 0$  ou  $N(\lambda) = 0$ 
  - par application du critère de Borne et Gentina.

2°) en traits pointillés : les extensions futures possibles :

- traitement d'une forme en flèche mince ou épaisse à paramètres  $\lambda_i$  quelconques
  - par application du critère de Borne et Gentina
  - par définition d'une forme quadratique  $\frac{F(.) + F^T(.)}{2}$

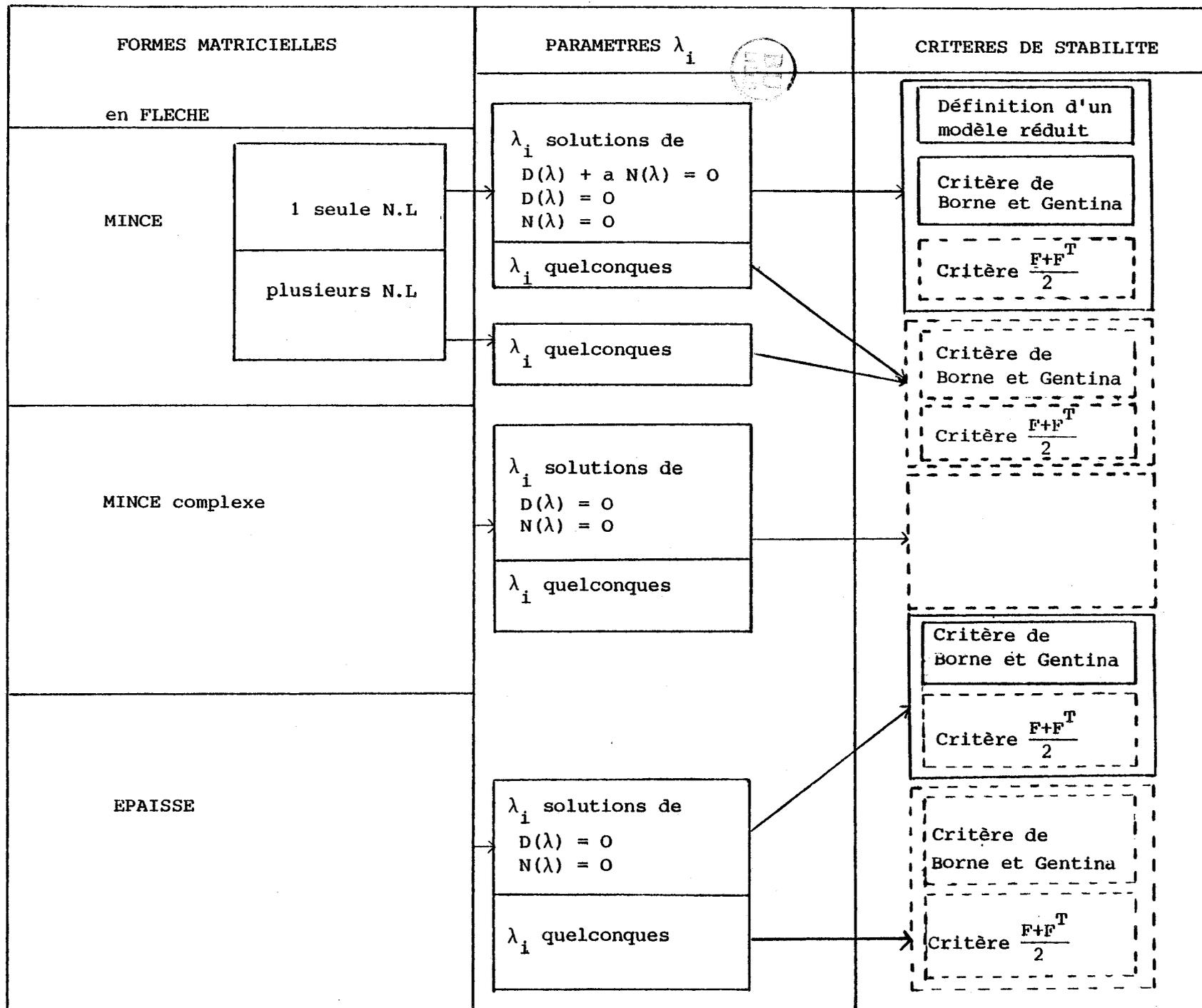


Fig. IV.2.

- traitement d'une forme en flèche mince à plusieurs non-linéarités
- traitement d'une forme en flèche mince complexe.

Il est à noter que la réalisation actuelle permet de conclure dans de nombreux cas, si l'on ne considère pas l'éventualité d'une forme mince complexe ou à plusieurs non-linéarités. Dans ce dernier cas, seule la modélisation est traitée.

En effet, si un critère ne donne pas de résultats sur une forme en flèche donnée, il est possible de modifier les paramètres  $\lambda_i$  de modélisation de cette dernière et d'appliquer de nouveau le même critère sur la nouvelle forme.

### II.3.2 - Réalisation informatique.

La réalisation du théorème énoncé au § II.1, définissant un modèle réduit comprend les étapes suivantes :

1°) définition d'un modèle réduit.

On se fixe un signe pour le terme  $g^*$  et on sélectionne les termes de la dernière colonne positifs (si  $g^* > 0$ ) ou négatifs (si  $g^* < 0$ ).

2°) application du critère de Borne-Gentina au modèle réduit. On calcule le signe de l'expression  $(-1)^{r+1} \cdot \det(F_r)$  où  $r$  est la dimension du modèle réduit,  $F_r$  la matrice.

Lorsqu'il n'est pas possible de définir un modèle réduit, on utilise directement le critère de Borne et Gentina. Cette application se fait en deux parties :

1°) La première définit une matrice pseudo-majorante  $M(F(.))$ .

2°) La seconde calcule le signe des mineurs successifs de  $-M(F(.))$

Dans le cas d'une forme en flèche mince, les  $q-1$  premiers mineurs sont positifs par construction de la matrice.

Dans le cas d'une forme en flèche épaisse, la condition (IV.20) doit être vérifiée.

Le dernier calcul est celui du signe du mineur de plus grande dimension de  $-M(F(.))$  c'est à dire de l'expression suivante :  $(-1)^q \det(M(F(.)))$

La mise en oeuvre de ces deux critères nécessite le calcul d'un déterminant d'une forme en flèche mince ou épaisse à termes non constants. Ce calcul est détaillé dans les parties suivantes :

### II.3.2.1 - Calcul informatisé du déterminant d'une forme en flèche mince.

Soit une matrice en flèche mince définie au paragraphe I.2.1.2 du chapitre III, pour calculer son déterminant on réalise la décomposition suivante :

$$(IV.25) \quad \det \begin{pmatrix} \lambda_1 & & \delta_1 + \gamma_1 f^* \\ & \lambda_{q-1} & \delta_{q-1} + \gamma_{q-1} f^* \\ 1 & \text{---} & 1 & -a_{q-1} - b_{q-1} f^* - \sum_{i=1}^{q-1} \lambda_i \end{pmatrix} = D_1 + f^* D_2$$

avec

$$D_1 = \det \begin{pmatrix} \lambda_1 & & \delta_1 \\ & \lambda_{q-1} & \delta_{q-1} \\ 1 & \text{---} & 1 & -a_{q-1} - \sum_{i=k}^{q-1} \lambda_i \end{pmatrix} \quad D_2 = \det \begin{pmatrix} \lambda_1 & & \gamma_1 \\ & \lambda_{q-1} & \gamma_{q-1} \\ 1 & \text{---} & 1 & -b_{q-1} \end{pmatrix}$$

En utilisant les résultats présentés dans /BENREJEB 1980/ sur les déterminants des formes en flèche mince, on obtient :

$$\bullet D_1 = \left[ (-a_{q-1} - \sum_{i=1}^{q-1} \lambda_i) - \left( \sum_{i=1}^{q-1} \frac{\delta_i}{\lambda_i} \right) \right] \prod_{i=1}^{q-1} \lambda_i$$

(IV.26)

$$\bullet D_2 = (-b_{q-1} - \sum_{i=1}^{q-1} \frac{\gamma_i}{\lambda_i}) \prod_{i=1}^{q-1} \lambda_i$$

Ces formules ont donc été programmées.

### II.3.2.2 - Calcul informatisé du déterminant d'une forme en flèche épaisse.

Comme nous l'avons rappelé précédemment l'application du critère de Borne et Gentina sur une forme en flèche épaisse nécessite le calcul de l'expression  $(-1)^q \det M(F(.))$ .

Nous donnons donc dans cette partie la formule de calcul du déterminant de  $M(F(.))$  avec  $F(.)$  matrice en flèche épaisse à paramètres complexes ou multiples solutions, des équations,  $D = 0$  ou  $N = 0$ . En effet, seul ce cas est envisagé actuellement (cf. tableau figure (IV.2)).

Le détail des calculs est donné dans une annexe située à la fin de ce chapitre.

On y énonce également la formule permettant le calcul du déterminant d'une forme en flèche épaisse quelconque, en vue d'une extension future de l'application du critère de Borne et Gentina à une forme en flèche quelconque.

Soit une matrice en flèche épaisse  $F(.)$  définie en (III.10) au chapitre III, le calcul du déterminant de la matrice pseudo-majorante  $M(F(.))$  est analogue au calcul du déterminant d'une flèche mince. Les termes qui interviennent ne sont pas dans ce cas des scalaires mais des vecteurs.

La forme générale de  $M(F(.))$  est la suivante :

$$M(F(.)) = \begin{pmatrix} |M_1| & & & & |\alpha_1(\cdot)| \\ & |M_s| & & & |\alpha_s(\cdot)| \\ & & |c_1| & & |\beta_1(\cdot)| \\ & & & |c_t| & |\beta_t(\cdot)| \\ 1 & 1 & 1 & 1 & \gamma(\cdot) \end{pmatrix}$$

$$\text{où } \left\{ \begin{array}{l} |M_i| \text{ bloc pseudo-majorant de } M_i \text{ } (|M_i| = M_i \text{ d'après } \S \text{ II.2.2}) \\ |C_j| \text{ bloc pseudo-majorant de } C_j \\ |\alpha_i(\cdot)| \in \mathbb{R}^{n_i} \quad |\alpha_i^T(\cdot)| = (|\alpha_i^{n_i}(\cdot)|, \dots, \alpha_i^1(\cdot)) \\ |\beta_j(\cdot)| \in \mathbb{R}^{2m_j}, \quad |\beta_j^T(\cdot)| = (|\beta_{j2}^{m_j}(\cdot)|, |\beta_{j1}^{m_j}(\cdot)|, \dots, |\beta_{j2}^1(\cdot)|, |\beta_{j1}^1(\cdot)|) \\ \gamma(\cdot) = -a_{q-1} - b_{q-1} f^* - \sum_{i=1}^s n_i \lambda_i - 2 \sum_{j=1}^t \mu_j \cdot m_j \end{array} \right.$$

On décompose les vecteurs  $\alpha_i(\cdot)$  et  $\beta_j(\cdot)$  comme suit

$$\left\{ \begin{array}{l} \alpha_i(\cdot) = \delta_i + \delta_i f^* \\ \beta_j(\cdot) = \delta'_j + \gamma'_j f^* \end{array} \right.$$

avec

$$\delta_i^T = (\delta_i^{n_i}, \dots, \delta_i^1)$$

$$\gamma_i^T = (\gamma_i^{n_i}, \dots, \gamma_i^1)$$

et

$$\delta'_j{}^T = (\delta'_{j2}{}^{n_j}, \delta'_{j1}{}^{n_j}, \dots, \delta'_{j2}{}^1, \delta'_{j1}{}^1)$$

$$\gamma'_j{}^T = (\gamma'_{j2}{}^{m_j}, \gamma'_{j1}{}^{m_j}, \dots, \gamma'_{j2}{}^1, \gamma'_{j1}{}^1)$$

Le cas d'une racine multiple n'a pas été envisagé pour la programmation. La multiplicité  $m_j$  est donc égale à 1.

Les vecteurs  $\delta'_j{}^T$  et  $\gamma'_j{}^T$  sont réduits à :

$$\delta'_j{}^T = (\delta'_{j2}{}^1, \delta'_{j1}{}^1)$$

$$\gamma'_j{}^T = (\gamma'_{j2}{}^1, \gamma'_{j1}{}^1)$$

D'autre part les paramètres de modélisation sont solutions de l'équation  $D = 0$  ou  $N = 0$  et les vecteurs  $\alpha_i(\cdot)$  et  $\beta_j(\cdot)$  ont leurs termes respectivement tous proportionnels à  $f^*$  ou tous constants.

Il vient ainsi :

$$\text{- soit } |\alpha_i(\cdot)| = |\gamma_i f^*|$$

$$|\beta_j(\cdot)| = |\gamma_j' f^*|$$

$$\text{- soit } |\alpha_i(\cdot)| = |\delta_i|$$

$$|\beta_j(\cdot)| = |\delta_j'|$$

On démontre en annexe que le déterminant de  $M(F(\cdot))$  s'écrit

$$\det M(F(\cdot)) = D_1 + f^* D_2$$



avec :

$$(IV.27) \quad \bullet D_1 = \left\{ \left( -a_{q-1} - \sum_{i=1}^s n_i \lambda_i - 2 \sum_{j=1}^t \mu_j m_j \right) - \left( \sum_{i=1}^s \sum_{k=1}^{n_i} \sum_{h=1}^k |\delta_i|^{n_i-k+1} \frac{(-1)^{h+1}}{\lambda_i^h} \right) - \left( \sum_{j=1}^t \frac{|\delta_j'| + |\delta_j|}{\mu_j + |\nu_j|} \right) \right\} \prod_{j=1}^t (\mu_j^2 - |\nu_j|^2) \prod_{i=1}^s (\lambda_i)^{n_i}$$

$$(IV.28) \quad \bullet D_2 = \left\{ -b_{q-1} - \left( \sum_{i=1}^s \sum_{k=1}^{n_i} \sum_{h=1}^k |\gamma_i|^{n_i-k+1} \frac{(-1)^{h+1}}{\lambda_i^h} \right) - \left( \sum_{j=1}^t \frac{|\gamma_j' 2| + |\gamma_j' 1|}{\mu_j + |\nu_j|} \right) \right\} \prod_{j=1}^t (\mu_j^2 - |\nu_j|^2) \prod_{i=1}^s (\lambda_i)^{n_i}$$

### Remarque.

On suppose d'après (IV.20) que  $\mu_j \neq -|\nu_j|$ . Ce sont ces formulations qui sont utilisées pour programmer le calcul d'un déterminant d'une matrice en flèche à termes non constants.

## II.4 - Exemple.

Soit le système défini par sa fonction de transfert  $\frac{N(p)}{D(p)}$

$$\frac{N(p)}{D(p)} = \frac{p^4 + 4p^3 + 6p^2 + 4p + 1}{p^5 + 14p^4 + 71p^3 + 154p^2 + 120p} = \frac{(p+1)^4}{p(p+2)(p+3)(p+4)(p+5)}$$

Dans une première partie, on choisit les paramètres de modélisation  $\lambda_i$  racines de  $D(p)$ .

La représentation matricielle est dans ce cas la forme en flèche mince suivante :

$$F(.) = \begin{pmatrix} -2 & 0 & 0 & 0 & -0.167 f^* \\ 0 & -3 & 0 & 0 & 8 f^* \\ 0 & 0 & -4 & 0 & -40 f^* \\ 0 & 0 & 0 & -5 & 42.6 f^* \\ 1 & 1 & 1 & 1 & -f^* \end{pmatrix}$$

Les termes de la dernière colonne sont tous proportionnels à  $f^*$ .

Si  $f^*$  ne dépend que d'une variable d'état  $\varepsilon$ , on peut appliquer le théorème du système réduit sous son expression (IV.15) ( $f^* > 0$ ).

Il vient, en supposant donc  $f^* > 0$ , le modèle réduit suivant :

$$F_r(0) = \begin{pmatrix} -3 & 0 & 8 f^* \\ 0 & -5 & 42.6 f^* \\ 1 & 1 & -f^* \end{pmatrix}$$

Le calcul de  $(-1)^3 \det F_r(.)$  se fait suivant la formule (IV.26)

$$(-1)^3 \det F_r(.) = - \det F_r(.) = -(D_1 + f^* D_2)$$

$$\text{avec } \begin{cases} D_1 = 0 \\ D_2 = (-1 - (-\frac{8}{3} - \frac{42.6}{5})) \times 15 = 152.8 f^* \end{cases}$$

$$\text{d'où } -\det F_r(.) = -152.8 f^*.$$

Les conditions du théorème (IV.15) donnent :

$$1^\circ) f^* > 0$$

$$2^\circ) -152.8 f^* > 0$$

Ces deux conditions sont contradictoires et le théorème définissant un modèle réduit ne permet pas de conclure.

On se propose dans un second temps d'essayer le critère de Borne et Gentina sur une forme en flèche épaisse.

On choisit dans ce cas une racine multiple d'ordre 4, solution de  $N(p) = 0$ . La forme en flèche associée est :

$$F(.) = \begin{pmatrix} -1 & 1 & 0 & 0 & -10 \\ 0 & -1 & 1 & 0 & -41 \\ 0 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & -1 & 24 \\ 1 & 1 & 1 & 1 & -10-f^* \end{pmatrix}$$



On en tire la matrice pseudo-majorante  $M(F(.))$

$$M(F(.)) = \begin{pmatrix} -1 & 1 & 0 & 0 & 10 \\ 0 & -1 & 1 & 0 & 41 \\ 0 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & -1 & 24 \\ 1 & 1 & 1 & 1 & -10-f^* \end{pmatrix}$$

Cherchons le signe de  $(-1)^5 \det M(F(.))$ .

Le calcul informatisé du déterminant  $\det M(F(.))$  s'effectue suivant les formules (IV.27) et (IV.28).

On obtient :

$$D_1 = \left( -10 - \sum_{k=1}^4 \sum_{h=1}^k \frac{|\delta_i^{4-k+1}| (-1)^{h+1}}{(-1)^h} \right) (-1)^4$$

ou encore  $D_1 = 184$  et  $D_2 = -1$

d'où  $(-1)^5 \det M(F(.)) = -184 + f^*$

L'application de critère de Borne-Gentina conduit au résultat suivant :

$$(-1)^5 \det M(F(.)) > 0$$

c'est à dire  $f^* > 184$ .

## CONCLUSION.

Les paramètres de modélisation  $\lambda_i$  d'un système définissent la forme de la matrice en flèche et l'expression de ses termes non constants.

Nous avons montré dans ce chapitre l'intérêt de concevoir un même programme pour répondre à deux problématiques différentes :

- 1°) choisir ces paramètres  $\lambda_i$  de façon à obtenir la meilleure séparation des dynamiques du système pour une certaine variation de  $f^*$ .
- 2°) rechercher le plus grand domaine de stabilité par essais successifs de critères, tels que le critère de Borne-Gentina, le critère des formes quadratiques et le critère du modèle réduit, sur une forme en flèche donnée ou sur différentes formes en flèches représentatives d'un même système.

Outre le programme principal de modélisation en flèche, nous avons donc proposé un programme d'aide à l'analyse qui met en oeuvre le critère de Borne et Gentina ou le critère du modèle réduit sur des formes en flèche dont les paramètres  $\lambda_i$  sont solution de l'équation  $D(\lambda) + a N(\lambda) = 0$ .

Le critère du modèle réduit n'est utilisé que dans le cas d'une forme en flèche mince.

Des extensions futures du programme existant actuellement sont envisageables. Elles permettraient l'application d'autres critères sur toutes les formes en flèche possibles.

ANNEXE

CALCUL DU DETERMINANT D'UNE FORME EN FLECHE EPAISSE QUELCONQUE.

Soit une matrice en flèche épaisse définie en (III.10) au chapitre III, on montre que son déterminant se décompose comme suit :

$$\det F(.) = D_1 + f * D_2$$

$$\text{avec } D_1 = \left( \begin{array}{cccc|cccc} M_1 & & & & \delta_1 & & & \\ & M_s & & & | & & & \\ & & C_1 & & \delta_s & & & \\ & & & C_t & \delta'_1 & & & \\ & & & & | & & & \\ & & & & \delta'_t & & & \\ \hline 1 & \text{---} & 1 & \text{---} & 1 & \text{---} & 1 & -a_{q-1} - \sum_{i=1}^s n_i \lambda_i - 2 \sum_{i=1}^t \mu_i m_i \end{array} \right)$$

$$D_2 = \left( \begin{array}{cccc|c} M_1 & & & & \gamma_1 \\ & M_s & & & | \\ & & C_1 & & \gamma_s \\ & & & C_t & \gamma'_1 \\ & & & & | \\ & & & & \gamma'_t \\ \hline 1 & \text{---} & 1 & \text{---} & 1 & -b_{q-1} \end{array} \right)$$

D'après le théorème relatif au déterminant d'une flèche épaisse présenté dans /BENREJEB 1980/ et rappelé au paragraphe I.1 du chapitre III, on détermine  $D_1$  et  $D_2$

(A<sub>1</sub>.1)

$$\bullet D_1 = \left\{ \left[ -a_{q-1} - \sum_{i=1}^s n_i \lambda_i - 2 \sum_{j=1}^t \mu_j m_j \right] - \left[ \sum_{i=1}^s (1 \dots 1) M_i^{-1} \delta_i \right] - \left[ \sum_{j=1}^t (1 \dots 1) C_j^{-1} \delta'_j \right] \right\} \prod_{j=1}^t (\mu_j^2 + \gamma_j^2) \prod_{i=1}^s (\lambda_i)^{n_i}$$

(A<sub>1</sub>.2)

$$\bullet D_2 = \left\{ -b_{q-1} - \left[ \sum_{i=1}^s (1 \dots 1) M_i^{-1} \gamma_i \right] - \sum_{j=1}^t (1 \dots 1) C_j^{-1} \gamma'_j \right\} \prod_{j=1}^t (\mu_j^2 + \gamma_j^2) \prod_{i=1}^s (\lambda_i)^{n_i}$$

Ces formules correspondent au calcul du déterminant d'une forme en flèche épaisse quelconque.

Elles ne sont pas exactement celles utilisées dans la programmation du critère de Borne et Gentina.

En effet, dans ce cas il ne s'agit plus de calculer  $\det F(\cdot)$  mais  $\det M(F(\cdot))$ ,  $M(F(\cdot))$  matrice pseudo-majorante, en tenant compte de l'hypothèse supplémentaire suivante : les paramètres sont solutions de  $D = 0$  et  $N = 0$  (cf § II.3.2.2 du chapitre IV).

Il vient alors :

(A<sub>1</sub>.3)

$$\bullet D_1 = \left\{ \left[ -a_{q-1} - \sum_{i=1}^s n_i \lambda_i - 2 \sum_{j=1}^t \mu_j m_j \right] - \left[ \sum_{i=1}^s (1 \dots 1) M_i^{-1} |\delta_i| \right] - \left[ \sum_{j=1}^t (1 \dots 1) (M(C_j))^{-1} |\delta'_j| \right] \right\} \prod_{j=1}^t (\mu_j^2 - |\nu_j|^2) \prod_{i=1}^s (\lambda_i)^{n_i}$$

(A<sub>1</sub>.4)

$$\bullet D_2 = \left\{ -b_{q-1} - \left[ \sum_{i=1}^s (1 \dots 1) M_i^{-1} |\gamma_i| \right] - \sum_{j=1}^t (1 \dots 1) (M(C_j))^{-1} |\gamma'_j| \right\} \prod_{j=1}^t (\mu_j^2 - |\gamma_j|^2) \prod_{i=1}^s (\lambda_i)^{n_i}$$

Remarque.

