

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التربية الوطنية
MINISTRE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : Electronique

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Projet de Fin d'Etudes

SUJET

Simplification des Modèles Complexes par la Méthode des Réalisations Equilibrées

Proposé par :
- Mr F. CHIGARA

Etudié par :
- Melle A.B.H ADAMOU

Dirigé par :
- Mr F. CHIGARA

Promotion
Juin 1993

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

*A Maman
avec tous mes Amours*

Amel.

R & M & R & P & M