Les expressions  $\sum_{i=1}^s (1 \dots 1) M_i^{-1} |\delta_i|$ ,  $\sum_{j=1}^t (1 \dots 1) (M(C_j))^{-1} |\delta_j|$  et

$\sum_{i=1}^s (1\dots 1) M_i^{-1} |\gamma_i|$ ,  $\sum_{j=1}^t (1\dots 1) (M(C_j))^{-1} |\gamma_j|$  peuvent être simplifiées.

On montre en effet que la matrice  $M_i^{-1}$  existe et est définie par :

$$(A_1.5) \quad M_i^{-1} = \begin{bmatrix} \frac{1}{\lambda_i} & \frac{-1}{\lambda_i^2} & \frac{1}{\lambda_i^3} & \dots & \frac{(-1)^{n_i+1}}{\lambda_i^{n_i}} \\ 0 & \frac{1}{\lambda_i} & \frac{-1}{\lambda_i^2} & \dots & \frac{1}{\lambda_i^3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \frac{1}{\lambda_i} \end{bmatrix}$$

$M_i^{-1} \in \mathbb{R}^{n_i \times n_i}$ ,  $n_i$  est la multiplicité de la racine  $\lambda_i$ .

Le produit  $(1\dots 1) M_i^{-1} |\delta_i|$  est égal à un scalaire puisque  $(1\dots 1)^T \in \mathbb{R}^{n_i}$ ,  $M_i^{-1} \in \mathbb{R}^{n_i \times n_i}$  et  $\delta_i \in \mathbb{R}^{n_i}$ .

Il peut s'écrire en utilisant l'expression (A<sub>1</sub>.5) de  $M_i^{-1}$  comme suit :

$$(A_1.6) \quad (1\dots 1) M_i^{-1} |\delta_i| = \sum_{k=1}^{n_i} \sum_{h=1}^k |\delta_i^{n_i-k+1}| \frac{(-1)^{h+1}}{\lambda_i^h}$$

L'expression du produit  $\sum_{j=1}^t (1\dots 1) (M(C_j))^{-1} |\delta_j|$  peut également être simplifiée.

Le cas d'une racine complexe a été étudié uniquement pour une racine complexe simple. La matrice  $M(C_j)$  est donc une matrice carrée d'ordre 2. Son inverse existe et est facilement calculé.

$$(M(C_j))^{-1} = \frac{1}{\mu_j^2 - \nu_j^2} \begin{pmatrix} \mu_j & -|\nu_j| \\ -|\nu_j| & \mu_j \end{pmatrix}$$

Le produit matriciel (1 1) (M(C\_j))^{-1} |\delta\_j| est réduit à un scalaire

$$(1 \ 1) (M(C_j))^{-1} |\delta_j| = \frac{|\delta_j^1| + |\delta_j^1|}{j_2} \frac{j_1}{\mu_j + |\nu_j|}$$

Un raisonnement analogue permet de simplifier les expressions suivantes

$$\sum_{i=1}^s (1 \dots 1) M_i^{-1} |\gamma_i| \text{ et } \sum_{j=1}^t (1 \dots 1) (M(C_j))^{-1} |\gamma_j|$$

Il vient alors

$$(A_1.7) \quad (1 \dots 1) M_i^{-1} |\gamma_i| = \sum_{k=1}^{n_i} \sum_{h=1}^k |\gamma_i^h| \frac{(-1)^{h+1}}{\lambda_i^h}$$

et

$$(A_1.8) \quad (1 \ 1) (M(C_j))^{-1} |\gamma_j| = \frac{|\gamma_j^1| + |\gamma_j^1|}{j_2} \frac{j_1}{\mu_j + |\nu_j|}$$

En remplaçant chaque expression par sa valeur, on obtient de nouvelles formulations de D<sub>1</sub> et D<sub>2</sub> présentées au §II.3.2.2 du chapitre IV.

$$\begin{aligned} \bullet \text{ (det 1)'} &= \left\{ \left\{ -a_{q-1} - \sum_{i=1}^s n_i \lambda_i - 2 \sum_{j=1}^t \mu_j \right\} - \left( \sum_{i=1}^s \sum_{k=1}^{n_i} \sum_{h=1}^k \delta_i \frac{(-1)^{h+1}}{\lambda_i^h} \right) \right. \\ &\quad \left. - \sum_{j=1}^t \frac{\delta_j^1 + \delta_j^1}{j_2} \frac{j_1}{\mu_j + |\nu_j|} \right\} \prod_{j=1}^t (\mu_j^2 - |\nu_j|^2) \prod_{i=1}^s (\lambda_i)^{n_i} \end{aligned}$$

$$(A_1.9) \quad \bullet \text{ (det 2)'} = \left\{ -b_{q-1} - \left( \sum_{i=1}^s \sum_{k=1}^{n_i} \sum_{h=1}^k \gamma_i \frac{(-1)^{h+1}}{\lambda_i^h} \right) - \left( \sum_{j=1}^t \frac{\gamma_j^1 + \gamma_j^1}{\mu_j + |\nu_j|} \right) \right\} \prod_{j=1}^t (\mu_j^2 - |\nu_j|^2) \prod_{i=1}^s (\lambda_i)^{n_i}$$

## CHAPITRE V.

DESCRIPTION DU PROGRAMME D'AIDE A LA MODELISATION

ET

A L'ANALYSE DES SYSTEMES NON-LINEAIRES DE TYPE LUR'E POSTNIKOV.

## INTRODUCTION.

Dans les chapitres précédents, nous avons justifié l'utilisation d'un logiciel d'aide à la modélisation et à l'analyse des systèmes non linéaires de type mLP.

Ce programme doit servir à plusieurs utilisateurs ayant des niveaux de connaissance différents.

Il a été écrit de façon à pouvoir être utilisé par des chercheurs, afin de réaliser plus rapidement certaines applications, ou par des étudiants dans le cadre d'enseignement assisté par ordinateur.

Il est donc nécessaire d'envisager des procédures de traitement simples et rapides, souples et modifiables à volonté.

Enfin, il faut veiller à ce que l'aspect informatique de la réalisation soit le plus transparent possible c'est à dire qu'aucune connaissance informatique ne doit être requise pour la mise en oeuvre du système.

Dans ce chapitre, nous expliquons comment il a été possible de répondre à ces besoins en créant un programme modulaire à aspect conversationnel.

Nous détaillons la structure du programme et de ses modules, les possibilités du programme.

Nous décrivons également les sous-programmes appelés.

Enfin, des exemples d'application sont présentés afin de mettre en évidence les possibilités du programme.

## I - PRÉSENTATION DU PROGRAMME.

### I.1 - La structure du programme.

Le langage de programmation utilisé est le FORTRAN 77 /LIGNELET 1982/. Un tel choix se justifie par l'orientation du FORTRAN vers les problèmes de nature numérique.

Les algorithmes mis au point sont stables et robustes afin de ne pas gêner l'utilisateur par des problèmes d'analyse numérique.

Le programme est implémenté sur un calculateur VAX 11/750. Il contient environ 4000 lignes et 300 K mémoire.

L'écriture du programme a été réalisée de façon modulaire pour faciliter les extensions et les modifications sans remettre en cause le logiciel existant.

Par exemple, l'adjonction d'un ensemble réalisant la synthèse de systèmes non linéaires de type mLP pourra être effectuée en ajoutant au programme actuel différents sous-programmes relatifs à la synthèse.

Le logiciel LIMA (Logiciel de Modélisation et d'Analyse des systèmes non linéaires dans l'espace d'état) comprend les sous-ensembles suivants :

- les modules conversationnels pour la définition, le choix des paramètres, la modélisation et l'analyse du système,
- les modules d'édition des résultats,
- un module d'aide expliquant les différents choix de modules possibles.

La structure et les possibilités du programme sont décrites par le schéma figure V.1 qui présente les différents modules.

Dans le programme, les possibilités sont communiquées à l'utilisateur par l'affichage du menu.

Le contenu de chacun des modules est précisé par les figures V2, V.3, V.4, V.5.

La description détaillée du programme principal et de ses sous-programmes sera faite dans le paragraphe II de ce chapitre.

#### STRUCTURE DU PROGRAMME PRINCIPAL LIMA

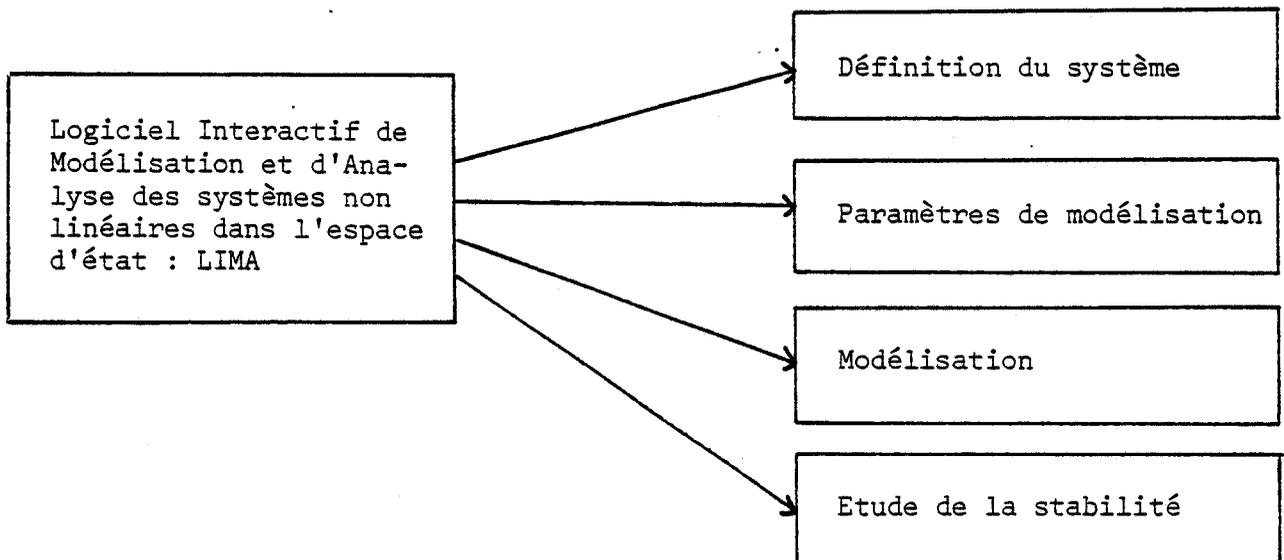


Fig. V.1.

STRUCTURE DU MODULE DEFINITION DU SYSTEME.

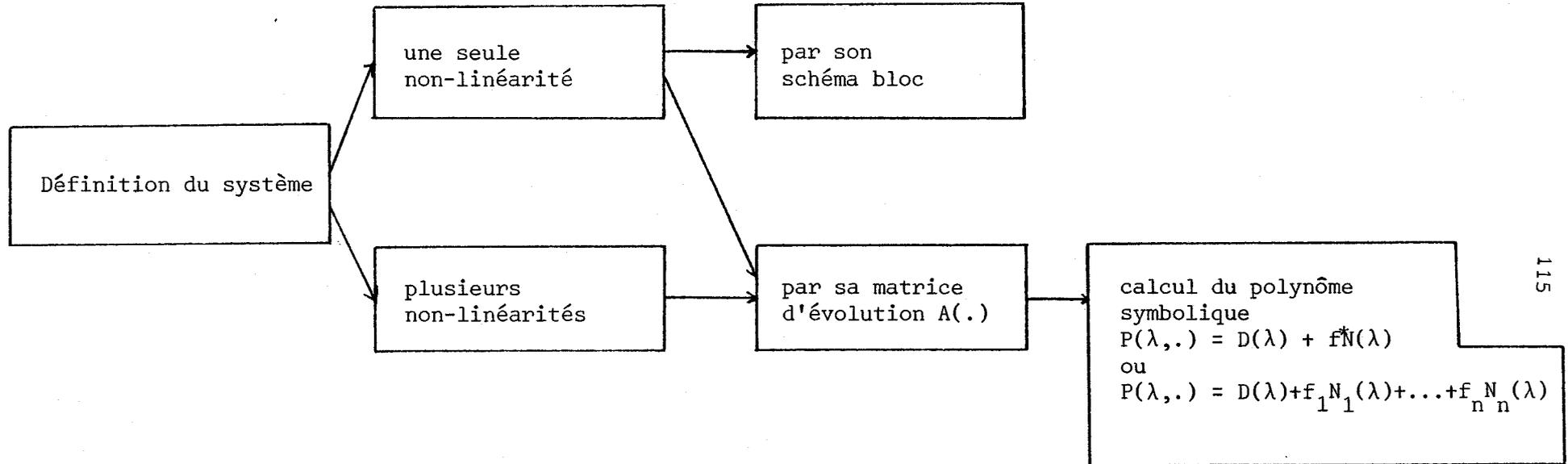


Fig. V.2.

STRUCTURE DU MODULE PARAMETRES DU SYSTEME.

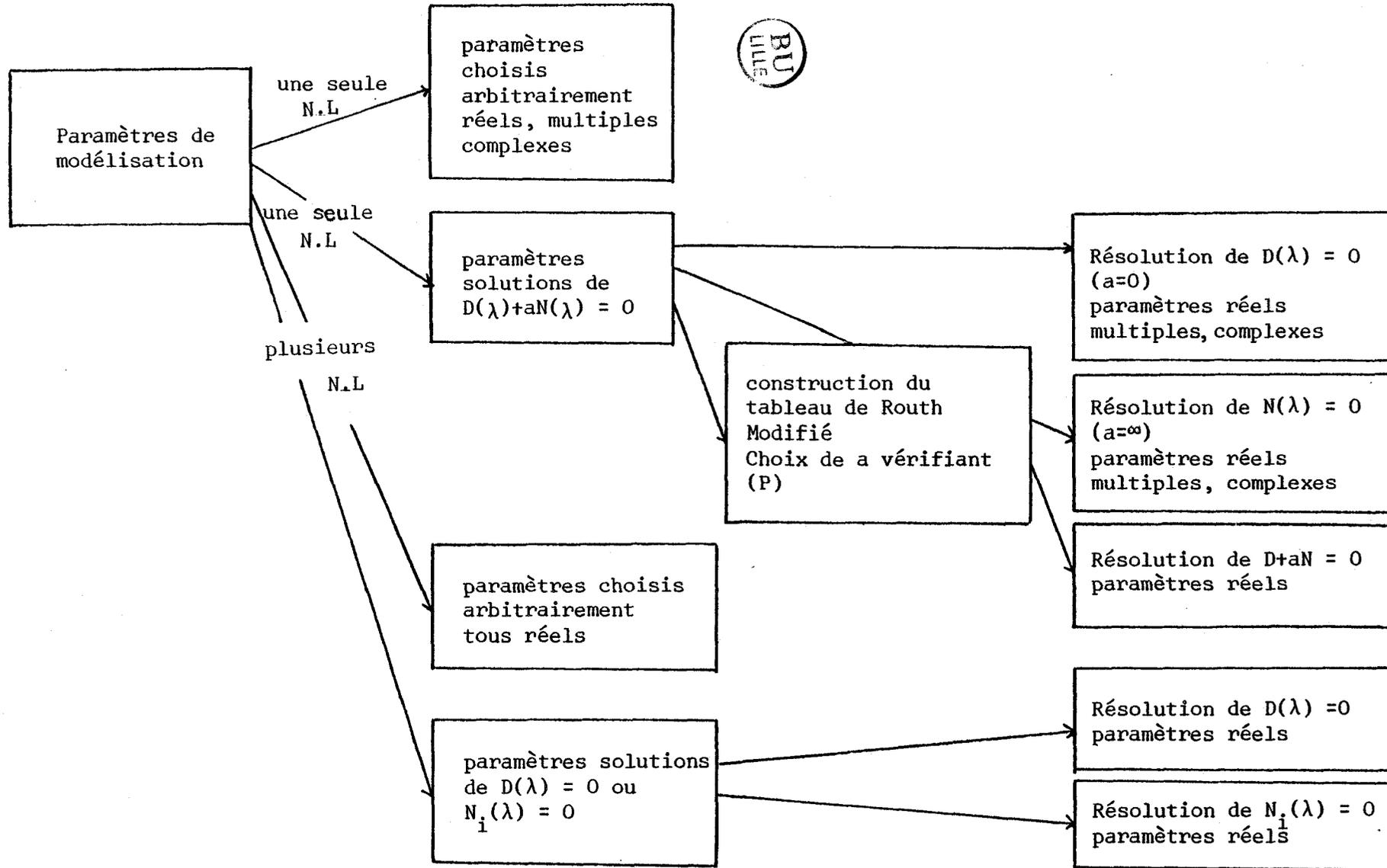


Fig. V.3.

## STRUCTURE DU MODULE DE MODELISATION.

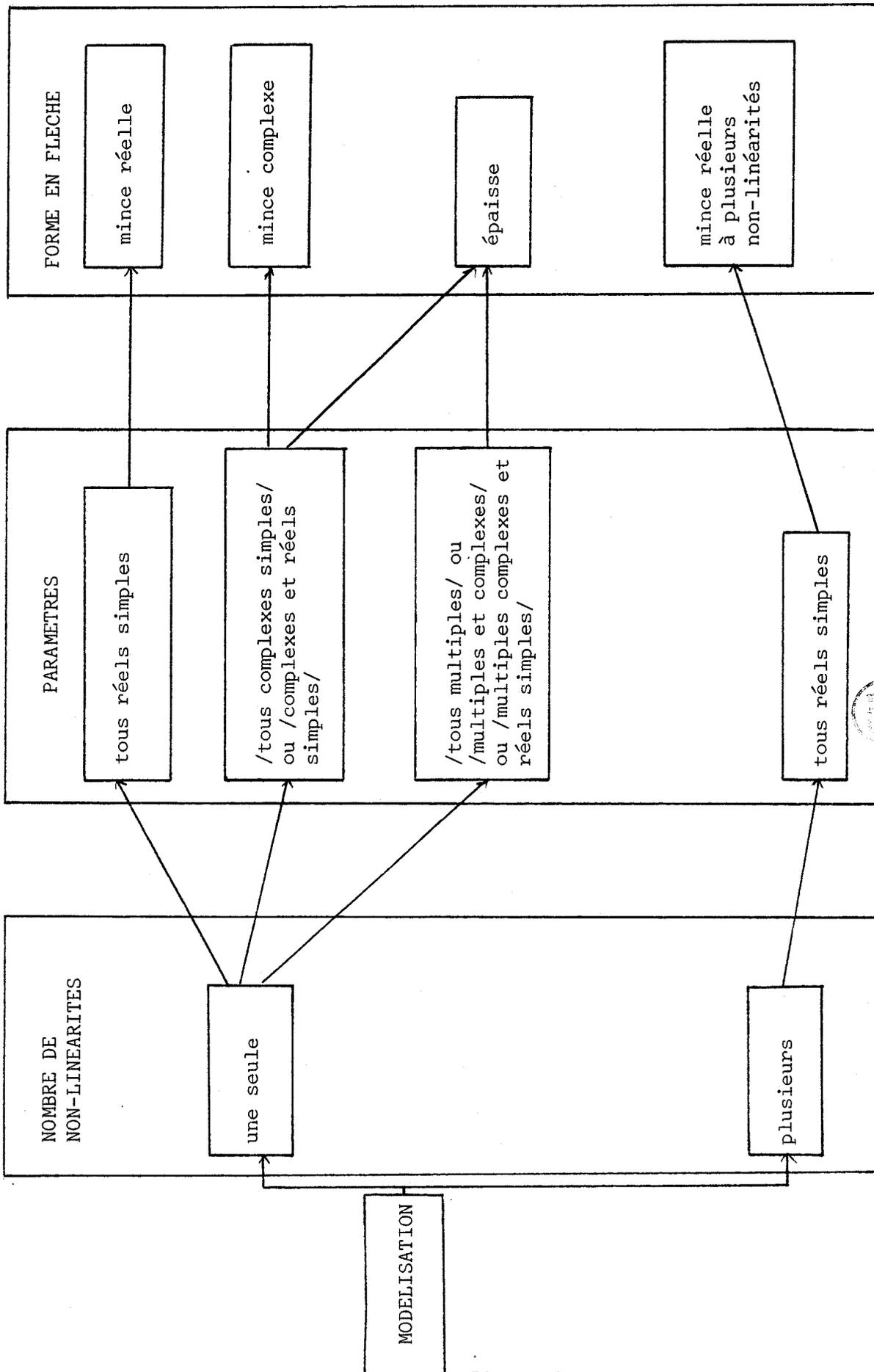


Fig. V.4.

## STRUCTURE DU MODULE D'ETUDE DE LA STABILITE.

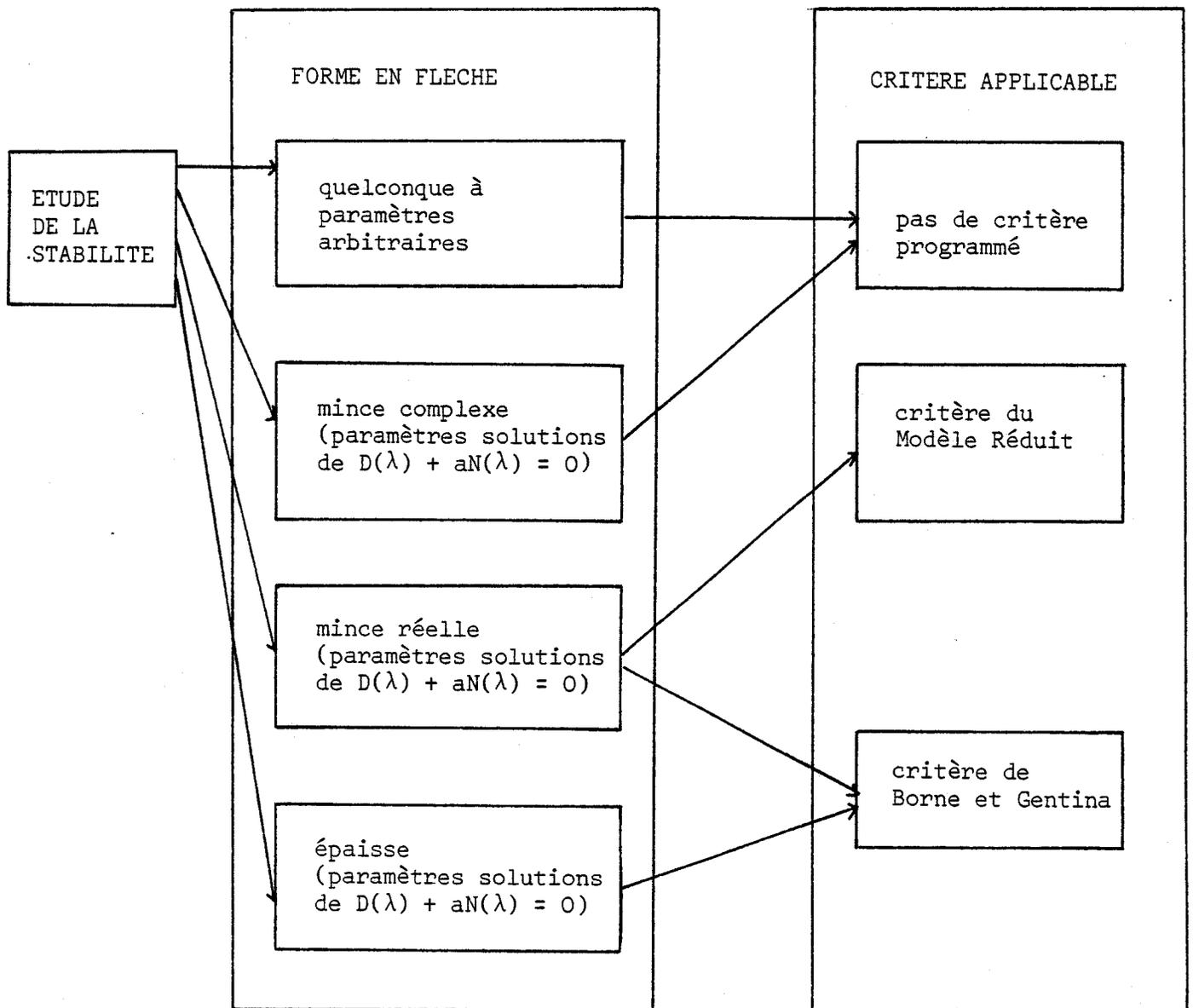


Fig. V.5.

## I.2 - Aspect conversationnel du programme. Mode d'emploi

Afin d'établir un dialogue simple avec le calculateur, le programme possède une structure conversationnelle, que nous détaillons dans cette partie.

Les informations sont fournies au système sous forme de questions-réponses à orientations différentes.

Elle permettent en effet soit un choix de type de données, soit une confirmation de ce choix, soit un appel à une directive ou une entrée de paramètres.

### I.2.1 - Données et résultats.

Les données à entrer peuvent être des paramètres du système ou certaines options fixant l'évolution des modules.

Nous détaillons pour chacun des modules l'expression des données en énonçant les diverses questions fournies par le calculateur ainsi que les différentes réponses.

---

## PRESENTATION DU PROGRAMME LIMA

---

### Affichage du menu.

MENU

====

#### 1. DECLARATION DU SYSTEME

- Saisie des donnees
  - Schems bloc
  - Matrice d'évolution
- Lecture des donnees dans fichier
- Calcul du polynome symbolique

#### 2. PARAMETRES DE MODELISATION

- Choix arbitraire
- Solutions de  $cD(P) + dN(P) = 0$

#### 3. MODELISATION

- FORME EN FLECHE
  - Mince reelle
  - Mince complexe
  - Epaisse

#### 4. ETUDE DE LA STABILITE

- Modele reduit
- Critere de Borne & Gentina

Identification du fichier.

L'utilisateur entre un nom de fichier :

NOM DE FICHER-SYSTEME ENTRE ' ', PUIS RETURN  
===>

Si le fichier est nouveau, l'utilisateur confirme :

NOUVEAU FICHER ?

CONFIRMEZ PAR ' ' OU 'N'  
===>

S'il est ancien, il confirme et le contenu du fichier s'affiche sur l'écran  
ou il ne confirme pas et attribue un autre nom de fichier

ANCIEN FICHER ?

CONFIRMEZ PAR ' ' OU 'N'  
===>

Remarques.

Dans le déroulement du programme, toute valeur entrée doit être confirmée  
grâce à la question suivante :

CONFIRMEZ PAR ' ' OU 'N'  
===>

D'autre part, tous les paramètres entrés ou les résultats obtenus peuvent être stockés dans le fichier si l'opérateur le désire. Il lui suffit de répondre à la question suivante :

VOULEZ VOUS STOCKER LES DONNEES ?  
TAPEZ ' ' OU 'N'==>

---

## DEFINITION

---

### Option.

L'utilisateur précise la façon dont est défini le système :

LE SYSTEME EST DEFINI PAR:

1. Son schéma bloc
2. Sa matrice d'évolution

TAPER '1' ou '2' ==>

### Paramètres.

Cas 1 (Système défini par son schéma bloc).

Ordre du système

Entrer Q , ordre du systeme ==>

Coefficients du schéma bloc :

Entrer les coefficients  $A(i)$  de  $D(P)$ , puis  $B(i)$  de  $N(P)$  comme suit :

$A_0, \dots, A_{q-1}, B_0, \dots, B_{q-1}$  ==>

Cas 2 (Système défini par sa matrice)

Ordre du système.

Entrer Q , ordre du systeme ==&gt;

CONFIRMEZ PAR ' ' OU 'N'  
===>

Nombre de non-linéarités.

Entrer N , nombre de non\_linearites ==&gt;

CONFIRMEZ PAR ' ' OU 'N'  
===>

Coefficients des matrices

## SAISIE DES COEFFICIENTS DE LA MATRICE C (qxq) LIGNE/LIGNE

Introduire l'element 1, 1 de la Matrice :  
 Introduire l'element 1, 2 de la Matrice :  
 Introduire l'element 1, 3 de la Matrice :  
 Introduire l'element 2, 1 de la Matrice :  
 Introduire l'element 2, 2 de la Matrice :  
 Introduire l'element 2, 3 de la Matrice :  
 Introduire l'element 3, 1 de la Matrice :  
 Introduire l'element 3, 2 de la Matrice :  
 Introduire l'element 3, 3 de la Matrice :

## SAISIE DES COEFFICIENTS DE LA MATRICE CNL(qxn) LIGNE/LIGNE

Introduire l'element 1, 1 de la Matrice :  
 Introduire l'element 1, 2 de la Matrice :  
 Introduire l'element 2, 1 de la Matrice :  
 Introduire l'element 2, 2 de la Matrice :  
 Introduire l'element 3, 1 de la Matrice :  
 Introduire l'element 3, 2 de la Matrice :

Résultats.

Les coefficients du polynôme symbolique s'affichent sur l'écran.

Exemple à une seule N.L.

L'expression du polynôme symbolique est de la forme:  $D(P) + f.N(P)$   
avec

$$D(P) = P^{**4} + 0.7000000E+01 * P^{**3} + 0.1800000E+02 * P^{**2} \\ + 0.2200000E+02 * P^{**1} + 0.1200000E+02 * P^{**0}$$

et.

$$N(P) = 0.1000000E+01 * P^{**1} + 0.1000000E+01 * P^{**0}$$

Exemple à deux N.L.

L'expression du polynôme symbolique est de la forme:  $D(P) + f_1.N_1(P) + \dots + f_n.N_n(P)$   
ou  $n = 2$  et

$$D(P) = P^{**3} + 0.1000000E+01 * P^{**2} + -0.6000000E+01 * P^{**1} \\ + -0.7000000E+01 * P^{**0}$$

et

$$N_1(P) = 0.1000000E+01 * P^{**2} + -0.2000000E+01 * P^{**1} + -0.3000000E+01 * P^{**0}$$

$$N_2(P) = 0.1000000E+01 * P^{**2} + -0.2000000E+01 * P^{**0}$$

---

 PARAMETRES
 

---

Option.

- L'utilisateur précise si les paramètres sont arbitraires ou non.

## CHOIX DES PARAMETRES

1. lambda(i) choisis arbitrairement
2. lambda(i) solutions de  $cD(P) + dN(P) = 0$

TAPER '1' OU '2' suivant votre choix ==>

•S'il tape '2', 3 possibilités sont offertes :

3 cas sont envisageables

1.  $\lambda(i)$  solution de  $D(P)=0$  ( $c=0$ )
2.  $\lambda(i)$  solution de  $N(P)=0$  ( $d=0$ )
3.  $\lambda(i)$  solution de  $D(P)+aN(P)=0$  ( $a=d/c$ )

CAS CHOISI, TAPER '1', '2' OU '3'

===>

•Pour '1' et '2', différents types de paramètres sont proposés

SAISIE DES PARAMETRES LAMBDA(I)

1.  $\lambda(i)$  tous réels
2.  $\lambda(i)$  complexes simples  
complexes simples & réels simples
3.  $\lambda(i)$  réels multiples  
réels multiples & complexes simples

TAPER '1', '2' OU '3'

===>

•Pour '3', un seul type possible : paramètres tous réels.

SAISIE DES PARAMETRES LAMBDA(I)

1.  $\lambda(i)$  tous réels

- Il s'agit des solutions de  $D+aN = 0$  déterminées par le tableau de ROUTH modifié. Un intervalle solution pour  $a$  est déterminé dans lequel diverses valeurs de  $a$  peuvent être sélectionnées.

L'utilisateur choisit une valeur de  $a$  :

Entrer la valeur de  $a$  choisie ==>

- Il peut sélectionner d'autres paramètres (pour une valeur de  $a$  différente) :

Desirez vous d'autres valeurs de  $\lambda(i)$  (' ' 'N') ?

- Si l'ordre du système est supérieure à 4, on ne peut résoudre  $D+aN = 0$

L'ordre du système est trop grand .Le tableau de Routh ne permet pas de trouver les solutions de  $D(P)+a N(P)=0$

On choisit dans ce cas des paramètres  $\lambda$  arbitraires

• Dans le cas '1' et '2', les paramètres peuvent être complexes, on a alors deux formes en flèche possible.

Dans ce cas ,deux formes en fleche sont envisageables:

1. FLECHE EPAISSE REELLE
2. FLECHE MINCE COMPLEXE

TAPER '1'OU '2' suivant votre choix ==>



### Paramètres.

L'utilisateur entre la valeur des paramètres en respectant le format propre au type de paramètres :

• type réel

ENTRER les  $\lambda(i)$  ( $i=1, iQ-1$ ), separees par une virgule  
==>

•type réel multiple, complexe

ENTRER dans l'ordre le nombre de réels ,de complexes,separés par une virgule  
 ==>

CONFIRMER ' ' OU 'N'  
 ==>

ENTRER la valeur des  $\lambda(i)$  réels puis leur ordre respectif,separés par une virgule  
 ==>

ENTRER les parties réelles des  $\lambda(i)$ ,les parties imaginaires puis leur ordre respectif,separés par une virgule  
 ==>

### Résultats.

Le programme fournit les résultats intermédiaires permettant le choix des paramètres

•racines de  $D(p) = 0$  et  $N(p) = 0$

par exemple :

LAMBDA REEL X = -0.1655D-02

LAMBDA REEL X = -0.3661D+01

LAMBDA COMPLEXES Z = -0.3752D+01 +/- J 0.3665D+01

Les paramètres sont choisis parmi ces valeurs de  $\lambda(i)$

•racines de  $D(p) + aN(p) = 0$

-tableau de Routh Modifié

\*\*\*\*\*  
 \* TABLEAU DE ROUTH MODIFIE \*  
 \*\*\*\*\*

Numerateur de la fonction de transfert : N(p)

$$N(p) = 0.1000000D+02 * p ** 2 + 0.2000000D+02 * p ** 1 + 0.2000000D+02 * p ** 0$$

Denominateur de la fonction de transfert : D(p)

$$D(p) = 0.1000000D+02 * p ** 3 + 0.4000000D+02 * p ** 2 + 0.4000000D+02 * p ** 1$$

PREMIERE COLONNE DU TABLEAU DE ROUTH MODIFIE

Ligne 1

Le polynome est reduit a une constante

$$P 1(a) = 0.1000000D+02 * a ** 0$$

Ligne 2

Le polynome est reduit a une constante

$$P 2(a) = 0.3000000D+00 * a ** 0$$

Ligne 4

Le desre du polynome est: 2

$$P 4(a) = 0.2000000D+00 * a ** 2 + 0.4000001D+00 * a ** 1 + 0.8000001D+00 * a ** 0$$

Ligne 6

Le desre du polynome est: 4

$$P 6(a) = -0.3600001D-01 * a ** 4 + -0.7200001D-01 * a ** 3 + -0.3960001D+00 * a ** 2 + 0.5760001D+00 * a ** 1 + -0.3191891D-14 * a ** 0$$

DIAGONALE DU TABLEAU DE ROUTH MODIFIE

Ligne 1

Le desre du polynome est: 1

$$P 1(a) = 0.2000000D+02 * a ** 1 +$$

Ligne 2

Le desre du polynome est: 1

$$P 2(a) = 0.2000000D+00 * a ** 1 + 0.4000000D+00 * a ** 0$$

Ligne 4

Le desre du polynome est: 2

$$P 4(a) = 0.2000000D+00 * a ** 2 + -0.6000000D+00 * a ** 1 + 0.1600000D+01 * a ** 0$$

Ligne 6

Le desre du polynome est: 4

$$P 6(a) = -0.3600001D-01 * a ** 4 + -0.7200001D-01 * a ** 3 + -0.3960001D+00 * a ** 2 + 0.5760001D+00 * a ** 1 + -0.3191891D-14 * a ** 0$$

CALCUL DES RACINES ; POLYNOMES DE LA PREMIERE COLONNE  
\*\*\*\*\*

Ligne 4 ;Polynome de degre 2  
\*\*\*\*\*

Racines complexes Z =-.10000000372528969D+01 +/- J 0.17320508290748468D+01

Ligne 6 ;Polynome de degre 4  
\*\*\*\*\*

Racine reelle X =0.11078127177489362D+01

Racine reelle X =0.55372373353179682D-14

Racines complexes Z =-.15539063588744654D+01 +/- J -.34681766796046660D+01

TRI DES RACINES PAR ORDRE CROISSANT  
\*\*\*\*\*

Nombre total de racines distinctes: 2  
-----

Racines trieés :  
-----

5.5372373353179682E-15 1.107812717748936

CALCUL DES RACINES ; POLYNOMES DE LA DIAGONALE  
\*\*\*\*\*

Ligne 1 ;Polynome de degre 1  
\*\*\*\*\*

Racine reelle X =0.0000000000000000D+00

Ligne 2 ;Polynome de degre 1  
\*\*\*\*\*

Racine reelle X =-.2000000000000000D+01

Ligne 4 ;Polynome de degre 2  
\*\*\*\*\*

Racines complexes Z =0.14999998882413038D+01 +/- J 0.23979158315662533D+01

Ligne 6 ;Polynome de degre 4  
\*\*\*\*\*

Racine reelle X =0.11078127177489362D+01

Racine reelle X =0.55372373353179682D-14

Racines complexes Z =-.15539063588744654D+01 +/- J -.34681766796046660D+01

TRI DES RACINES PAR ORDRE CROISSANT  
\*\*\*\*\*

Nombre total de racines distinctes: 3  
-----

Racines trieés :  
-----

-2.000000000000000 0.000000000000000E+00 1.107812717748936

\*\*\*\*\*

TRI DES RACINES PAR ORDRE CROISSANT  
 =====

Nombre total de racines distinctes: 3  
 -----

Racines triees :  
 A 1=-0.2000000D+01  
 A 2= 0.0000000D+00  
 A 3= 0.1107813D+01

TABLEAU DE VARIATION  
 =====

Nombre de changements de signes dans la premiere colonne  
 -----

+	+	+	+
+	+	+	+
+	+	+	+
-	-	+	-

1	1	0	1
---	---	---	---

Nombre de changements de signes dans la diagonale  
 -----

-	-	+	+
-	+	+	+
+	+	+	+
-	-	+	-

2	2	0	1
---	---	---	---

-intervalle solution pour a.

CAS OU L=F=0 ET N=0

-----  
 Intervalle possible :  $0.0000000D+00 < a < 0.1107813D+01$

- solutions possibles

par exemple :

Les parametres negatifs verifiant  $D(P)+a.N(P)$  sont:

lambda 1=-0.5857864D+00

lambda 2=-0.1000000D+01

lambda 3=-0.3414214D+01



---

## MODELISATION

---

Dans ce module, aucune question n'est posée à l'utilisateur. Les résultats concernant la forme en flèche lui sont fournis.

### Résultats.

Le format de présentation est le suivant :

- forme de la matrice en flèche
- colonne des paramètres et dernière colonne de la matrice.

---

## STABILITE

---

### Option.

Lorsque la matrice est en flèche mince réelle l'utilisateur peut choisir le critère qu'il désire essayer.

LA MATRICE EST EN FLECHE MINCE REELLE

Deux criteres sont applicables

1. critere du MODELE REDUIT
2. critere de BORNE & GENTINA.

TAPER '1' OU '2' suivant votre choix

==>

Il fixe ensuite le signe de  $f^*$  - a ou  $f^*$

Choix du signe de  $f^*$ -a AVEC A= 1.00

-----

1.  $f^* - 1.00 > 0$
2.  $f^* - 1.00 < 0$

Choix du signe de  $f(x)$

-----

1.  $f(x) > 0$
2.  $f(x) < 0$

TAPER '1' ou '2' suivant votre choix:

Une fois le signe fixé, il peut essayer l'autre cas.

Desirez vous essayer l'autre cas  $F(x) < 0$  ?

TAPER ' ' OU 'N' suivant votre choix:

Résultat.



Suivant la forme de la matrice, des critères choisis, les résultats obtenus se présentent comme suit :

Pour le critère de Borne et Gentina :

La valeur du déterminant est:  $\text{Det } M = (0.0000E+00) + f(x) * (-7.000$

CONDITIONS A REMPLIR :  $f(x) < 0.00$  et  $\text{Det } M > 0$

-----

CONCLUSION :  $f(x) < 0.0000E+00$

La valeur du déterminant est:  $\text{Det } M = (0.0000E+00) + f(x) * ( 7.000$

CONDITIONS A REMPLIR :  $f(x) > 0.00$  et  $\text{Det } M > 0$

-----

CONCLUSION :  $f(x) > 0.0000E+00$

Pour le critère du modèle réduit :

L'ORDRE DU SYSTEME REDUIT EST : 2

COLONNE DES PARAMETRES

COLONNE DES  $K(I)*f(*)$

LAMBDA 1= -7.00000

-9.00000 + 20.0000 \*  $f(*)$   
 0.000000E+00 \*  $f(*)$

La valeur du determinant est:

Det M= ( 63.00 ) +  $f(*) * (-20.00$  )

CONDITIONS A REMPLIR :  $f(*) > 0.00$  et Det M > 0

CONCLUSION :  $0.00000E+00 < f(*) < 3.150$

### 1.2.2 - Messages d'erreurs.

Dans un système conversationnel, il faut prévoir une solution lorsque l'utilisateur ne répond pas correctement à la question et faire apparaître clairement les messages d'erreur.

Ainsi lorsqu'une réponse est mal formulée, la question est immédiatement reposée.

Lors de l'entrée des paramètres du système, si l'utilisateur fait une erreur, un message d'erreurs s'affiche.

Les messages d'erreurs signalés sont :

ERREUR,NOMBRE DE PARAMETRES INCORRECT

ERREUR:les paramètres doivent être distincts

ERREUR:la partie imaginaire est positive

### 1.2.3 - Directives.

Le dialogue utilisateur-calculateur s'effectue également à l'aide de directives permettant l'accès à chaque module du programme.

Leur rôle est décrit dans le tableau figure (V.6).

TYPE DE DIRECTIVE	FONCTION	SYNTAXE
DIRECTIVE D'AIDE	• liste des directives	'AIO'
DIRECTIVE D'ARRET	• arrêt du programme	'FIN'
DIRECTIVE DE DEFINITION	• entrée des données du système • définition du polynôme symbolique	'DEF'
DIRECTIVE DE CHOIX DE PARAMETRES	• résolution d'équations polynomiales • tableau de Routh modifié	'PAR'
DIRECTIVE DE MODELISATION	• flèche mince réelle • flèche mince complexe • flèche épaisse • flèche mince à plusieurs N.L	'MOD'
DIRECTIVE DE CHOIX DE CRITERES DE STABILITE	• critère de Borne et Gentina • critère du modèle réduit	'STA'

Fig. V.6.

La directive d'aide fournit sur l'écran les renseignements suivants :

- 'DEF' : DEFINITION D'UN NOUVEAU SYSTEME
- 'PAR' : CHOIX DES PARAMETRES DE MODELISATION
- 'MOD' : MODELISATION PAR UNE MATRICE EN FLECHE
- 'STA' : CHOIX D'UN CRITERE DE STABILITE
- 'AID' : LISTE DES COMMANDES DISPONIBLES
- 'FIN' : ARRET DU PROGRAMME

A la fin de chaque module s'affiche une série de directives permettant de continuer le programme. L'utilisateur en choisit une et fait évoluer le programme comme il le souhaite en respectant l'enchaînement des modules décrit figure (V.7).

POUR CONTINUER ,TAPER 'PAR','DEF','FIN','AID' suivant votre choix:

TAPER 'MOD','PAR','DEF','FIN','AID' suivant votre choix:

TAPER 'STA','PAR','DEF','FIN','AID' suivant votre choix:

## II - DESCRIPTION DU PROGRAMME.

Le programme principal a pour rôle de relier les différents modules décrits précédemment entre eux. A chaque module correspond un sous-programme que nous appelons sous-programme primaire (en abrégé SPP). Dans le programme principal sont appelés successivement les sous-programmes primaires que l'on désire traiter, par exemple le sous-programme DEFSYS qui définit les données du système.

De plus, chaque sous-programme primaire appelle différents sous-programmes secondaires (en abrégé SPS).

Dans cette partie sont décrits le programme principal, les sous-programmes primaires et secondaires.

### II.1 - Le programme principal

---

#### PROGRAMME PRINCIPAL

---

##### Déclaration des variables.

\*type\_parameter

PARAMETER (IDIM = ..) : IDIM est le majorant de la dimension des tableaux utilisés.

IDIM = 20. Si l'utilisateur désire modifier l'ordre du système et par conséquent la dimension des tableaux, il doit changer la valeur de IDIM dans le programme.

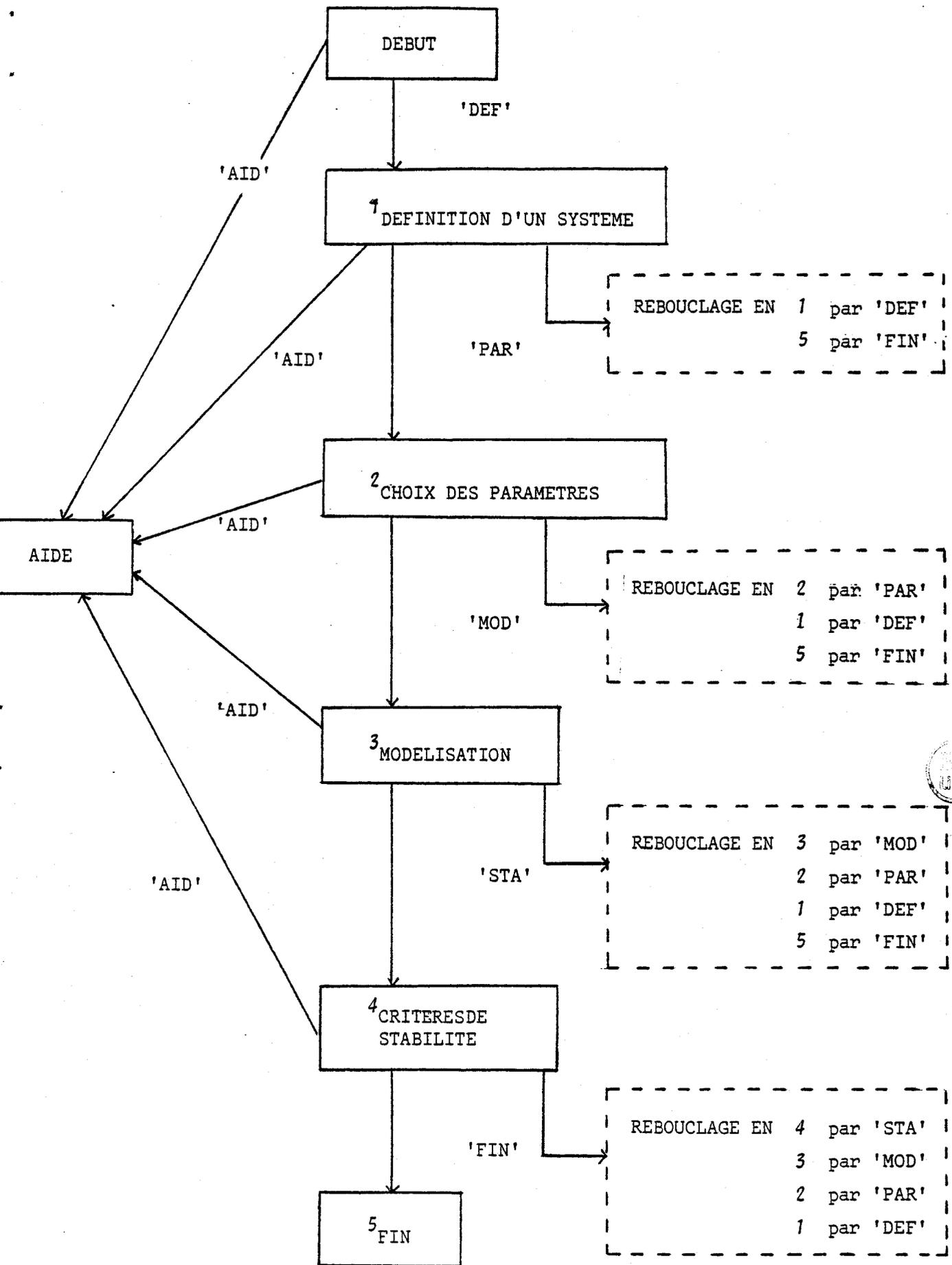


Fig. V.7.

type caractère

DIR : nom des directives

NOMFIC : nom du fichier  
Il comprend au plus 9 lettres

CONFIR : validation

CDEF : définition du système

GENR : choix des paramètres

CSOL : choix de polynôme

CTYP : forme de la matrice

CPAR : choix des paramètres

•type logique

NEW : test sur fichier

- 'DEF'
- 'PAR'
- 'MOD'
- 'STA'
- 'AID'
- 'FIN'

- ' '
- 'N'

- '1' •schéma bloc
- '2' •matrice

- '1' •tous réels
- '2' •complexes simples
- complexes et réels simples
- '3' •réels multiples
- réels multiples et complexes simples

- '1' •paramètres solutions de  $D(p) = 0$
- '2' •paramètres solutions de  $N(p) = 0$
- '3' •paramètres solutions de  $D(p) + aN(p) = 0$

- '1' •flèche épaisse
- '2' •flèche mince complexe

- '1' •arbitraire
- '2' •solutions  $\text{dec}D(p) + dN(p) = 0$

- TRUE• fichier neuf
- FALSE• fichier ancien

•Tableau type réel (à une dimension)

DPOS	: coefficients de D(p), N(p) rangés par puissance croissante	}	cas où le système est défini par son schéma bloc
DPOS6	: coefficients de D(p), N(p) rangés par puissance croissante		
DPOS2	: coefficients des puissances de N(p) ou N1(p) rangés par puissance décroissante		
DPOS7	: coefficients des puissances de NI(p) (I≠1) rangés par puissance décroissante		
CC	: coefficients des puissances de D(p) rangés par puissance décroissante		
DONR	: coefficients de modélisation, réels		
DONI	: coefficients de modélisation, complexes		
DONSOR	: résultats, coefficients de la dernière colonne de la matrice		

•Tableau type réel double précision (à une dimension)

CC1	: coefficients des puissances de D(p) rangés par puissance croissante	}	cas où le système est défini par sa matrice
DPOS5	: coefficients des puissances de N(p) rangés par puissance croissante		

Sous programmes appelés.

DEFSYS  
LISTCOM  
PARSYS  
MODSYS  
STABSYS

Description.

Les différentes tâches de programme principal sont :

- affichage du menu
- création d'un fichier ou identification d'un ancien fichier.
- appel des différents modules
- appel du module LISTCOM

Remarque.

Le plupart des variables du programme principal constituent les paramètres d'appel des sous-programme.

## II.2 - Les sous-programmes primaires.

On définit dans cette partie chacun des SP comme suit :

- on précise la façon dont est appelé le S.P.
- on donne la fonction de chacun des paramètres d'appel.
- on déclare toutes les variables propres au SP sans répéter celles définies précédemment pour un autre SP.
- on énonce tous les SP appelés par le SP considéré
- on décrit la fonction du SP.

## II.2.1 - SP de définition du système.

---

 DEFSYS
 

---

Appel.

call DEFSYS (NEW, IQ, CDEF, DPOS, CC, DPOS2, DPOS7, NBNL).

Paramètres.

Les entiers IQ et NBNL représentent respectivement l'ordre et le nombre de non-linéarités du système.

Les autres paramètres ont été définis dans la partie précédente §II.1.

Déclaration des variables.

## •Type caractère

POLSYM : expression du polynôme symbolique 'D(p)+fN(p)' ou  
'D(p) + f<sub>1</sub>.N<sub>1</sub>(p) + ... + f<sub>n</sub>.N<sub>n</sub>(p)'

CHAR : variable permettant de distinguer dans le fichier si le système est défini par son schéma bloc ou sa matrice.

ERRE : validation

$$\begin{bmatrix} 'N' \\ ' ' \\ ' ' \end{bmatrix}$$

CSTOC : stockage des données dans fichier

$$\begin{bmatrix} 'N' \\ ' ' \\ ' ' \end{bmatrix}$$

• Tableau type entier (à une dimension)

GG : puissances de p correspondant à un coefficient non nul de D(p)

DG : même chose pour N(p).

• Tableau type réel (à une dimension)

DPOS1 : même tableau que DPOS en enlevant les coefficients nuls

(à deux dimensions)

CNL : coefficients des puissances de NI(p) rangés par puissance décroissante

MTCNL : matrice des coefficients des non-linéarités

MTC ; matrice des coefficients constants.

Sous programmes appelés.

LIRMAT

IMPMAT

IMPMAF

LECTMAF

ECRIT1

POSNI

Description.

Le rôle de ce SP est de définir le système :

- Il permet d'entrer les données soit sous forme de schéma bloc, soit sous forme matricielle.

Dans ce dernier cas, à l'aide du sous-programme POSNI il est possible d'établir l'expression du polynôme symbolique du système.

- Il transfère les données dans le fichier dont le nom a été déterminé dans le programme principal.

Dans le cas d'un ancien fichier, le sous-programme DEFSYS effectue la lecture des données du système correspondant.

### II.2.2 - SP de choix des paramètres de modélisation.

---

#### PARSYS

---

##### Appel.

```
call PARSYS (IQ, IS, IT, IZ, CDEF, GENR, CSOL, CTYP, CPAR, DPOS
DPOS2, DPOS5, DPOS6, DPOS7, DONR, DONI, CC, CC1, NBNL, AL)
```

##### Paramètres.

##### • type entier :

IS	représente le nombre de réels simples ou multiples
IT	représente le nombre de complexes dans le cas d'une modélisation en flèche épaisse réelle
IZ	représente le nombre de complexes dans le cas d'une modélisation en flèche mince complexe.

##### • type réel :

AL	représente le nombre réel a déterminé par le sous-programme secondaire ROUTH tel que $D(p) + aN(p) = 0$ admette q ou q-1 racines réelles distinctes.
----	--

Les autres paramètres ont été définis dans la partie concernant le programme principal. § II.1.

Déclaration des variables.\*type caractère

ECR : stockage des résultats dans le fichier  $\left[ \begin{array}{l} 'N' \\ ' ' \end{array} \right.$

REP : choix des paramètres (cas d'un système à plusieurs NL)

$\left[ \begin{array}{l} ' ' : \text{paramètres solution de } D(p)=0 \\ 'N' : \text{paramètres arbitraires ou } NI(p)=0 \end{array} \right.$

\*tableau type réel double précision (une seule dimension)

DPOS4 : même tableau que DPOS avec  $DPOS4(IQ+1) = 1$   
en double précision

DPOS3 : coefficients des puissances de  $D(p)$  rangés par puissance  
croissante (cas où le système est défini par son schéma  
bloc)

ROOTR : racines réelles de  $cD(p) + dN(p) = 0$

ROOTI : racines imaginaires de  $cD(p) + dN(p) = 0$ .

\*type complexe

*Définition implicite.* Toute variable commençant par la lettre F est considérée  
comme complexe.

FDELTA : test sur égalité de deux paramètres.

Sous-programmes appelés.

CALRAC  
ROUTH  
PARRES.

Description.

Ce SP réalise les tâches suivantes :

- définition des paramètres de modélisation
  - soit par un choix arbitraire
  - soit par calcul (résolution de  $cD(p) + dN(p) = 0$ )

- test sur la validité des paramètres choisis,
- appel du SP d'édition des résultats.

### II.2.3 - SP de modélisation du système.

---

#### MODSYS

---

##### Appel.

call MODSYS (IQ, NBNL, CC1, DPOS5, DPOS, DPOS6, DONR, DONI, IS, IT, IZ, CDEF, DPOS7, DONSOR, GENR, CSOL, CTYP, CPAR).

##### Paramètres.

Tous les paramètres ont été définis dans les SP précédents (cf. § II.1 et II.2.2).

##### Déclaration des variables.

##### Type caractère.

TYPE : type de la matrice

'flèche mince réelle'
'flèche mince complexe'
'flèche mince à plusieurs NL'
'flèche épaisse réelle'

##### Tableau type réel (à une seule dimension)

DPOST	:	fusion des tableaux CC1 et DPOS5
S1	}	: sommes intermédiaires permettant le calcul de $R_j( )$ dans le SPS FEPAIS
S2		
SIG		
DONSRI	:	résultats, coefficients complexes de la dernière colonne (cas d'une flèche mince complexe).

##### Type complexe.

*Définition implicite*: Toute variable commençant par la lettre F est un complexe.

Tableau type complexe.

FDESOR : dénominateurs des termes de la dernière colonne de la matrice (cas d'une flèche mince complexe)  
 FTC : termes constants complexes de la dernière colonne  
 FTNC : termes non constants complexes de la dernière colonne.

Sous-programmes appelés.

FMINCE  
 FMNL  
 FMCOMP  
 FEPAIS  
 MODRES  
 MODREF

Description.

Ce SP permet la modélisation du système. Il réalise :

- l'appel des différents sous-programmes de modélisation suivant la forme en flèche désirée.
- l'appel des sous-programmes d'édition des résultats sur l'écran et dans le fichier.

## II.2.4 - SP d'étude de la stabilité du système.

---

 STABSYS
 

---

Appel.

call STABSYS (IQ, IS, IT, CPAR, GENR, CTYP, CSOL, DONR, DONI, DONSOR, AL, NBNL).

Paramètres.

Tous les paramètres ont été définis dans les parties décrivant le programme principal (§.II.1) et le SPP PARSYS (§.II.2.2.).

Déclaration des variables.Type caractère.

CRIT	:	choix du critère	[ '1' critère du modèle réduit '2' critère de Borne et Gentina
------	---	------------------	---

Type logique.

TES	:	test sur $\mu_i +  v_i $	[ •TRUE• $\mu_i +  v_i  = 0$ •FALSE• $\mu_i +  v_i  \neq 0$
-----	---	--------------------------	--

Sous programmes appelés.

CRITBG  
MODRED

Description.

Les fonctions de ce sous-programme sont les suivantes :

- application d'un critère précis sur une forme en flèche donnée par appel des sous-programmes d'étude de la stabilité
- écriture des résultats dans fichier.

## 11.2.5 - SP d'aide.

---

LISTCOM

---

Appel.

call LISTCOM.

Paramètres.

Il n'existe pas de paramètres d'appel.

Déclaration de variables.

Aucune variable n'est utilisée à l'intérieur de ce SP.

Description.

Ce sous-programme affiche sur l'écran les différentes commandes dont dispose l'utilisateur.

Sous-programme appelés.

CRITBG

MDRED

Description.

Les fonctions de ce sous-programme sont les suivantes :

- application d'un critère précis sur une forme en flèche donnée par appel des sous-programmes d'étude de la stabilité
- écriture des résultats dans fichier.

## II.3 - Les sous-programmes secondaires.

Chacun des sous-programmes primaires appelle un ou plusieurs sous-programmes secondaires SPS.

Ces sous-programmes secondaires sont divisés en deux parties :

- la première comporte les sous-programmes utilisés pour la présentation des résultats sur la console ou dans le fichier
- la seconde contient tous les sous-programmes de calcul.

Les SPS sont présentés suivant le plan défini pour les SPP.

Nous commençons par les SPS de calcul.

## II.3.1 - Les sous-programmes de calcul.

II.3.1.1 - Définition du système.

Dans le module de définition du système DEFSYS, un seul sous-programme de calcul est appelé. Il s'agit du sous-programme POSNI.

---

 POSNI
 

---

Appel.

cal POSNI (IQ, CC, CNL, MTC, MTCNL, NBNL)

Paramètres.

IQ CC	}	ces paramètres ont été définis dans le paragraphe II.1.
NBNL CNL MTC MTCNL	}	ces paramètres ont été définis dans la partie concernant le SPP DEFSYS II.2.1.

Déclaration des variables.Type entier

Q : ordre du système (= IQ)  
 N : nombre de non-linéarités (= NBNL)

Tableau type réel (à une dimension)

ALPHA : coefficients du polynôme de la matrice MCNL pour une non linéarité donnée  
 BETA : coefficients du polynômes caractéristique de la matrice MCR.

(à deux dimensions)

MC : : même chose que MTC  
 MCNL : : même chose que MTCNL  
 MCR : : matrice MC à laquelle on a enlevé la dernière ligne et la dernière colonne  
 MALPHA : matrice dont les colonnes correspondent aux vecteurs ALPHA pour l'ensemble des non-linéarités

Sous-programmes appelés.

IMPMAT (description donnée au § II.3.2)

LEVER

Description.

Ce sous-programme effectue le rangement des coefficients des polynômes caractéristiques des matrices MC et MCNL dans les tableaux CC et CNL.

Ce sous-programme appelle un autre sous-programme de calcul LEVER dont les caractéristiques sont les suivantes :

---

LEVER

---

Appel.

call LEVER (MC,N,CC)

call LEVER (MCR, NM1, BETA)

call LEVER (MC,N,ALPHA)

Paramètres.

Ils ont été définis dans la partie précédente traitant le SP POSNI.

Déclaration des variables.•Type entier.

ORDRE : dimension de la matrice

•Tableau type réel (à une dimension)

CDEF : coefficients  $a_i$  (chap II § II.3.3)

(à deux dimensions)

MAT : matrice initiale

B : matrice de calcul intermédiaire

P : matrice produit MAT\*B

chap II § II.3.3

Description.

Ce sous-programme permet la réalisation de l'algorithme de Leverrier et calcule par conséquent le polynôme caractéristique d'une matrice.

II.3.1.2 - Choix des paramètres du système.

Les deux SPS de calcul sont les sous-programmes CALRAC et ROUTH.

## II.3.1.2.1 - Résolution d'une équation polynômiale.

CALRAC est un sous-programme résolvant une équation polynômiale dans  $\mathbb{C}$  en double précision.

---

 CALRAC
 

---

Appel.

```
call CALRAC (IDIM, DPOS4, IQ, ROOTR, ROOTI, IR, IC)
call CALRAC (IDIM, CC1, IQ, ROOTR, ROOTI, IR, IC)
call CALRAC (IDIM, DPOS3, IQ-1, ROOTR, ROOTI, IR, IC)
call CALRAC (IDIM, DPOS5, IQ-1, ROOTR, ROOTI, IR, IC).
```

Paramètres.

Les seuls paramètres qu'il reste à définir sont :

IR : nombre de racines réelles  
 IC : nombre de racines complexes.

Les autres sont définis la partie II.2.2

Déclaration de variables.Type entier.

K1M : nombre d'itérations  
 K2M : nombre d'itérations  
 N : degré du polynôme

EXP : degré du polynôme.

•Type réel double précision

S	]		
P			
S2		:	variables intermédiaires permettant la mise
P2			en oeuvre de l'algorithme de BAIRSTOW /DURAND 1960/
GS			
GP			
DEL			
DISC	:	discriminant correspondant à la résolution d'une équation du second degré	
AUX	:	opposé de DISC	
D	:	discriminant de l'équation $x^2 - Sx + P = 0$	
TOL	:	précision	

•Tableau type réel double précision (à une dimension)

AI	:	coefficients du polynôme rangés par puissance croissante
B	]	tableaux nécessaires à différents calculs intermédiaires. (cf. annexe).
C		

Description.

Ce sous-programme met en oeuvre la méthode de Bairstow et donne les racines réelles ou complexes d'un polynôme.

II.3.1.2.2 - Résolution de l'équation  $D(p) + aN(p) = 0$

Ce programme tient une place importante dans le sous-programme primaire PARSYS. Il appelle d'autres sous-programmes que nous présentons dans cette partie.

---

ROUTH

---

Appel.

call ROUTH (IQ, DPOS, AL)  
call ROUTH (IQ, DPOS6, AL)

Paramètres.

Définis dans la partie concernant le SPP PARSYS II.2.2.

Déclaration de variables.Type caractère.

OTVAL : choix d'autres valeurs pour  $\lambda_i$   $\left[ \begin{array}{l} '' \\ 'N' \end{array} \right.$

Tableau type caractère.

SIGN : signes des polynômes étudiés dans différents intervalles.

Type entier.

DEG : ordre du système.

SJ : DEG +1.

LAMB : variable qui prend la valeur 1 lorsqu'il est possible de trouver  $q$  ou  $q-1$  racines solutions de  $D(p) + aN(p) = 0$ .

M1 : nombre de coefficients non nuls pour chacun des polynômes.

Tableau type entier (à une dimension)

E : puissances du polynôme correspondant à un coefficient non nul.

FT : nombre de changements de signe dans chaque intervalle pour la première colonne du tableau.

LT : nombre de changements de signe dans chaque intervalle par la diagonale du tableau.

Type réel.

AL : coefficient  $a$  de l'équation  $D(p) + aN(p) = 0$ .

Type réel double précision.

X : valeur choisie à l'intérieur de chaque intervalle pour calculer le signe du polynôme.

MIN : valeur minimale.

MAX : valeur maximale.

Tableau type réel (à une dimension)

D : degrés du polynôme pour chaque ligne du tableau  
 DR : coefficients du polynôme dérivé de  $D(p) + aN(p)$ .

•Tableau type réel double précision.(à une dimension)

CPOS : coefficients du polynôme  $D(p)+aN(p)$  rangés par puissance croissante.  
 SP : racines des polynômes de la diagonale.  
 SC : racines des polynômes de la première colonne.  
 SCO : même tableau que SC dans lequel on a enlevé les racines égales  
 SDO : même tableau que SD dans lequel on a enlevé les racines égales.  
 SS : tableau regroupant SCO et SDO.  
 SO : même tableau que S dans lequel on a enlevé les racines égales.

(à trois dimensions).

A : coefficients des polynômes de chaque case du tableau de Routh. Les trois dimensions I, J, K représentent respectivement le numéro de ligne, le numéro de colonne, l'ordre du coefficient dans le polynôme.  
 (Par puissance croissante).

B : même tableau que A dans lequel on a enlevé les coefficients nuls.

Sous-programmes appelés.

DERPOL  
 ECRIT  
 RESUL  
 TRIRAC  
 ELIMI  
 CALRAC  
 HORN

Description.

Ce sous-programme réalise les fonctions suivantes :

- calcul du degré de chaque polynôme du tableau,
- calcul des coefficients de chaque polynôme,
- impression de chaque polynôme de la première colonne du tableau ligne par ligne,
- impression de chaque polynôme de la diagonale du tableau ligne par ligne,
- calcul des racines réelles (par CALRAC) de chaque polynôme de la première colonne et de la diagonale,
- rangement des racines par ordre croissant,
- étude du signe de chacun des polynômes entre deux racines
- recherche des intervalles possibles pour a,
- choix de a
- résolution de  $D(p) + aN(p) = 0$  dans  $\mathbb{R}$

Remarque importante.

Le sous-programme Routh ne peut être réalisé actuellement que pour un système d'ordre inférieur à 5.

En effet, pour un système d'ordre égal à 5, la dernière ligne du tableau contient les coefficients d'un polynôme de degré 29 (cf §II.2.3.3 du chapitre III).

Ces coefficients sont rangés dans les tableaux A et B à trois dimensions qui tiennent alors une place considérable en mémoire et le programme ne peut s'exécuter.

La solution qui consisterait à garder un seul tableau A en mémoire ne permet pas de résoudre le problème.

Une possibilité serait de stocker les coefficients du tableau A dans un fichier annexe.

Nous ne l'avons pas essayée et nous sommes contents d'utiliser le tableau de Routh modifié pour un système d'ordre faible. Pour un ordre élevé, ce tableau fournit des polynômes de degré trop important et semble inexploitable.

### II.3.1.2.3. Sous-programmes appelés par le SPS ROUTH.

---

#### DERPOL

---

##### Appel.

call DERPOL (DPOS, DR, DEG+1, IDIM)

##### Paramètres.

Ils ont été définis dans la partie concernant le SPS ROUTH.

##### Déclaration des variables.

##### Type entier.

NCP1 : degré du polynôme  
 NDP1 : degré du polynôme dérivé (égal à NCP1-1)  
 CI : puissance du polynôme.

##### Tableau type réel (à une dimension)

C : coefficients du polynôme rangés par puissance croissante.

##### Description.

Ce sous-programme effectue la dérivation formelle d'un polynôme.

---

#### ECRIT

---

##### Appel.

call ECRIT (IDIM, DEG, L, K, A, B, E, M<sub>1</sub>).

Paramètres.

Ils ont été définis dans la partie concernant le SPS ROUTH.

Déclaration des variables.•Type entier.

IMAX : DEG + 1  
 IND : indice qui donne M1, correspondant au nombre de coefficients non nuls.

Description.

Ce sous-programme élimine dans les deux premières lignes du tableau A les coefficients nuls.

Le résultat est rangé dans le tableau B.

D'autre part, l'ensemble des puissances pour lesquelles le coefficient n'est pas nul est formé par le tableau E. Ce sous-programme permet d'afficher les polynômes de la première et deuxième ligne du tableau sans écrire les coefficients nuls.

---

RESUL

---

Appel.

call RESUL (IDIM, L, J, A, D, B, E, M1).

Paramètres.

Ils ont été définis dans la partie concernant le SPS ROUTH.

Déclaration des variables.•Type entier.

IMAX : degré du polynôme de la ligne incrémenté de 1  
 INO : voir ci-dessus.

Description.

Ce sous-programme a le même rôle que le SP ECRIT pour les lignes du tableau autres que la première et la seconde.

---

 TRIRAC
 

---

Appel.

```

call TRIRAC (M, SC)
call TRIRAC (MC, SD)
call TRIRAC (M,S)

```

Paramètres.Type entier.

MC : nombre de racine du polynôme de la diagonale  
 M : nombre de racines du polynôme de la première colonne  
 puis nombre de racines total ( $M = M+MC$ )

SD	]	: paramètres définis dans la partie décrivant le SPS ROUTH (II.3.1.2.2)
SC		
S		

Déclaration de variables.Type réel double précision.

AMI : variable annexe permettant la réalisation d'une permutation dans le tableau S.

Description.

Ce sous-programme range les racines par ordre croissant.

---

 ELIMI
 

---

Appel.

call ELIMI (SC, SCO, M)  
 call ELIMI (SD, SDO, MC)  
 call ELIMI (S, SO, M)

Paramètres.

Ils ont été définis dans la partie concernant le SPS ROUTH.

Déclaration de variables.• Type entier.

PR ] : indices permettant de comparer le contenu de chaque  
 ED ] élément du tableau traité.

Description.

Ce sous-programme élimine les racines nulles ou très petites (précision  $10^{-6}$ ).

---

 CALRAC
 

---

cf §. II.3.2.1.

---

 HORN
 

---

Appel.

call HORN (VAL, X, AI, EXP).

Paramètres.• Type entier.

EXP : déjà défini pour le SPS ROUTH.

Type réel double précision.

VAL : variable dans laquelle est rangé à chaque itération  
le résultat

X : valeur pour laquelle est calculée l'expression du polynôme

Tableau type réel.

Toutes les variables déclarées ont été définies précédemment.

Description.

Ce sous-programme met en oeuvre l'algorithme de HORNER qui permet le calcul d'un polynôme en une valeur X donnée /LEPELLETIER 1982/

II.3.1.3. - Modélisation du système.

Dans le SPP MODSYS, quatre sous-programmes secondaires de calcul sont appelés. Il s'agit des SP : FMINCE, FEPAIS, FMCOMP, FMNL.

---

FMINCE

---

Appel.

```
call FMINCE (IQ, DONR, DONSOR, DPOS)
call FMINCE (IQ, DONR, DONSOR, DPOS6)
call FMINCE (IQ, DONR, DONSOR, DPOST)
```

Paramètres.

DPOST : défini dans le paragraphe concernant MODSYS (§ II.2.3)

<p>IQ DPOS DPOS6 DONR DONSOR</p>	}	<p>définis dans le paragraphe concernant le programme principal (§ II.1)</p>
--	---	--

Déclaration de variables.

Toutes les variables ont été définies.

Description.

Ce sous-programme permet de calculer les termes de la dernière colonne d'une matrice en flèche mince lorsque les paramètres de modélisation sont tous réels.

Il fournit les coefficients constants  $\delta_i$  et non constants  $\gamma_i$  pour chaque ligne de la matrice.

---

FEP AIS

---

Appel.

call FEP AIS (IQ, IS, IT, DONR, DONI, S1, S2, SIG, DONSOR, DPOS)

call FEP AIS (IQ, IS, IT, DONR, DONI, S1, S2, SIG, DONSOR, DPOST).

Paramètres.

Ils ont tous été définis précédemment.

IS	]	définis dans le paragraphe II.2.2
IT		
IZ		

DONR	]	définis dans le paragraphe II.1
DONI		
DPOS		

DPOST	]	définis dans le paragraphe II.2.3
S1		
S2		
SIG		

Déclaration de variables.

Afin de comprendre la signification de chacune des variables, il est conseillé de se référer au paragraphe I.2.2 du chapitre II

Type entier.

FACT : fonction factorielle présente dans l'expression de  $(B)^*$

Type complexe double précision.

CNU : nombre complexe représentant la valeur de  $N(\mu_j + i\nu_j)$

CDE : nombre complexe représentant la valeur de  $D(\mu_j + i\nu_j)$

CPO : valeur de  $\prod_{\ell=1}^t ((\lambda - \mu_\ell)^2 + \nu_\ell^2)$  pour  $\lambda = \mu_j + i\nu_j$

CP1 : valeur de  $\prod_{i=1}^s (\lambda - \lambda_i)^{n_i}$  pour  $\lambda = \mu_j + i\nu_j$

CJ : nombre complexe  $\mu_j + i\nu_j$

CJK : nombre complexe  $(CJ * (1,0))^k$

CSOM 1 : somme -  $\sum_{k=1}^{j-1} \alpha_i^k$

CSOM 2 : somme -  $\sum_{\ell=1}^{p+1} \alpha_i^\ell$  contenue dans  $(B)^*$

CALPHA : terme  $\alpha_i^k$

CRIM : terme  $C_{j-1}^p R_i^{j-p-1} (\lambda)_p! \sum_{\ell=1}^{p+1} \alpha_i^\ell$

CRIK :  $\sum_{p=0}^{k-2} \text{CRIM.}$

Tableau type réel à une dimension.

S1  
S2  
SIG

}] sommes intermédiaires permettant de calculer  $R_i(\lambda)$ .

Description.

Ce sous-programme calcule les termes de la dernière colonne d'une matrice en flèche épaisse lorsque les paramètres de modélisation sont complexes ou multiples où à la fois complexes et multiples.

Il fournit les vecteurs  $\alpha_i$  et  $\beta_j$  (cf. I.2.2.12 chap. III).

---

FMNL

---

Appel.

call FMNL (IQ, DONR, DPOS7, DONSOR).

Paramètres.

Ils ont été définis précédemment dans la partie concernant le programme principal (§ II.1).

Déclaration de variables.

•Tableau type réel. (à une dimension)

XNUNL : coefficients correspondant au calcul de  $N(\lambda_i)$  pour une non-linéarité donnée. A chaque non linéarité est associée un vecteur XNUNL (I)

Description.

Ce sous-programme calcule les termes de la dernière colonne d'une matrice en flèche mince comprenant plusieurs non-linéarités.

---

FMCOMP

---

Appel.

call FMCOMP (IQ, IZ, DONR, DONI, DPOS, FTC, FTNC).

Paramètres.

IQ ]  
IZ ] définis dans le paragraphe II.2.2

DONR ]  
DONI ] définis dans le paragraphe II.1  
DPOS ]

FTC ]  
FTNC ] définis dans le paragraphe II.2.3

Déclaration des variables.

Les variables déclarées ont été définies précédemment.

Sous-programmes appelés.

POLVC

Description.

Ce sous-programme calcule les termes de la dernière colonne d'une matrice en flèche mince complexe correspondant à des paramètres de modélisation complexes. Il fournit les vecteurs  $\alpha_i$  et  $\beta_j$  (cf II.2.2.2 chap III).

---

POLVC

---

Appel.

call POLVC (IDIM, IQ, IZ, IND, DPOS, DONI, UDE, UNU, VDE, VNU).

Paramètres.\*Type entier.

IDIM : défini dans le paragraphe II.1

IQ ]  
IZ ] : définis dans le paragraphe II.2.2.

IND : variable qui s'incrémente dès que l'on passe à l'itération suivante.

• Type réel double précision.

UDE : valeur réelle de  $D(\lambda_i)$ ,  $\lambda_i \in \mathbb{C}$ .  
 VDE : valeur imaginaire de  $D(\lambda_i)$   
 UNU : valeur réelle de  $N(\lambda_i)$   
 VNU : valeur imaginaire de  $N(\lambda_i)$ .

• Tableau type réel.

DONI ] : définis dans le paragraphe II.1.  
 DPOS ]

Déclaration des variables.

• Type réel double précision.

XI ] variables intermédiaires permettant de calculer  
 XIP ] la valeur réelle du polynôme étudié.  
 YI ] variables intermédiaires permettant de calculer la  
 YIP ] valeur imaginaire du polynôme étudié.

Description.

Ce sous-programme calcule la valeur d'un polynôme pour une variable complexe.

II.3.1.4 - Etude de la stabilité du système.

Dans le SPP STABSYS, deux sous-programmes de calcul sont appelés : CRITBG et MODRED.

II.3.1.4.1 - Critère de Borne et Gentina.

---

CRITBG

---

Appel.

call CRITBG (IQ, IS, IT, AL, DONR, DONSOR, CSOL, GENR, ECR)

Paramètres.

IQ	] ] ] ] ]	définis dans le paragraphe II.2.2
IS		
IT		
AL		
ECR		
DONR	] ] ] ]	définis dans le paragraphe II.1
DONSOR		
CSOL		
GENR		

Déclaration de variables.Type caractère.

CHOIX : variable qui prend les valeurs '1' ou '2'  
 '1' signifie que l'on a choisi le cas  $f(*)-a > 0$  ou  $f(*) > 0$   
 '2' signifie que l'on a choisi le cas  $f(*)-a < 0$  ou  $f(*) < 0$

NONLI : représente  $f(*)-a$  ou  $f(*)$

Sous-programmes appelés.

DETFE  
 DETFM  
 STAB1

Description.

Ce sous-programme met en oeuvre le critère de Borne et Gentina et fournit les conditions de stabilité du système.  
 Il appelle les sous-programmes DETFM, DETFE et STAB1 que nous détaillons maintenant.

---

 DETFM
 

---

Appel.

cal DETFM (DONR, DONSOR, IQ, DET1, DET2)

Paramètres.

DONR	] définis dans la partie II.1
DONSOR	
IQ	

- Type réel double précision.

DET1 : coefficient du terme constant du déterminant de la matrice traitée (il s'agit dans ce cas de la pseudo-majorante d'une flèche mince).

DET2 : coefficient de terme non constant du déterminant.

Déclaration de variables.

- Type réel double précision.

P : produit des paramètres de modélisation  $\lambda_i$

SG : somme des coefficients non constants chacun étant divisé par le paramètre de modélisation qui lui correspond

SD : même définition pour les coefficients constants

- Tableau type réel à une dimension.

DONR1 : même tableau que DONR

DONSOR1: même tableau que DONSOR

Description.

Ce sous-programme calcule le déterminant d'une pseudo-majorante d'une forme en flèche mince.

---

 DETFE
 

---

Appel.

call DETFE (DONR, DONSOR, IQ, IS, IT, DET1, DET2)

DONR	] définis dans la partie II.1
DONSOR	
IQ	
IS	
IT	

•Type réel double précision.

DET1	] même signification que dans le SP DETFM avec cette fois le calcul de déterminant d'une pseudo-majorante d'une forme en flèche épaisse.
DET2	

Déclaration de variables.•Type entier.

INP : indice permettant le repérage des coefficients du tableau DONSOR pour des racines multiples.

INT : même rôle dans le cas de racines complexes.

•Type réel double précision.

Afin de comprendre le rôle de chaque variable, il est souhaitable de se référer à l'annexe du chapitre IV.

SD : variable correspondant à  $\sum_{k=1}^{n_i} \sum_{h=1}^k \delta_i^{n_i-k+1} \frac{(-1)^{h+1}}{\lambda_i^h}$

SG : variable correspondant à  $\sum_{k=1}^{n_i} \sum_{h=1}^k \gamma_i^{n_i-k+1} \frac{(-1)^{h+1}}{\lambda_i^h}$

SM1 : prend la valeur de  $\sum_{i=1}^s \delta_i$  SD

- SM2 : prend la valeur de  $\sum_{i=1}^s SG$
- PM : produit  $\prod_{j=1}^t (\mu_j^2 - |v_j|^2)$
- SC1 : correspond à  $\sum_{j=1}^t \frac{\delta'_{j2} + \delta'_{j1}}{\mu_j + |v_j|}$
- SC2 : correspond à  $\sum_{j=1}^t \frac{\gamma'_{j2} + \gamma'_{j1}}{\mu_j + |v_j|}$
- PC : produit  $\prod_{i=1}^s (\lambda_i)^{n_i}$

Description.

Ce sous-programme calcule le déterminant d'une pseudo-majorante d'une forme en flèche épaisse.




---

 STAB1
 

---

Appel.

call STAB1 (CSOL, CHOIX, AL, W, DET1, DET2, ECR).

Paramètres.

- CHOIX : défini dans la partie concernant CRITBG
- AL     ]
- ECR    ]     définis dans le paragraphe II.2.2
- CSOL : défini dans le paragraphe II.1

Type réel double précision.

- W : égal à -DET1/DET2

Description.

Ce sous-programme calcule les conditions de stabilité du système étudié.

### II.3.1.4.2 - Définition d'un modèle réduit.

---

#### MODRED

---

#### Appel.

call MDRED (IQ, AL, DONR, DONSOR, CSOL, ECR).

#### Paramètres.

IQ	]	définis dans le paragraphe II.1
DONR		
DONSOR		
CSOL		
AL	]	définis dans le paragraphe II.2.2
ECR		

#### Déclaration de variables.

Tableau type réel (à une dimension)

DONMR : paramètres de modélisation sélectionnés pour définir le modèle réduit

DONMODR: termes de la dernière colonne de la matrice du modèle réduit.

#### Sous-programmes appelés.

MDRESUL	]	cf paragraphe précédent.
DETFM		
STAB1		

#### Description.

Ce sous-programme détermine le modèle réduit d'un système représenté par une forme en flèche mince.

Parmi les 3 sous-programmes appelés par le SPS MODRED, deux sont également appelés par le SPS CRITBG. Leur description a donc été faite précédemment. Seul reste à détailler le SP MDRESUL.

---

## MDRESUL

---

### Appel.

call MDRESUL (AL, NONLI, CSOL, DONMODR, DONMR, K, IQR, ECR).

### Paramètres.

AL ] : définis dans le paragraphe II.2.2.  
ECR ]

DONMR ] définis ci-dessus.  
DONMODR ]

NONLI : défini pour le SPS CRITBG II.3.1.4.1  
CSOL : défini au paragraphe II.1  
IQR : ordre du système réduit  
K : IQR - 1

### Description.

A l'aide de ce sous-programme sont édités les coefficients du modèle réduit dans le fichier et sur la console.

#### II.3.2 - Les sous-programmes : "édition des résultats".

##### II.3.2.1 - Définition du système.

Le SPP DEFSYS fait appel à 5 sous-programmes de lecture ou écriture des données qui sont : LIRMAT, IMPMAT, IMPMAF, LECTMAF, ECRIT1.

Les sous-programmes LIRMAT et LECTMAF permettent respectivement la lecture d'une matrice sur la visu ou dans un fichier. Les sous-programmes IMPMAT et IMPMAF permettent l'écriture d'une matrice sur la visu ou dans un fichier.

---

## LIRMAT

---

### Appel.

```
call LIRMAT (MTC, IQ, IQ)
call LIRMAT (MTCNL, IQ, NBNL)
```

### Paramètres.

Ils sont définis au paragraphe II.2.1.

### Déclaration de variables.

#### Type entier.

```
MIG      :   nombre de lignes de la matrice
NCOL     :   nombre de colonnes de la matrice.
```

#### Tableau type réel.

```
MAT      :   matrice dont on veut lire les coefficients entiers par
              l'intermédiaire de la console.
```

### Description.

Ce sous-programme lit les coefficients de la matrice que nous entrons.

---

## LECTMAF

---

### Appel.

```
call LECTMAF (MTC, IQ, IQ)
call LECTMAT (MTCNL, IQ, NBNL)
```

### Paramètres.

Ils ont été définis au paragraphe II.2.1.

Déclaration de variables.

Les variables déclarées sont les mêmes que dans le sous-programme LIRMAT.

Description.

Ce sous-programme lit les coefficients de la matrice dans le fichier.

---

 IMPMAT
 

---

Appel.

```
call IMPMAT (MTC, IQ, IQ)
call IMPMAT (MTCNL, IQ, NBNL)
```

Paramètres.

Ils ont été définis au paragraphe II.2.1.

Déclaration de variables.

Les variables déclarées sont les mêmes que dans le sous-programme LIRMAT.

Description.

Ce sous-programme écrit sur l'écran la matrice que l'on étudie.

---

 IMPMAF
 

---

Appel.

```
Call IMPMAF (MTC1, IQ, IQ)
Call IMPMAF (MTCNL, IQ, NBNL)
```

Paramètres.

Ils ont été définis au paragraphe II.2.1.

Déclaration de variables.

Les variables déclarées sont les mêmes que dans le sous-programme IMPMAF.

Description.

Ce sous-programme écrit dans le fichier les coefficients de la matrice.

---

 ECRIT1
 

---

Appel.

```
call ECRIT1 (IDIM, IQ, CC, DPOS1, DG, GG, M2, PI)
call ECRIT1 (IDIM, IQ, DPOS2, DPOS1, DG, GG, M2, PI)
call ECRIT1 (IDIM, IQ, DPOS, DPOS1, DG, GG, M2, PI)
```

Paramètres.

DPOS2	] définis au paragraphe II.1 [
DPOS	
DPOS5	
CC	
IDIM	
DPOS1	] définis au paragraphe II.2.1 [
DG	
GG	

Type entier.

M2 : nombre de coefficients non nuls.

Déclaration de variables.

Toutes les variables ont déjà été définies.

Description.

Ce sous-programme a le même rôle que les sous-programmes appelés par le SPS ROUTH : RESUL et ECRIT. Il permet de donner un tableau DPOS1 à coefficients tous non nuls et deux tableaux DG et GG donnant les puissances respectives de chacun des coefficients pour le dénominateur D(p) et le numérateur N(p) (ou les polynômes NI(p)). Ceci permet ensuite l'affichage des polynômes D(p), N(p) et NI(p) en écrivant uniquement les coefficients non nuls.

II.3.2.2 - Paramètres du système.

Le SPP PARSYS fait appel au sous-programme d'édition des résultats PARRES.

---

PARRES

---

Appel.

```
call PARRES (IQ, IS, IT, IZ, CPAR, CSOL, GENR, CTYP, DONR, DONI,
            AL)
```

Paramètres.

IQ	] définis au paragraphe II.1
CPAR	
CSOL	
GENR	
CTYP	
DONR	
DONI	] définis au paragraphe II.2.2
IS	
IT	
IZ	
AL	

Déclaration de variables.Type caractère.

C130 : succession du symbole \* permettant de faire la distinction entre les différents paragraphes.

Description.

Ce sous-programme écrit les résultats fournis par le programme PARSYS dans le fichier.

Il imprime donc la façon dont sont choisis les paramètres (arbitrairement ou solution de «D+dN) ainsi que leurs valeurs.

II.3.2.3 - Modélisation du système.

Le SPP MODSYS appelle deux sous-programmes d'édition des résultats MODRES et MODREF.

Le premier écrit sur l'écran, le second dans un fichier. Nous les décrivons simultanément, leur structure étant la même. La seule différence apparaît lors de l'instruction d'écriture.

---

MODRES - MODREF

---

Appel.

call MODRES (IQ, IS, IT, IZ, TYPE, CPAR, CSOL, GENR, DONR, DONI,  
DONSOR, FTC, FTNC)

call MODREF (IQ, IS, IT, IZ, TYPE, CPAR, CSOL, GENR, DONR, DONI,  
DONSOR, FTC, FTNC)

Paramètres.

IQ	]      définis au paragraphe II.1
CPAR	
CSOL	
GENR	
DONR	
DONI	
DONSOR	

IS	] définis au paragraphe II.2.2
IT	
IZ	
TYPE	] définis au paragraphe II.2.3.
FTC	
FTNC	

#### Déclaration de variables.

Toutes les variables déclarées ont déjà été définies.

#### Description.

Le sous-programme MODRES affiche sur l'écran les résultats concernant le sous-programme primaire MODSYS. Il précise donc la forme de la matrice, les paramètres de modélisation, et la dernière colonne de la matrice. Le sous-programme écrit les mêmes résultats dans le fichier.

#### II.4 - Récapitulatif.

Les résultats exposés dans le paragraphe II sont récapitulés grâce au tableau de la figure V.8 sur lequel figurent les sous-programmes primaires et secondaires. La fonction de chacun des sous-programmes est résumée dans le tableau figure V.9. D'autre part, le déroulement du programme principal et des sous-programmes primaires est donné en annexe. On a utilisé pour cela un pseudo-langage accessible à tous. De plus, le listing du programme comporte de nombreux commentaires permettant, en cas d'éventuelle extension, de comprendre la signification de chacune des lignes.

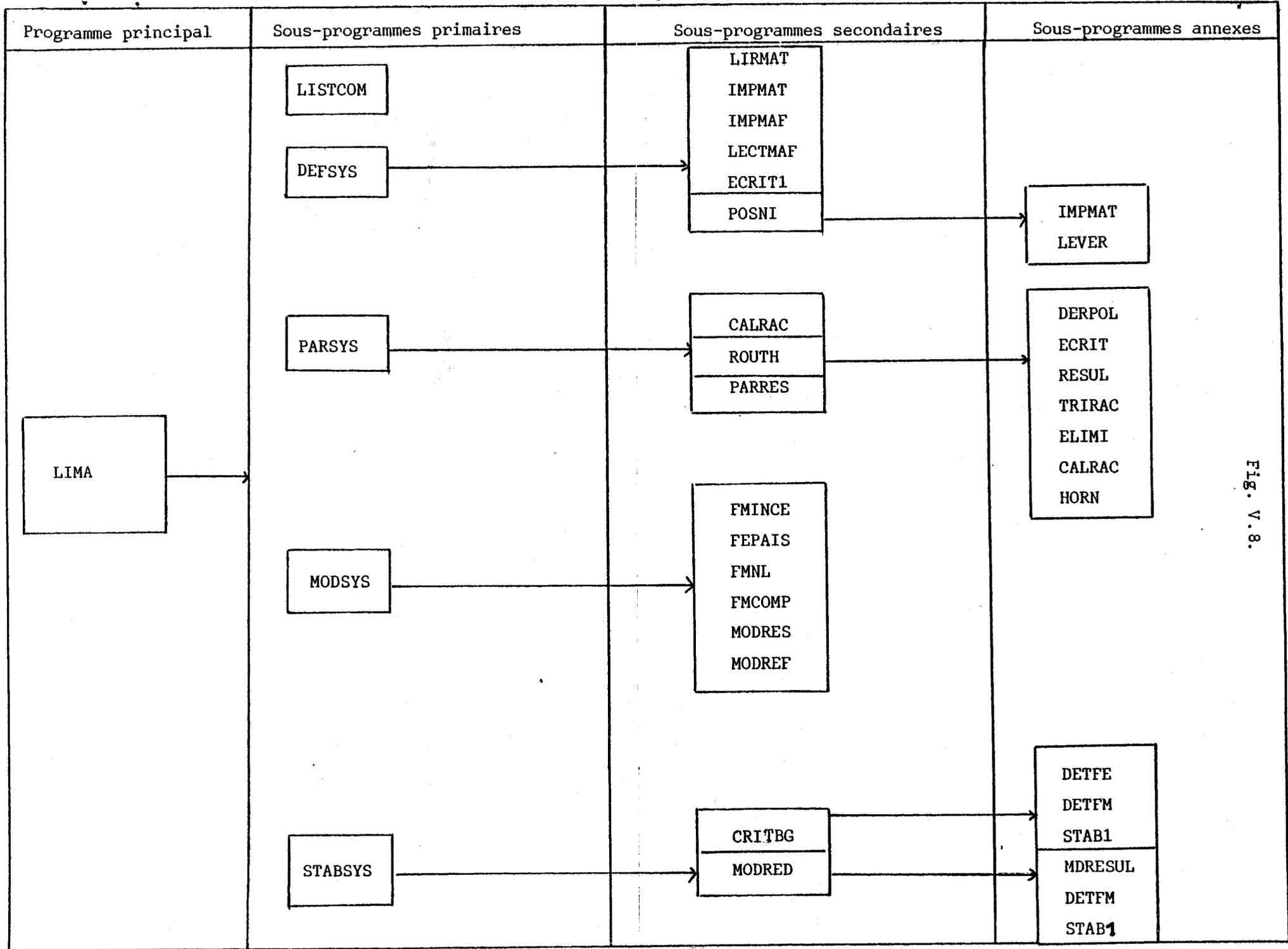


Fig. V.8.

CLASSE	NOM	ROLE
Programme principal	LIMA	Modélisation et étude de la stabilité des systèmes non linéaires de type mLP
Sous-programmes primaires	LISTCOM	Affichage des différentes commandes
	DEFSYS	Définition du système
	PARSYS	Détermination des paramètres de modélisation
	MODSYS	Modélisation du système
	STABSYS	Etude de la stabilité
Sous-programmes secondaires	LIRMAT	Lecture des coefficients de la matrice sur l'écran
	LECTMAF	Lecture des coefficients de la matrice dans le fichier
	IMPMAT	Ecriture de la matrice sur l'écran
	IMPMAF	Ecriture de la matrice dans le fichier
	ECRIT1	Ecriture des polynômes $N(p)$ $D(p)$ et $NI(p)$
	POSNI	Rangement des coefficients du polynôme caractéristique
	CALRAC	Calcul des racines d'une équation polynomiale pour la méthode de Bairstow
	ROUTH	Choix de $a$ tel que $D+aN = 0$ admette $q$ ou $q-1$ racines réelle négatives
	PARRES	Ecriture des résultats de PARSYS
	FMINCE	Calcul de la forme en flèche mince
	FEPAIS	Calcul de la forme en flèche épaisse
	FMNL	Calcul de la forme en flèche à plusieurs N.L.
	FMCOMP	Calcul de la forme en flèche mince complexe
	MODRES	Ecriture des résultats de MODSYS sur l'écran
	MODREF	Ecriture des résultats de MODREF dans le fichier
	CRITBG	Mise en oeuvre du critère de Borne et Gentina
	MODRED	Détermination d'un modèle réduit

Sous-programmes annexes	LEVER	Calcul du polynôme caractéristique d'une matrice par la méthode de Leverrier
	DERPOL	Dérivation formelle d'un polynôme
	ECRIT	Ecriture des deux premières lignes du tableau de Routh
	RESUL	Ecriture des autres lignes du tableau
	TRIRAC	Tri des racines
	ELIMI	Elimination des racines nulles
	HORN	Calcul d'un polynôme pour une valeur réelle
	DETFE	Calcul du déterminant d'une matrice pseudo-majorante d'une forme en flèche épaisse
	DETFM	Calcul du déterminant d'une matrice pseudo majorante d'une forme en flèche mince
	STAB1	Enoncé des conditions de stabilité
MODRESUL	Ecriture du modèle réduit	

Fig. V.9.

## III - EXEMPLES.

EXEMPLE : 1

=====

LE SYSTEME EST DEFINI PAR :

-

- Sa matrice d'evolution :

ORDRE DU SYSTEME : 3

NOMBRE DE NON-LINEARITES : 2

La matrice C(axa) des coefficients constants est :

-1.000000	0.0000000E+00	1.000000
0.0000000E+00	-2.000000	2.000000
1.000000	1.000000	-1.000000

La matrice CNL(axn) des coefficients des N-L est :

2.000000	1.000000
3.000000	2.000000
-1.000000	-1.000000

L'expression du polynome symbolique est de la forme:  $D(P)+f_1.N_1(P)+...+f_n.N_n(P)$   
ou  $n=2$  et

$$D(P) = P**3 + 0.1000000E+01*P**2 + -0.6000000E+01*P**1 + -0.7000000E+01*P**0$$

et

$$N_1(P) = 0.1000000E+01*P**2 + -0.2000000E+01*P**1 + -0.5000000E+01*P**0$$

$$N_2(P) = 0.1000000E+01*P**2 + -0.2000000E+01*P**0$$

\*\*\*\*\*

PARAMETRES DE MODELISATION

=====

Les parametres choisis sont arbitraires

On selectionne les parametres reels suivants

LAMBDA 1= -1.00000

LAMBDA 2= -2.00000

\*\*\*\*\*

## MODELISATION

=====

LA MATRICE EST EN FLECHE MINCE A PLUSIEURS N.L

## COLONNE DES PARAMETRES

-----

Lambda 1=-1.000

Lambda 2=-2.000

## COLONNE DES DELTAS(I) + GAMMAS(I) \* FK(\*)

-----

1.000	+ 2.000	* f 1(*)+ 1.000	* f 2(*)
1.000	+ 3.000	* f 1(*)+ 2.000	* f 2(*)
2.000	+ -1.000	* f 1(*)+ -1.000	* f 2(*)

\*\*\*\*\*

## STABILITE

=====

LA MATRICE COMPORTE PLUSIEURS N.L

Les criteres enonces ne peuvent etre utilises sur  
une telle matrice

EXEMPLE : 2

=====

\*\*\*\*\*

## DEFINITION DU SYSTEME

=====

LE SYSTEME EST DEFINI PAR :

\*

- Son schema bloc :

ORDRE DU SYSTEME : 3

$$D(P) = P^3 + 0.1130000E+02 * P^2 + 0.1321000E+02 * P + 0.1200000E+01$$

et

$$N(P) = 0.1000000E+01 * P^2 + 0.6000000E+01 * P + 0.5000000E+01$$

\*\*\*\*\*

## PARAMETRES DE MODELISATION

=====

Les parametres choisis sont arbitraires

On selectionne les parametres complexes et reels simples suivants

$$\text{LAMBDA 1} = -10.2400$$

$$(\text{MU 1}, \text{NU 1}) = (-1.00000, -0.100000E-01)$$

\*\*\*\*\*

## MODELISATION

=====

LA MATRICE EST EN FLECHE MINCE COMPLEXE

COLONNE DES PARAMETRES

COLONNE DES DELTAS(I) + GAMMAS(I) \* f

COLONNE DES PARAMETRES	COLONNE DES DELTAS(I) + GAMMAS(I) * f
Lambda 1 = -10.24	(-2.481, -.2685E-02) + f(*) ( 5.240, 0.5671E-
(Mu 1, Nu 1) = (-1.000, -.1000E-01)	(0.1852, -.6715E-02) + f(*) (0.6137E-05, 0.4329E-
	(-.6000E-01, 0.1000E-01) + f(*) (-1.000, 0.0000E+

PARAMETRES DE MODELISATION  
=====

Les parametres choisis sont arbitraires

On selectionne les parametres complexes et reels simples suivants

LAMBDA 1= -10.2400

(MU 1,NU 1) = ( -0.100000 , -0.100000 )

\*\*\*\*\*  
MODELISATION  
=====

LA MATRICE EST EN FLECHE MINCE COMPLEXE

COLONNE DES PARAMETRES  
-----

COLONNE DES DELTAS(I)+ GAMMAS(I) \* f(\*)  
-----

Lambda 1=-10.24	(-2.169	,-.2139E-01)+f(*)	( 4.774	,0.4709E-01)	
(Mu 1, Nu 1)=(-.1000	,-.1000	)			
	(0.9789E-02,0.1074	) + f(*)	(-.4344	,0.5291E-01)	
	(-.9600	,0.1000	) + f(*)	(-1.000	,0.0000E+00)

\*\*\*\*\*  
PARAMETRES DE MODELISATION  
=====

Les parametres choisis sont arbitraires

On selectionne les parametres complexes et reels simples suivants

LAMBDA 1= -10.2400

(MU 1,NU 1) = ( -1.00000 , -0.100000E-02)



\*\*\*\*\*  
MODELISATION  
=====

LA MATRICE EST EN FLECHE MINCE COMPLEXE

COLONNE DES PARAMETRES  
-----

COLONNE DES DELTAS(I)+ GAMMAS(I) \* f(\*)  
-----

Lambda 1=-10.24                    (-2.381        ,-.2577E-03)+f(\*) ( 5.240        ,0.5671E-03)  
 (Mu 1, Nu 1)=(-1.000            ,-.1000E-02)  
                                   (0.1753        ,-.6823E-03)+f(\*) (0.6137E-07,0.4329E-07)  
                                   (-.6000E-01,0.1000E-02)+f(\*) (-1.000        ,0.0000E+0)

\*\*\*\*\*

PARAMETRES DE MODELISATION

=====

Les parametres choisis sont arbitraires

On selectionne les parametres complexes et reels simples suivants

LAMBDA 1= -10.2400

(MU 1, NU 1) = ( -0.500000        , -0.100000E-01)

\*\*\*\*\*

MODELISATION

=====

LA MATRICE EST EN FLECHE MINCE COMPLEXE

COLONNE DES PARAMETRES	COLONNE DES DELTAS(I)+ GAMMAS(I) * f(*)		
-----	-----		
Lambda 1=-10.24	(-2.259	,-.2319E-02)+f(*) ( 4.971	,0.5104E-03)
(Mu 1, Nu 1)=(-.5000	,-.1000E-01)	(0.2732	,0.2919E-02)+f(*) (-.2310
	(-.5600	,0.1000E-01)+f(*) (-1.000	,0.0000E+0)

\*\*\*\*\*

EXEMPLE : 3

=====

\*\*\*\*\*

DEFINITION DU SYSTEME

=====

LE SYSTEME EST DEFINI PAR :

\*

- Son schema bloc :

ORDRE DU SYSTEME : 9

$$D(P) = P^{**9} + 0.3200000E+02 * P^{**8} + 0.4380000E+03 * P^{**7} + 0.3324000E+04 * P^{**6} + 0.1510500E+05 * P^{**5} + 0.4110000E+05 * P^{**4} + 0.6200000E+05 * P^{**3} + 0.4000000E+05 * P^{**2}$$

et

$$N(P) = 0.1000000E+01 * P^{**8} + 0.1600000E+02 * P^{**7} + 0.1100000E+03 * P^{**6} + 0.4260000E+03 * P^{**5} + 0.1023000E+04 * P^{**4} + 0.1574000E+04 * P^{**3} + 0.1530000E+04 * P^{**2} + 0.8640000E+03 * P^{**1} + 0.2160000E+03 * P^{**0}$$

\*\*\*\*\*

PARAMETRES DE MODELISATION

=====

Les parametres choisis sont arbitraires

On selectionne les parametres reels multiples, complexes simples suivants:

LAMBDA 1 = -1.00000 MULTIPLICITE = 1.00

LAMBDA 2 = -2.00000 MULTIPLICITE = 2.00

LAMBDA 3 = -3.00000 MULTIPLICITE = 3.00

(MU 1, NU 1) = ( -1.00000 , 1.00000 )

\*\*\*\*\*

MODELISATION

=====

LA MATRICE EST EN FLECHE EPAISSE REELLE

COLONNE DES PARAMETRES

COLONNE DES DELTAS(I) + GAMMAS(I) \* f(\*)

-----

-----

Lambda 1=-1.000	, 1.000	-864.0	+0.0000E+00* f(*
Lambda 2=-2.000	, 2.000	-216.0	+0.0000E+00* f(*
Lambda 2=-2.000	, 2.000	1296.	+0.0000E+00* f(*
Lambda 3=-3.000	, 3.000	297.1	+0.0000E+00* f(*
Lambda 3=-3.000	, 3.000	95.52	+0.0000E+00* f(*
Lambda 3=-3.000	, 3.000	14.40	+0.0000E+00* f(*
(Mu 1,Nu 1)=-1.000	, 1.000	-562.6	+0.0000E+00* f(*
		-132.4	+0.0000E+00* f(*
		-16.00	+ -1.000 * f(*

\*\*\*\*\*

STABILITE  
=====

Les termes non constants de la matrice sont quelconques  
Les criteres enonces ne peuvent etre utilises sur  
une telle matrice

\*\*\*\*\*

PARAMETRES DE MODELISATION  
=====

Les parametres sont choisis parmi les solutions de  $N(p) = 0$

On selectionne les parametres reels multiples, complexes simples suivants:

LAMBDA 1 =	-1.00000	MULTIPLICITE =	1.00
LAMBDA 2 =	-2.00000	MULTIPLICITE =	2.00
LAMBDA 3 =	-3.00000	MULTIPLICITE =	3.00
(MU 1,NU 1) =	( -1.00000	,	1.00000 )

\*\*\*\*\*

MODELISATION  
=====

LA MATRICE EST EN FLECHE EPAISSE REELLE

COLONNE DES PARAMETRES

COLONNE DES DELTAS(I)

Lambda 1=-1.000	, 1.000	-864.0	
Lambda 2=-2.000	, 2.000	-216.0	
Lambda 2=-2.000	, 2.000	1296.	
Lambda 3=-3.000	, 3.000	297.1	
Lambda 3=-3.000	, 3.000	95.52	
Lambda 3=-3.000	, 3.000	14.40	
(Mu 1,Nu 1)=(-1.000	, 1.000 )	-562.6	
		-132.4	
		-16.00	+ -1.000 * f(*)

\*\*\*\*\*  
STABILITE  
=====

ON NE PEUT CONCLURE DANS CE CAS  
LE CRITERE DE BORNE ET GENTINA N'EST PAS APPLICABLE

On a en effet  $\mu^2 - \text{abs}(\nu)^2 = 0$  et le signe du plus petit mineur n'est pas strictement positif

\*\*\*\*\*

DEFINITION DU SYSTEME

=====

LE SYSTEME EST DEFINI PAR :

\*

- Son schema bloc :

ORDRE DU SYSTEME : 4

$$D(P) = P^4 + 0.4000000E+01 P^3 + 0.4000000E+01 P^2$$

et

$$N(P) = 0.1000000E+01 P^3 + 0.2000000E+01 P^2 + 0.2000000E+01 P + 0.1000000E+01$$

\*\*\*\*\*

PARAMETRES DE MODELISATION

=====

Les parametres sont choisis parmi les solutions de  $N(P) = 0$

On selectionne les parametres complexes et reels simples suivants

$$\text{LAMBDA 1} = -1.00000$$

$$(\text{MU 1, NU 1}) = (-0.500000, -0.866000)$$

$$(\text{MU 2, NU 2}) = (-0.500000, 0.866000)$$

\*\*\*\*\*

MODELISATION

=====

LA MATRICE EST EN FLECHE MINCE COMPLEXE

COLONNE DES PARAMETRES

-----

COLONNE DES DELTAS(I)+ GAMMAS(I) \* f

-----

Lambda 1=-1.000	(-1.000	, 0.0000E+00)+f(*)	(0.0000E+00, 0.0000E+00)
(Mu 1, Nu 1)=(-.5000	, -.8660	)	
(Mu 2, Nu 2)=(-.5000	, 0.8660	)	
	( 1.500	, 0.8660	) + f(*) (0.0000E+00, -.2541E-
	( 1.500	, -.8660	) + f(*) (0.0000E+00, 0.2541E-
	(-2.000	, 0.0000E+00)+f(*)	(-1.000, 0.0000E+

\*\*\*\*\*

MODELISATION

=====

LA MATRICE EST EN FLECHE EPAISSE REELLE

COLONNE DES PARAMETRES

-----

COLONNE DES DELTAS(I)

-----

Lambda 1=-1.000 , 1.000  
 (Mu 1,Nu 1)=(-.5000 ,0.8660 )

-1.000  
 0.6340  
 2.366  
 -2.000      +-1.000      \* f(\*)

\*\*\*\*\*

\*\*\*\*\*

EXEMPLE : 5

=====

\*\*\*\*\*

DEFINITION DU SYSTEME

=====

LE SYSTEME EST DEFINI PAR :

\*

- Son schema bloc :

ORDRE DU SYSTEME : 4

$$D(p) = p^{**} 4 + 0.6000000E+01 * p^{**} 3 + 0.1100000E+02 * p^{**} 2 + 0.6000000E+01 * p^{**} 1$$

et

$$N(p) = 0.1000000E+01 * p^{**} 3 + 0.8000000E+01 * p^{**} 2 + 0.2000000E+02 * p^{**} 1 + 0.1700000E+02 * p^{**} 0$$

\*\*\*\*\*

PARAMETRES DE MODELISATION

=====

Les parametres sont choisis parmi les solutions de  $D(p) + a N(p) = 0$

Avec  $a = 0.10$

On selectionne les parametres reels suivants

LAMBDA 1= -0.420680

LAMBDA 2= -0.665080

LAMBDA 3= -2.04910

\*\*\*\*\*

MODELISATION

=====

LA MATRICE EST EN FLECHE MINCE REELLE

COLONNE DES PARAMETRES

COLONNE DES DELTAS(I)+ GAMMAS(I) \* f(\*)

COLONNE DES PARAMETRES	COLONNE DES DELTAS(I)+ GAMMAS(I) * f(*)		
Lambda 1=-.4207	2.494	+ -24.94	* f(*)
Lambda 2=-.6651	-2.053	+ 20.53	* f(*)
Lambda 3=-2.049	0.4453E-01	+ -.4458	* f(*)
	-2.865	+ -1.000	* f(*)

\*\*\*\*\*

## STABILITE

=====

Les termes de la dernière colonne sont proportionnels à  $s(*) = f(*) - a$   
Avec  $a = 0.10$

CAS  $f(*) - a > 0$ CONDITIONS A REMPLIR :  $f(*) > 0.10$  et  $\text{Det } M > 0$ 

-----

CONCLUSION :  $0.1000 < f(*) < 1.129$ CAS  $f(*) - a < 0$ CONDITIONS A REMPLIR :  $f(*) < 0.10$  et  $\text{Det } M > 0$ 

-----

CONCLUSION : LE CRITERE NE PERMET PAS DE CONCLURE

\*\*\*\*\*

## STABILITE

=====

Les termes de la dernière colonne sont proportionnels à  $s(*) = f(*) - a$

CAS  $f(*) - a > 0$ 

-----

La valeur du déterminant est:  $\text{Det } M = ( 6.824 ) + f(*) * (-51.24 )$ CONDITIONS A REMPLIR :  $f(*) > 0.10$  et  $\text{Det } M > 0$ 

-----

CONCLUSION :  $0.1000 < f(*) < 0.1332$

CAS  $f(*) - a < 0$

-----

La valeur du determinant est:  $\text{Det } M = (-3.539 \quad ) + f(*) * ( 52.39$

CONDITIONS A REMPLIR :  $f(*) < 0.10$  et  $\text{Det } M > 0$

-----

CONCLUSION :  $0.6755E-01 < f(*) < 0.1000$

\*\*\*\*\*

## CONCLUSION.

Dans ce chapitre sont détaillées la structure générale du programme et ses différentes possibilités.

Son caractère conversationnel permet à l'utilisateur de vérifier la bonne interprétation du texte frappé. Le dialogue questions-réponses est très efficace.

Sur le plan logiciel, le programme a été conçu de façon modulaire, pour lui permettre d'évoluer en fonction des besoins ultérieurs, par l'intermédiaire de directives.

D'autre part, la description des variables utilisées, les nombreux commentaires, la description des programmes en annexe facilitent toutes modifications éventuelles dans le programme.

Les extensions qui sont envisageables à ce jour sont :

- programmation de nouvelles formes matricielles
- utilisation du tableau de Routh pour un système d'ordre supérieur à 5
- étude de la stabilité pour toutes formes matricielles par application de nouveaux critères.

ANNEXE 1

---

PROGRAMME PRINCIPAL

---

DEBUT

faire

ECRIRE LE MENU

faittant que 1 (CONFIR = 'N') faire

LIRE LE NOM DE FICHER

si ( $\neq$  LXIST) alors

NEW = VRAI

sinon

NEW = FAUX

fin sifin tant que 1si (NEW) alors

CREER LE FICHER 'NOMFIC'

sinon

RELIRE LE FICHER 'NOMFIC'

fin sifaire

DIR = 'DEF'

faittant que 2 (DIR = 'DEF') faire

APPEL DEFSYS (NEW, IQ, CDEF, DPOS, CC, DPOS2, DPOS7, NBNL)

tant que 3 (DIR = 'PAR' ou DIR = 'AID' ou DIR = 'FIN') fairesi (DIR = 'AID') alors

APPEL LISTCOM

sinonsi (DIR = 'FIN') alors

ALLER A LA FIN

sinonAPPEL PARSYS (IQ, IS, IT, IZ, CDEF, GENR, CSOL, CTYP, CPAR, DPOS,  
DPOS2, DPOS5, DPOS6, DPOS7, DONI, CC, CC1, NBNL, AL)fin sifin sitant que 4 (DIR = 'MOD' ou DIR = 'AID' ou DIR = 'FIN') fairesi DIR = 'AID' alors

APPEL LISTCOM

```

sinon
  si (DIR = 'FIN') alors
    ALLER A LA FIN
  sinon
    APPEL MODSYS (IQ, NBNL, CC1, DPOS5, DPOS6, DONR, DONI, IS, IT, IZ,
      CDEF, DPOS7, DONSOR, GENR, CSOL, CTYP, CPAR)
  fin si
fin si
tant que 5 (DIR = 'STA' ou DIR = 'AID' ou DIR = 'FIN') faire
  si ('DIR' = 'AID') alors
    APPEL LISTCOM
  sinon
    si (DIR = 'FIN') alors
      ALLER A LA FIN
    sinon
      APPEL STABSYS (IQ, IS, IT, CPAR, GENR, CIMP, CSOL, DONR, DONI,
        DONSOR, AL, NBNL)
    fin si
  fin si
tant que 6 (DIR = 'AID' ou DIR = 'FIN') faire
  si (DIR = 'AID') alors
    APPEL LISTCOM
  sinon
    ALLER A LA FIN
  fin si
fin tant que 6
fin tant que 5
fin tant que 4
fin tant que 3
fin tant que 2
FIN

```

---

 SUBROUTINE DEFSYS
 

---

DEBUT :

```

si (NEW) alors {nouveau fichier}
  si (CDET = '2') alors {système défini par sa matrice}
    CHAR = ' '
    APPEL LIRMAT (MTC, IQ, IQ)
    APPEL IMPMAT (MTC, IQ, IQ)
    APPEL LIRMAT (MTCNL, IQ, NBNL)
    APPEL IMPMAT (MTCNL, IQ, NBNL)
    APPEL POSNI (IQ, CC, CNL, MTC, MTCNL, NBNL)
    si (CSTOC = ' ') alors {écriture des matrices dans fichier}
      APPEL IMPMAF (MTC, IQ, IQ)
      APPEL IMPMAF (MTCNL, IQ, NBNL)
    fin si
    si (NBNL ≠ 1) alors {système à plusieurs N.L.}
      POLSYM = 'D(p) + f1N1(p) + ... + fnNn(p)'
    sinon
      POLSYM = 'D(p) + fN(p)''
    fin si
    ECRIRE POLSYM
    APPEL ECRIT1 (IDIM, IQ, CC, DPOS, DPOS1, DG, M2, PI)
    M21= M2
    pour I = M2 à 1 pas - 1 {écriture du polynôme D(p) sur écran}
      ECRIRE IQ, DPOS(I), GG(I)
    fin pour
    si (CSTOC = ' ') alors {écriture du polynôme D(p) dans le fichier}
      ECRIRE POLSYM
      pour I = M21 à 1 pas -1
        ECRIRE IQ, DPOS1 (I), GG(I)
      fin pour
    fin si
    pour J = 1 à NBNL pas 1
      pour I = 1 à IQ pas 1
        DPOS 2(I) = CNL (I,J)
      fin pour

```

```

pour I = J*IQ + 1 à (J+1) IQ pas 1
      DPOS7 (I) = CNL ((J+1) * IQ - I + 1,J)
fin pour
APPEL ECRIT1 (IDIM, IQ, DPOS2, DPOS1, DG, GG, M2, PI)
si NBNL = 1 alors
  pour I = M2 à 1 pas -1
    ECRIRE DPOS1 (I), GG(I) {écriture du polynôme N(p) sur écran}
  fin pour
sinon
  pour I = M2 à 1 pas -1
    ECRIRE J, DPOS1 (I), GG(I) {écriture du polynôme NI(p) sur
                                écran}
  fin pour
fin si
M22 = M2
si (CSTOC = ' ') alors {écriture des polynômes dans fichier}
  ECRIRE M22
  si (NBNL = 1) alors
    pour I = M22 à 1 pas -1
      ECRIRE DPOS1 (I), GG(I)
    fin pour
  sinon
    pour I = M22 à 1 pas -1
      ECRIRE DPOS1 (I), GG(I)
    fin pour
  fin si
fin si
fin pour
sinon { système défini par son schéma bloc}
  CHAR = '*'
  NBNL = 1
  pour I = 1 à 2*IQ pas 1 { lecture des données }
    ECRIRE DPOS(I)
    LIRE DPOS(I)
  fin pour

```

```

POLSYM = 'D(p) + f.N(p)'
ECRIRE POLSYM
APPEL ECRIT1 (IDIM, IQ, DPOS, DPOS1, DG, GG, M2, PI)
M21 = M2
pour I = 1 à M21 pas 1
    ECRIRE IQ, DPOS1(I), DG(I) {écriture du polynôme D(p) sur écran}
fin pour
si (CSTOC = ' ') alors {écriture du polynôme D(p) dans fichier}
    pour J = 1 à M21 pas 1
        ECRIRE IQ, DPOS1(I), DG(I)
    fin pour
fin si
PI = IQ
pour I = IQ + 1 à 2*IQ pas 1
    DG(I) = 0
fin pour
APPEL ECRIT 1 (IDIM, IQ, DPOS, DPOS1, DG, GG, M2, PI)
M22 = M2
pour I = 1 à M22 pas 1
    ECRIRE DPOS1(I), DG(I) {écriture du polynôme N(p) sur écran}
fin pour
si (CSTOC = ' ') alors {écriture du polynôme N(p) dans fichier}
    pour I = 1 à M22 pas 1
        ECRIRE DPOS1(I), DG(I)
    fin pour
fin si
fin si
sinon {ancien fichier} {système défini par sa matrice, lecture des matrices
    et du polynôme symbolique dans fichier puis écriture
    sur écran}

si (CHAR = ' ') alors
    APPEL LECTMAF (MTC, IQ, IQ)
    APPEL IMPMAT (MTC, IQ, IQ)
    APPEL LECTMAF (MTCNL, IQ, NBNL)
    APPEL IMPMAT (MTCNL, IQ, NBNL)
    LIRE POLSYM
    pour I = M21 à 1 pas -1
        LIRE IQ, DPOS (I, GG(I) {lecture dans fichier de D(p)}
    fin pour
    pour I = M2 à 1 pas -1
        ECRIRE IQ, DPOS1(I), GG(I) {écriture sur écran de D(p)}
    fin pour

```

```

pour J=1 à NBNL pas 1
  M2=M22
  si (NBNL = 1) alors {système à une seule NL}
    pour I - M22 à 1 pas -1
      LIRE DPOS1(I), GG(I) { lecture de N(p) dans fichier }
    fin pour
    pour I = M2 à 1 pas - 1
      ECRIRE J1,DPOS1(I), GG(I) { écriture de N(p) sur écran }
    fin pour
  sinon { système à plusieurs NL }
    J1 = J
    pour I = M22 à 1 pas - 1
      LIRE J1, DPOS1(I), GG(I) { lecture des NI(p) dans fichier }
    fin pour
    pour I = M2 à 1 pas -1
      ECRIRE J1, DPOS1(I), GG(I) { écriture des NI(p) sur écran }
    fin pour
  fin si
fin pour
sinon { système défini par schéma bloc, lecture des polynômes N(p)
  N(p) et D(p) dans fichier et écriture sur écran }
  NBNL = 1
  M2 = M21
  pour I = 1 à M21 pas 1
    LIRE IQ, DPOS1(I), DG(I) { lecture de D(p) dans fichier }
    ECRIRE IQ, DPOS1 (I), DG(I) { écriture de D(p) sur écran }
  fin pour
  M2 = M22
  pour I = 1 à M22 pas 1
    LIRE DPOS1(I), DG(I) { lecture de N(p) dans fichier }
    ECRIRE IQ, DPOS1(I), DG(I) { écriture de D(p) sur écran }
  fin pour
fin si
fin si

```

FIN

---

 SUBROUTINE PARSYS
 

---

DEBUT

```

  si (NBNL  $\neq$  1) alors {système à plusieurs NL}
    LIRE REP
    si (REP = ' ') alors {paramètres solutions de  $D(p) = 0$  ou  $NI(p) = 0$ }
      LIRE MDN
      pour I = 1 à MDN+1 pas 1
        LIRE CPOL (I)
      fin pour
      APPEL CALRAC (IDIM, CPOL, MDN, ROOTR, ROOTI, IR, IC)
      pour I = 1 à IR pas 1
        ECRIRE ROOTR (I) {écriture des racines réelles}
      fin pour
      pour I = 1 à IC pas 1
        ECRIRE ROOTI (I) {écriture des racines complexes}
      fin pour
    fin si
    GENR = '1'
    CPAR = '1'
    ALLER A 150 {entrée des paramètres choisis}
  fin si
  LIRE CPAR
  si (CPAR = '2') alors {paramètres solutions de  $cD(p)+dN(p) = 0$ }
    LIRE CSOL
    si (CSOL = '1') alors {paramètres solutions de  $D(p) = 0$ }
      si (CDEF = '1') alors {système défini par schéma bloc}
        DPOS4(IQ+I) = 1
        pour I = 1 à IQ pas 1
          DPOS4(I) = DPOS(I)
        fin pour
        APPEL CALRAC
        pour I=1 à IR pas 1
          ECRIRE ROOTR (I) {écriture des racines réelles}
        fin pour
        pour I = 1 à IC pas 1
          ECRIRE ROOTI (I) {écriture des racines complexes}
        fin pour

```

sinon{système défini par sa matrice}

I = IQ

CC1 (I+1) = 1

pour I = 1 à IQ pas 1

CC1 (IQ-I+1) = CC(I)

fin pour

APPEL CALRAC (IDIM, CC1, IQ, ROOTR, ROOTI, IR, IC)

pour I = 1 à IR pas 1

    ECRIRE ROOTR (I) écriture des racines réelles

fin pour

pour I = 1 à IC pas 1

    ECRIRE ROOTI (I) {écriture des racines complexes }

fin pour

fin si

sinon

si (CSOL = '2') alors {paramètres solutions de  $N(p) = 0$  }

    ECRIRE MDN

si (CDEF = '1') alors {système défini par son schéma bloc}

pour I=1 à IQ pas 1

            DPOS3(I) = DPOS (I+IQ)

fin pour

    APPEL CALRAC (IDIM, DPOS3, MDN, ROOTR, ROOTI, IR, IC)

pour I = 1 à IR pas 1

        ECRIRE ROOTI (I) {écriture des racines réelles}

fin pour

pour I = 1 à IC pas 1

        ECRIRE ROOTI (I) {écriture des racines complexes}

fin pour

sinon {système défini par sa matrice}

pour I = 1 à IQ pas 1

        DPOS 5 (I) = DPOS 2 (IQ-I+1)

fin pour

    APPEL CALRAC (IDIM, DPOS5, MDN, ROOTR, ROOTI, IR, IC)

pour I = 1 à IR pas 1

        ECRIRE ROOTR (I) {écriture des racines réelles}

fin pour

pour I = 1 à IC pas 1

        ECRIRE ROOTI (I) {écriture des racines complexes }

fin pour

fin si

sinon

si (IQ ≤ 4) alors {ordre du système inférieur à 5 }

si (CDEF = '1') alors {système défini par son schéma bloc}

            APPEL ROUTH (IQ, DPOS, AL)



```

    sinon système défini par sa matrice
      pour I = 1 à IQ pas 1
        DPOS6 (IQ-I+1) = CC(I)
      fin pour

      pour I = 1 à IQ pas 1
        DPOS6(I+IQ) = DPOS2 (IQ-I+1)
      fin pour
      APPEL ROUTH (IQ, DPOS6, AL)
      fin si
      GENR = '1'
      ALLER A 170 {entrée des paramètres choisis}
    sinon
      CPAR = '1'
      ALLER A 2020 {on recommence le choix de CPAR}
      fin si
      fin si
      fin si
      fin si
      ECRIRE GENR
150 si (CDEF = '2') alors {système défini par sa matrice}
      pour I = 1 à IQ pas 1
        CC1(IQ-I+1) = CC(I)
      fin pour
      si (NBNL = 1) alors {système à une seule N.L}
        pour I = 1 à IQ pas 1
          DPOS5(I) = DPOS2(IQ-I+1)
        fin pour
        fin si
        CC1(IQ+1) = 1
      fin si
170 si (GENR = '1') alors {paramètres tous réels }
50   pour I = 1 à IQ-1 pas 1
      LIRE DONR(I) {lecture des paramètres}
      fin pour
      pour I = 1 à IQ-2 pas 1
        pour J = I+1 à IQ-1 pas 1
          DELTA = DONR(I) - DONR(J)
          si (DELTA = 0) alors { les paramètres sont égaux }
            ALLER A 50 { on entre de nouveau les paramètres }
          fin si
        fin pour
      fin pour

```

fin pour 201

IS = IQ-1

IT = 0

pour I = 1 à IS pas 1

DONR (I+IS) = 1

fin pour

sinon

si (GENR = '2') alors {paramètres complexes }

LIRE CTYP

sinon

CTYP = '1'

fin si

si CTYP = '1' alors {matrice en flèche épaisse réelle }

LIRE IS, IT

sinon {matrice en flèche mince complexe }

LIRE IS, IZ

fin si

si (IS ≠ 0) alors { il y a des paramètres réels }

54

pour J = 1 à 2\*IS pas 1

LIRE DONR (J)

fin pour

pour I1 = 1 à IS-1 pas 1

pour J1 = I1+1 à IS pas 1

DELTA = DONR(I1) - DONR(I1)

si (DELTA = 0) alors {les paramètres sont égaux }

ALLER A 54 { on entre de nouveau les paramètres réels }

fin si

fin pour

fin pour

fin si

si ( IT ≠ 0 ou IZ ≠ 0) alors { il y a des paramètres complexes }

56

si (CTYP = '1') alors {matrice en flèche épaisse réelle }

pour J = 1 à 3\*IT pas 1

LIRE DONI(J)

fin pour

sinon {matrice en flèche mince complexe }

pour J=1 à 2\*I2 pas 1

LIRE DONI(I)

fin pour

fin si

si (CTYP = '1') alors {matrice en flèche épaisse réelle }

pour I1 = 1 à IT pas 1

si (DONI(IT+I1) < 0) alors {partie imaginaire négative }

ALLER A 56 {on entre de nouveau les paramètres complexes }

```

        fin si
        fin pour
    fin si
    si (CTYP = '1') alors {matrice en flèche épaisse réelle}
        I2 = IT
    sinon
        I2 = IZ
    fin si
    pour I1 = 1 à I2-1 pas 1
        pour J1 = I1+1 à I2 pas 1
            FDELTA = CMPLX (DONI(I1), DONI(I2+I1))-CMPLX(DONI(J),
                DONI(IZ+J))
            si FDELTA =CMPLX(0,0) alors {les paramètres sont égaux}
                ALLER A 56 {on entre de nouveau les paramètres}
            fin si
        fin pour
    fin pour
    fin si
    INT=0
    pour I1 = 1 à IS pas 1
        INT = INT + DONR (IS+I1)
    fin pour
    si (CTYP = '1') alors {matrice en flèche épaisse réelle}
        I1 = INT + 2*IT+1
    sinon
        I1 = INT+IZ+1
    fin si
    si (I1 ≠ IQ) alors {nombre de paramètres incorrect}
        ECRIRE "ERREUR"
    fin si
fin si
LIRE ECR
si (ECR = ' ') alors {écriture des paramètres dans fichier}
    APPEL PARRES (IQ, IS, IT, IZ, CPAR, CSOL, GENR, CTYP, DONR, DONT
        AL)
fin si
FIN

```

---

 SUBROUTINE MODSYS
 

---

DEBUT :

```

  si (CTYP = '1') alors {matrice en flèche épaisse réelle}
    I2=IT
  sinon {matrice en flèche mince complexe}
    I2=IZ
  fin si
  si (CDEF = '2') alors {système défini par sa matrice}
    si (NBNL = 1) alors {système à une seule NL}
      pour I = 1 à IQ pas 1
        DPOST(I) = CC1(I)
      fin pour
      pour I = IQ+1 à 2*IQ pas 1
        DPOST(I) = DPOS5(I-IQ)
      fin pour
    sinon
      pour I = 1 à IQ pas 1
        DPOS7 (I) = CC1(I)
      fin pour
    fin si
  fin si
  si (GENR = '1') alors {paramètres tous réels}
    si (NBNL = 1) alors {système à une seule NL}
      TYPE = 'FLECHE MINCE REELLE'
      si (CDEF = '1') alors {système défini par son schéma bloc}
        APPEL FMINCE (IQ, DONR, DONSOR, DPOS)
      sinon
        si (CSOL = '3') alors {paramètres solutions de  $D(p) + aN(p) = 0$ }
          APPEL FMINCE (IQ, DONR, DONSOR, DPOS6 )
        sinon
          APPEL FMINCE (IQ, DONR, DONSOR, DPOST)
        fin si
      fin si
    sinon
      TYPE = 'FLECHE MINCE A PLUSIEURS N.L'
      APPEL FMNL (IQ, NBNL, DONR, DPOS7, DONSOR)
    fin si

```

sinon

si (GENR = '2') alors { paramètres complexes }

si (CTYP = '2') alors { matrice en flèche mince complexe }

TYPE = 'FLECHE MINCE COMPLEXE'

si (CDEF = '1') alors { système défini par son schéma bloc }

APPEL FMCOMP (IQ, IS, IZ, DONR, DONI, DPOS, FTC, FTNC)

sinon

APPEL FMCOMP (IQ, IS, IZ, DONR, DONI, DPOST, FTC, FTNC)

fin si

sinon { matrice en flèche épaisse réelle }

TYPE = 'FLECHE EPAISSE RELLE'

si (CDEF = '1') alors système défini par son schéma bloc

CALL FEPAIS (IQ, IS, IT, DONR, DONI, S1, S2, SIG, DONSOR, DPOS)

sinon

CALL FEPAIS (IQ, IS, IT, DONR, DONI, S1, S2, SIG, DONSOR, DPOST)

fin si

fin si

sinon

TYPE = 'FLECHE EPAISSE REELLE'

si (CDEF = '1') alors { système défini par son schéma bloc }

CALL FEPAIS (IQ, IS, IT, DONR, DONI, S1, S2, SIG, DONSOR, DPOS)

sinon

CALL FEPAIS (IQ, IS, IT, DONR, DONI, S1, S2, SIG, DONSOR, DPOST)

fin si

fin si

fin si

CALL MODRES (IQ, IS, IT, IZ, TYPE, CPAR, CSOL, GENR, DONR, DONI, DONSOR, FTC, FTNC)

LIRE ECR

si (ECR = ' ') alors { écriture des résultats dans le fichier }

CALL MODREF (IQ, IS, IT, IZ, TYPE, CPAR, CSOL, GENR, DONR, DONI, DONSOR,  
FTC, FTNC)

fin si

FIN

---

 SUBROUTINE STABSYS
 

---

DEBUT :

faire

lire ECR

faitsi (ECR = ' ') alors

ECRITURE DES RESULTATS DANS FICHER.

fin si

TES = FAUX

si (CPAR = '1') alors { paramètres arbitraires }si (NBNL  $\neq$  1) alors { système à une seule NL }

ECRIRE "LA MATRICE COMPORTE PLUSIEURS NL, LES CRITERES NE PEUVENT  
ETRE UTILISES"

fin sisi (CTYP = '2') alors { matrice en flèche mince complexe }

ECRIRE "LA MATRICE EST EN FLECHE MINCE COMPLEXE, LES CRITERES  
NE PEUVENT ETRE UTILISES"

sinonsi (NBNL = 1) alors { système à une seule NL }

ECRIRE "LES TERMES DE LA DERNIERE COLONNE SONT QUELCONQUES.  
LES CRITERES NE PEUVENT ETRE UTILISES"

fin sifin sisinonsi (GENR  $\neq$  '1') alors { paramètres non tous réels }si (CTYP = '2') alors { matrice en flèche mince complexe }

ECRIRE "LA MATRICE EST EN FLECHE MINCE COMPLEXE, LES CRITERES  
NE PEUVENT ETRE UTILISES"

sinonpour I = 1 à IT pas 1si ((DONI (I) + ABS (DONI(I+IT)))=0) alors

TES = VRAI

fin sifin poursi (TES) alors

ECRIRE "ON NE PEUT PAS CONCLURE"

ALLER A FIN

fin si

APPEL CRITBG (IQ, IS, IT, AL, DONR, DONSOR, CSOL, GENR, ECR)

```
    fin si  
sinon  
    ECRIRE "LA MATRICE EST EN FLECHE MINCE, DEUX CRITERES SONT APPLI-  
        APPLICABLES!"  
    LIRE CRIT {choix du critère }  
    si (CRIT = '1') alors { critère du modèle réduit }  
        APPEL MODRED (IQ, AL, DONR, DONSOR, CSOL, ECR)  
    sinon  
        APPEL CRITBG (IQ, IS, IT, AL, DONR, DONSOR, CSOL, GENR, ECR)  
    fin si  
fin si  
fin si  
FIN
```

---

 SUBROUTINE ROUTH
 

---

DEBUT :

faireEXPRESSION DES ELEMENTS DU TABLEAU

{ CALCUL DU DEGRE DES POLYNOMES}

{ EXPRESSION DE LA PREMIERE LIGNE}

{ EXPRESSION DE LA DEUXIEME LIGNE}

APPEL DERPOL (DPOS, DR, DEG+1, IDIM)

{ ECRITURE DES DEUX PREMIERES LIGNES}

APPEL ECRIT (IDIM, DEG, L, K, A, B, E, M1)

{ CALCUL DES COEFFICIENTS DE CHAQUE POLYNOME}

{ IMPRESSION DE LA PREMIERE COLONNE}

APPEL RESUL (IDIM, L, J, A, D, B, E, M1)

{ IMPRESSION DE LA DIAGONALE DU TABLEAU}

APPEL RESUL (IDIM, L, J, A, D, B, E, M1)

faitfaireCALCUL DU NOMBRE DE CHANGEMENTS DE SIGNES

{ CALCULS DES RACINES DE LA PREMIERE COLONNE}

APPEL CALRAC (IDIM, AI, EXP, POOTR, ROOTI, IR, IC)

{ RANGEMENT DES RACINES (première colonne)}

APPEL TRIRAC (M, SC)

APPEL ELIMI (SC, SCO, M)

{ CALCUL DES RACINES DE LA PREMIERE COLONNE}

APPEL CALRAC (IDIM, AI, EXP, ROUTr, ROOTI, IP, IC)

{ RANGEMENT DES RACINES (diagonale)}

APPEL TRIRAC (M, SC)

APPEL ELIMI (SD, SDO, MC)

{ RANGEMENT DE TOUTES LES RACINES (première colonne et diagonale)}

APPEL TRIRAC (M,S)

APPEL ELIMI (S, SO, M)

{ TABLEAU DE VARIATION (partie 1)}

APPEL HORN (VAL, X, AI, EXP)

{ CALCUL DU NOMBRE DE CHANGEMENTS DE SIGNES}

{ TABLEAU DE VARIATION (partie 2)}

APPEL HORN (VAL, X, AI, EXP)

{ CALCUL DU NOMBRE DE CHANGEMENTS DE SIGNES}

{ RECHERCHE DES INTERVALLES POSSIBLES pour le paramètre a }

{CALCUL DES PARAMETRES DE MODELISATION solutions de  $D+aN=0$ }

APPEL CALRAC (IDIM, CPOS, EXP, ROOTR, ROOTT, IR, IC)

fait

FIN

ANNEXE 2.

## NOTICE D'UTILISATION DU PROGRAMME LIMA

SUR

VAX 11/750.

---

APPEL DU PROGRAMME

---

1 - Commandes usuelles.

- Le déroulement du programme peut être stoppé par appui simultané sur les touches CTRL et Y.
- Pour se déconnecter, taper L0.
- Toute réponse est validée par appui sur la touche RETURN.
- Pour arrêter le défilement des résultats sur l'écran, taper NO SCROLL.  
Pour continuer taper de nouveau NO SCROLL.

2 - Démarche à suivre.

## Remarques préliminaires.

- Le dialogue utilisateur calculateur est présenté à l'aide des abréviations suivantes :

SY qui représente les questions ou réponses fournies par le calculateur,

R qui représente les réponses devant être fournies par l'utilisateur.

- Toute valeur entrée à l'intérieur du programme doit être confirmée ou non suivant le cas.

```
SY :   CONFIRMEZ PAR ' ' OU 'N'
      ===>
```

le symbole correspond à un espace et ne doit pas être oublié.

### Appel du programme.

Allumer une console.

Dès que le pointeur apparaît sur l'écran, taper RETURN

```
SY : USERNAME
R   : taper LIMA
SY : PASSWORD
R   : taper AUTO
SY : le vax vous souhaite la bienvenue
      §
R   : RUN   LIMA
```

---

## PRESENTATION DU PROGRAMME

---

### 1. Affichage du menu.

Le menu du programme s'affiche sur l'écran.

### 2. Création d'un fichier.

```
SY :   NOM DE FICHIER-SYSTEME ENTRE ' ', PUIS RETURN
      ===>
```

```
R   : Entrer un nom de 1 à 9 caractères entre apostrophes
      ex : 'SY1'
```

Si le fichier a déjà été créé :

SY : ANCIEN FICHER ?

CONFIRMEZ PAR ' ' OU 'N'  
 ==>

Deux réponses sont possibles :

1. R : 'N'

Le système demande de nouveau un nom de fichier.

2. R : ' '

Le contenu du fichier s'affiche sur l'écran. Il s'agit des données définissant un système déjà étudié c'est à dire

- soit son polynôme symbolique
- soit sa matrice d'évolution puis son polynôme symbolique.

Si le fichier est nouveau :

SY : NOUVEAU FICHER ?

CONFIRMEZ PAR ' ' OU 'N'  
 ==>

Deux réponses sont possibles :

1. R : 'N'

Retour à la première question.

2. R : ' '

Un nouveau fichier est créé.

---

## DEFINITION DU SYSTEME

---

### 1. Entrée des données.

SY : LE SYSTEME EST DEFINI PAR:

1. Son schéma bloc
2. Sa matrice d'évolution

R : '1' ou '2'

Si l'utilisateur répond '1'

SY : Entrer Q , ordre du systeme ==>

R : Entrer un nombre compris entre 1 et 20

SY : Entrer les coefficients A(i) de D(p), puis B(i) de N(p)  
comme suit : A0, ..., Aq-1, B0, ..., Bq-1 ==>

R : Entrer les coefficients des polynômes D(p) puis N(p) en les  
séparant par une virgule.

Taper RETURN lorsque tous les coefficients sont entrés.

Si le programme ne se déroule plus, la liste des  
coefficients est incomplète.

Si l'utilisateur répond '2'.

SY : Entrer Q , ordre du systeme ==>

R : Entrer un nombre compris entre 1 et 20.

SY : Entrer N , nombre de non\_linearites ==>

R : Entrer un nombre compris entre 1 et 20

SY : SAISIE DES COEFFICIENTS DE LA MATRICE C (qxq) LIGNE/LIGNE  
Introduire l'element 1, 1 de la Matrice :  
Introduire l'element 1, 2 de la Matrice :  
Introduire l'element 1, 3 de la Matrice :

SAISIE DES COEFFICIENTS DE LA MATRICE CNL (qxN) LIGNE/LIGNE  
Introduire l'element 1, 1 de la Matrice :  
Introduire l'element 1, 2 de la Matrice :  
Introduire l'element 2, 1 de la Matrice :

R : Entrer les coefficients de la matrice ligne par ligne.

Appuyer sur RETURN après chaque coefficient.

Remarque. Chaque colonne de la matrice CNL contient  
les coefficients d'une même non linéarité  $f_i^*$ .

## 2. Résultats.

L'expression du polynôme symbolique s'affiche à l'écran.

---

 PARAMETRES DU SYSTEME
 

---

SY : CHOIX DES PARAMETRES

1.  $\lambda(i)$  choisis arbitrairement
2.  $\lambda(i)$  solutions de  $cD(\lambda) + dN(\lambda) = 0$

R : '1' ou '2'

Si l'utilisateur tape '1'

SY : SAISIE DES PARAMETRES LAMBDA(I)

1.  $\lambda(i)$  tous réels
2.  $\lambda(i)$  complexes simples  
complexes simples & réels simples
3.  $\lambda(i)$  réels multiples  
réels multiples & complexes simples

R : '1', '2' ou '3'

'1' : paramètres réels

 SY : ENTRER les  $\lambda(i)$  ( $i=1, q-1$ ), séparés par une virgule  
 ==>

 R : Entrer  $q-1$  paramètres  $\lambda(i)$  séparés par une virgule  
 puis taper RETURN.

'2' : paramètres complexes, complexes et réels simples.

 SY : ENTRER dans l'ordre le nombre de réels , de complexes  
 séparés par une virgule

 R : Entrer le nombre de réels, le nombre de complexes (la somme  
 de ces deux nombres étant inférieure à 20)

 SY : ENTRER la valeur des  $\lambda(i)$  réels puis leur ordre  
 respectif, séparés par une virgule

 R : Entrer les coefficients réels séparés par une virgule puis  
 leur multiplicité respectivement séparée par une virgule (dans  
 ce cas, les multiplicités valent 1)

SY : ENTRER les parties réelles des  $\lambda(i)$ , les parties imaginaires puis leur ordre respectif, séparés par une virgule

R : Entrer les parties réelles des complexes puis leurs parties imaginaires puis leur multiplicité.

Les multiplicités valent toutes 1.

Remarque. Dans le cas d'une flèche épaisse, les parties imaginaires sont choisies positives. Si cette condition n'est pas respectée, on recommence.

'3' : paramètres multiples.

Question et réponse ont le même format que dans le cas précédent.

Remarque. Un réel multiple de multiplicité  $n$  compte pour un seul réel.

Remarque.

Si la condition sur le nombre total de paramètres ( $\sum_{i=1}^s n_i + 2t + 1 = q$ ) n'est pas respectée, le message d'erreur suivant s'affiche

ERREUR, NOMBRE DE PARAMETRES INCORRECT

et on recommence l'entrée des paramètres.

Si l'utilisateur tape '2'.

3 cas sont envisageables

SY :

1.  $\lambda(i)$  solution de  $D(P)=0$  ( $c=0$ )
2.  $\lambda(i)$  solution de  $N(P)=0$  ( $d=0$ )
3.  $\lambda(i)$  solution de  $D(P)+aN(P)=0$  ( $a=d$ )

R : '1', '2' ou '3'.

'1' : paramètres solutions de  $D(p) = 0$ .

Le programme fournit les solutions de l'équation dans  $\mathbb{C}$ . L'utilisateur relève les valeurs inscrites sur l'écran.

'2' : paramètres solutions de  $N(p) = 0$ .

SY : ENTRER LE DEGRE DU POLYNÔME :

R : entrer le degré de  $N(p)$ .

Dans le cas d'un système à plusieurs NL il faut entrer le degré puis les coefficients du polynôme  $NI(p)$  choisi.

'3' : paramètres solutions de  $D(p) + aN(p) = 0$ .

Dans ce cas, les paramètres  $\lambda(i)$  sont tous réels. Seules les solutions de l'équation dans  $\mathbb{R}$  sont fournies.

L'utilisateur relève ces solutions.

Si le degré du polynôme  $D+aN$  est supérieur à 4, le tableau de Routh ne peut être utilisé et le programme renvoie l'utilisateur au début du module PARSYS.

Lorsque les solutions des équations  $D(p) = 0$ ,  $N(p) = 0$  ou  $D(p) + aN(p) = 0$  sont relevées, l'utilisateur peut entrer les valeurs suivant la démarche décrite dans le cas de paramètres arbitraires.

---

## MODELISATION DU SYSTEME

---

Dans ce module, aucune question n'est posée à l'utilisateur.

Les résultats suivants lui sont fournis :

- forme de la forme en flèche,
- coefficients de modélisation choisis,
- dernière colonne de la matrice.

LA MATRICE EST EN FLECHE MINCE REELLE

COLONNE DES PARAMETRES

---

Lambda 1=-.4207  
 Lambda 2=-.6651  
 Lambda 3=-2.049

COLONNE DES DELTAS(I)+ GAMMAS(I) \* f(\*)

---

2.494 +-24.94 \* f(\*)  
 -2.053 + 20.53 \* f(\*)  
 0.4453E-01+-.4458 \* f(\*)  
 -2.865 +-1.000 \* f(\*)

coefficients  $\lambda_i$

coefficients  $\delta_i + \gamma_i \cdot f^*$

coefficients  $\lambda_i$  réels multiples  
 de multiplicité 1, 2 et 3



LA MATRICE EST EN FLECHE EPAISSE REELLE

COLONNE DES PARAMETRES

---

Lambda 1=-1.000 , 1.000  
 Lambda 2=-2.000 , 2.000  
 Lambda 2=-2.000 , 2.000  
 Lambda 3=-3.000 , 3.000  
 Lambda 3=-3.000 , 3.000  
 Lambda 3=-3.000 , 3.000  
 (Mu 1, Nu 1)=(-1.000 , 1.000 )

COLONNE DES DELTAS(I)

---

-864.0  
 -216.0  
 1296.  
 297.1  
 95.52  
 14.40  
 -562.6  
 -132.4  
 -16.00 +-1.000 \* f(\*)

LA MATRICE EST EN FLECHE MINCE COMPLEXE

COLONNE DES PARAMETRES

COLONNE DES DELTAS(I) + GAMMAS(I) \* f(\*)

lambda 1=-10.24 (-2.381 ,-.2577E-03)+f(\*) ( 5.240 ,0.5671E-03)  
 Nu 1=(-1.000 ,-.1000E-02)  
 (0.1753 ,-.6823E-03)+f(\*) (0.6137E-07,0.4329E-03)  
 (-.6000E-01,0.1000E-02)+f(\*) (-1.000 ,0.0000E+00)

coefficients complexes  $\mu_1 + i\nu_1$

composantes complexes  $\delta_i + f^* \cdot \gamma_i$  du vecteur  $\alpha_i$   
 composantes complexes  $\delta'_j + f^* \gamma'_j$  du vecteur  $\beta_j$

avec  $\delta_i = -2.381 + i \cdot -.2577E-03$

$\gamma_i = 5.240 + i \cdot 0.5671E-03$



avec  $\delta'_j = 0.1753 + i \cdot -.6823E-03$

$\gamma'_j = .6137E-07 + i \cdot 0.4329E-03$

LA MATRICE EST EN FLECHE MINCE A PLUSIEURS N.L

COLONNE DES PARAMETRES

Lambda 1=-1.000  
 Lambda 2=-2.000

coefficients  $\delta_i + \sum_{k=1}^3 \gamma_k \cdot f_k^*$

COLONNE DES DELTAS(I) + GAMMAS(I) \* FK(\*)

1.000 + 2.000 \* f 1(\*)+ 1.000 \* f 2(\*)

1.000 + 3.000 \* f 1(\*)+ 2.000 \* f 2(\*)

2.000 +-1.000 \* f 1(\*)+-1.000 \* f 2(\*)

---

## STABILITE DU SYSTEME

---

Lorsque les paramètres ont été choisis arbitrairement, on ne peut pas conclure.

Lorsqu'ils sont solutions de  $D(p) = 0$ ,  $N(p) = 0$  ou  $D(p) + aN(p) = 0$ .

- si la matrice est en flèche mince, deux critères sont proposés.
- si la matrice est en flèche épaisse, seul le critère de Borne et Gentina peut être essayé.

Quel que soit le critère mis en oeuvre l'utilisateur choisit le signe de  $f^*$ -a ou  $f^*$ .

Remarques.

Critère du modèle réduit.

Si le modèle réduit est d'ordre 1, on ne peut appliquer le critère.

Critère de Borne et Gentina.

Si la différence  $\mu_i^2 - \nu_i^2$  est nulle, le critère n'est pas applicable.

---

## EDITION DES RESULTATS

---

- Pour obtenir la liste des fichiers créés, taper : DIR.
- Pour sortir sur imprimante les résultats qui s'affichent sur l'écran, taper : COPY NOM DE FICHER.DAT TTA3 :
- Pour relire sur l'écran, les résultats concernant un ancien fichier taper:EDT NOM DE FICHER.DAT.

- Pour quitter ce fichier taper simultanément CTRL et Z puis, une fois le symbole \* affiché, taper QUIT (ou EX si une modification est faite à l'intérieur de ce fichier).

Pour détruire un fichier, taper : DEL NOM DE FICHIER.DAT;numéro de version.

CONCLUSION GÉNÉRALE.

Le logiciel LIMA, implanté sur VAX 11/750 à l'I. D. N. permet actuellement d'étudier la modélisation et la stabilité des systèmes non linéaires monovariables.

Les résultats qu'il fournit sont les suivants :

- d'une part, il calcule le polynôme symbolique de tout système à non linéarités de rang un,

- d'autre part, il conduit rapidement à toute représentation du système sous forme en flèche,

- enfin, il réalise l'application de critères usuels de stabilité à partir de cette forme de description.

Ce logiciel permettra aux chercheurs de travailler sur des systèmes de plus grandes dimensions, de diminuer le temps de calcul et de faciliter la comparaison entre les formes de modélisation, en liaison avec les critères de stabilité existants, et les nouveaux résultats établis.

Il pourra également être utilisé dans un but pédagogique sans demander à l'étudiant d'aptitude particulière en informatique.

Par son aspect modulaire, le programme présenté peut évoluer avec l'insertion de nouveaux modules d'application. Une extension envisageable permettrait la réalisation d'un programme d'aide à la synthèse des systèmes non linéaires. Les modules d'analyse serviront alors de sous-programmes pour la partie synthèse. L'aspect interactif pourra alors être remplacé par une optimisation de conditions de stabilité ou de rapidité par rapport à toute une gamme de modèles et de critères pré-établis.

## BIBLIOGRAPHIE

## BIBLIOGRAPHIE.

/BARRAUD 1975/

*"Réalisation minimale et approximation optimale des systèmes dynamiques linéaires invariants"*.

Thèse de Doctorat d'Etat, Nantes 1975.

/BENREJEB 1976/

*"Sur la synchronisation des systèmes continus non linéaires en régime forcé"*.

Thèse de Docteur Ingénieur - Lille 1976.

/BENREJEB 1980/

*"Sur l'analyse et la synthèse des processus complexes hiérarchisés. Application aux systèmes singulièrement perturbés"*.

Thèse de Doctorat d'Etat - Lille 1980.

/BENREJEB, DAUPHIN, BORNE 1980/

*"Sur une nouvelle approche de la modélisation et de la simulation des processus non linéaires"*.

Congrès Simulation 80, Interlaken - Juin 1980.

/BENREJEB, BORNE, LAURENT 1982/

*"Sur une application de la représentation en flèche en l'analyse des processus"*.

Revue RAIRO Automatique 16 n° 2, p. 133-146 - 1982.

/BORNE 1976/

*"Contribution à l'étude des systèmes discrets non linéaires de grande dimension. Application aux systèmes interconnectés"*.

Thèse de Doctorat d'Etat, Lille 1976.

/BORNE, GENTINA 1972/

*"Sur une extension des conditions du linéaire de stabilité à certaines classes de systèmes échantillonnés non linéaires"*.

CRAS Paris, t274, p. 1275-1277 - Avril 1972.

/BORNE, GENTINA, LAURENT 1972/

*"Sur la stabilité des systèmes échantillonnés non linéaires"*.

Revue RAIRO Automatique, n° 52, p. 96-105 - 1972.

/BREZINSKI/

*"Algorithmique numérique"*.

Publications du Laboratoire de Calcul de l'Université des Sciences et Techniques de Lille".

/BRUNOVSKY 1970/

*"A classification of linear controllable systems"*.

Kybernetika 6, p. 173-188, 1970.

*/DAUPHIN-TANGUY 1983/*

*"Sur la représentation multi-modèle des systèmes singulièrement perturbés. Application à l'analyse et à la synthèse".*  
Thèse d'Etat - Lille 1983.

*/DEIF 1982/*

*"Advanced matrix theory for Scientists and Engineers".*  
Abacus Press - Halsted Press - 1982.

*/DESCUSSE 1981/*

*"Les invariants fondamentaux d'une paire (A,B), leurs liens avec certaines formes canoniques".*  
Outils et modèles mathématiques pour l'automatique, l'analyse de systèmes et le traitement du signal.  
Editions du CNRS, Volume 1 - 1981.

*/DIMSTER, ROTELLA, RICHARD 1984/*

*"Interactive software for analysis of non linear systems".*  
Actes AI'84, Innsbruck, Février 1984 - p. 96-103.

*/DURAND 1960/*

*"Solutions numériques des équations algébriques".*  
Tome 1 - Masson et Cie, édition 1960.

*/EL MOUDNI 1981/*

*"Introduction de nouveaux outils mathématiques pour la description des systèmes discrets non linéaires de grande dimension".*  
Thèse d'Etat - Lille 1981.

*/EVANS 1948/*

*"Graphical analysis of control systems".*  
Transactions AIEEE 67 : 547-551 - 1948.

*/FADEEV, FADEEVA 1963/*

*"Computational methods of linear algebra".*  
Freeman - San Francisco - 1963.

*/FOSSARD 1972/*

*"Commande des systèmes multidimensionnels".*  
Dunod, Paris 1972.

*/FOSSARD 1982/*

*"Modélisation et analyse des systèmes à échelles de temps multiples".*  
Cours ENSAE - Dept Automatique - 1982.

*/GANTMACHER 1966/*

*"Théorie des matrices".*  
Dunod 1966.

/GENTINA 1976/

*"Contribution à l'analyse et à la synthèse des systèmes continus non linéaires de grande dimensions"*.

Thèse d'Etat - Lille 1976.

/GENTINA, BORNE 1972/

*"Sur une condition d'application du critère de stabilité linéaire à certaines classes de systèmes continus non linéaires"*.

CRAS. Paris, t. 275, 16 août 1972.

/GENTINA, BORNE 1972/

*"Stabilité des systèmes continus non linéaires de grande dimension"*.

Revue RAIRO J3, p. 69-77, Aout 1972.

/GROUMPOS, SCOTT 1977/

*"The nested bordered diagonal method for symmetric eigenproblems"*.

Proc. 15 th Annual Conference on Communication Control and Computing - 1977.

/GRUJIC 1975/

*"Novel development of Ljapunov stability of motion"*.

Int. J. Control, vol. 22, n°4, p. 525-549, 1975.

/GRUJIC 1978/

*"Solution for the Lurie Postnikov and Aizerman problems"*.

Int. J. Syst. Sci., Vol. 9, n° 12, p. 1359-1372.

/GRUJIC, GENTINA, BORNE 1976/

*"General aggregation of large scale systems by vector Ljapunov functions and vector norms"*.

Int. J. Control, vol. 24, n° 4, p. 529-550 - 1976.

/HAHN 1963/

*"Theory and application of Ljapunov's direct method"*.

Prentice Hall, Englewood Cliffs, 1963.

/HAHN 1967/

*"Stability of motion"*.

Springer Verlag, 1967.

/HICKIN, SINHA 1977/

*"Transformation of linear multivariable systems to canonical forms"*.

Proceeding 15th Annual Allerton Conference on Communication Control and Computing 1977.

/HINRICHSSEN 1984/

*"Canonical forms and parametrization problems in linear systems theory (preliminary version)"*.

4th IMA Conference on Control Theory, p. 192 - 227, Cambridge, September 1984.

/KALMAN 1963/

*"Ljapunov functions for the problem of Lurie in automatic control"*.

Rapport Acad. Sc. USA, 49, p. 201-205.

/KRASOVSKII 1983/

"Some problems in the theory of stability of motion".  
Stanford Univ. Press, 1963.

/LASALLE, LEFSHETZ 1961/

"Stability by Ljapunov's direct method with applications".  
Academic Press, 1961.

/LAURENT 1968/

"Contribution à l'étude des systèmes échantillonnés non linéaires en régime dynamique".  
Thèse d'Etat, Lille 1968.

/LAURENT, PETLE, NGUYEN QUY HUNG 1970/

"Critère de stabilité des systèmes continus et échantillonnés".  
CRAS t. 270, p. 1637-1640, Juin 1970.

/LAURENT, EL MOUDNI, RICHARD, BORNE 1979/

"On initial stability conditions for non linear large-scale systems".  
4ème Congrès Informatica y Automatica, Madrid, 1979.

/LEFSCHETZ 1965/

"Stability of non linear control systems".  
Academic Press, New York, 1965.

/LEPELLETIER 1982/

"Techniques numériques appliquées au calcul scientifique".  
Masson 1982.

/LIGNELET 1982/

"Fortran 77 - Langage Fortran V.  
Masson 1982.

/LUENBERGER 1967/

"Canonical forms for linear multivariable systems".  
IEEE Trans. AC 12, p. 290-293 - 1967.

/LUR'E 1951/

"On some nonlinear problems in the theory of automatic control".  
Her majesty's stationery office, Londres 1951.

/LUR'E 1957/

"Some nonlinear problems in the theory of automatic control".  
Her majesty's stationery office, Londres 1957.

/LUR'E, POSTNIKOV 1944/

"Concerning the stability of regulating systems".  
Prilel Mat i Melch, Moscou 8, p. 246-248 - 1944.

/LJAPUNOV 1892/

"Problème général de la stabilité du mouvement".  
Article original (en russe) dans Comm. Soc. Math., Kharkow ; traduit  
en français en 1907 (Annales Fac, Toulouse, 9, p. 203-474),  
republié en anglais en 1949 (Annales Math. Studies, Princeton, vol 17)

/MARKUS 1955/

Mathemat. Zeits, n° 62, p. 310.

/PAILLET 1981/

"PAAS, Programme d'Aide à l'Analyse des Systèmes".

1st International Conference Applied Modelling and Simulation, p. 231-233, Lyon 1981.

/PAN, CHAO 1978/

"A Computer Aided Root - Locus Method"

IEEE Trans. on Automatic Control, vol AC 23, n° 5, p. 856-860, Octobre 1978.

/POPOV 1973/

"L'hyperstabilité des systèmes automatiques".

Dunod 1973.

/PRATZEL-WOLTERS 1983/

"Canonical forms for Linear Systems".

Linear Algebra and its applications n° 50, pp. 437-473, 1983.

/RICHARD 1981/

"Sur la mise en équation d'état de systèmes continus non linéaires par une méthode de calcul symbolique. Définition d'un invariant de représentation".

Thèse de Docteur Ingénieur - Lille 1981.

/RICHARD 1984/

"Définition d'invariants polynomiaux adaptés à la modélisation et à l'étude des systèmes non linéaires et non stationnaires".

Thèse d'Etat - Lille 1984.

/ROSENBROCK 1965/

"A method of investigating stability".

IFAC, Bâle 352/1, 1965.

/ROSENBROCK 1970/

"State Space and Multivariable theory".

Studies in dynamical systems - Nelson - 1970.

/ROSENBROCK, STOREY 1970/

"Mathematics of dynamical Systems".

Nelson 1970.

/ROTELLA 1983/

"Détermination de nouvelles représentations d'état adoptées à l'analyse et à la synthèse des systèmes continus non linéaires".

Thèse de Docteur Ingénieur - Lille 1983.

/ROTELLA, RICHARD 1983/

"Modélisation et synthèse de systèmes monovariabiles de type Lur'e Postnikov".

1st IASTED Symposium Applied Informatics. Vol 3, p. 173-176 - Lille 1983.

/ROTELLA, ZAMBETTAKIS, RICHARD 1982/

*"Application de la méthode de lieu des racines à la synthèse de systèmes non linéaires."*

Actes MECO'83, p. 271-275, Tunis, 1982.

/TITLI 1975/

*"Commande hiérarchisée et optimisation des processus complexes".*

Dunod Automatique, 1975.

/YAKUBOVITCH 1962/

*"La solution de quelques inégalités matricielles rencontrées dans la théorie du réglage automatique".*

(En Russe), Doklady, A.N. SSSR, t. 143, n° 6.

/ZAMBETTAKIS 1983/

*"Contribution à l'étude des systèmes à non-linéarités multiples. Application aux systèmes électromécaniques".*

Thèse de Docteur Ingénieur - Lille 1983.

/ZAMBETTAKIS, RICHARD, ROTELLA 1984/

*"Simplification of models for stability analysis of large scale systems"*

Chapitre 6 du livre "Multivariable control, new concepts and tools".

Part 1, Edit Tzafestas, North Holland 1984.

Notices VAX 11 FORTRAN.

Language Reference Manual.

User's guide.

1980. Digital Equipment Corporation.



## R E S U M E

L'étude des systèmes non linéaires pose encore plus que dans le cas linéaire, le problème du choix de la représentation.

En effet, les résultats obtenus diffèrent selon qu'un même critère (de stabilité, par exemple) est appliqué à une modélisation plutôt qu'à une autre.

D'autre part, sur une même modélisation, deux critères différents peuvent conduire à des conditions de stabilité différentes.

Il n'existe pas, à ce jour, de méthode générale déterminant "le meilleur modèle" à choisir afin d'obtenir une synthèse optimale d'un processus. Il est donc nécessaire de définir plusieurs modèles du système et d'essayer sur chacun les critères existants.

Pour les systèmes d'ordre élevé, une telle démarche nécessite alors l'utilisation d'un programme fournissant les différents modèles de représentation et les conditions de stabilité associées au modèle choisi.

Ce programme, appelé LIMA (Logiciel Interactif de Modélisation et d'Analyse), a été implanté sur VAX 11/750.

Les résultats fournis sont les suivants :

- il calcule le polynôme symbolique de tout système à non linéarités de rang 1,
- il conduit rapidement à toute représentation matricielle du système de type forme en flèche,
- il réalise l'application de critères usuels de stabilité, à partir de cette représentation.

### MOTS-CLES

Conception assistée  
Espace d'état  
Logiciel  
Polynôme caractéristique  
Stabilité  
*Modélisation assistée*

Equation d'état  
Fonction de Ljapunov  
Modélisation  
Programme enseignement  
Système non linéaire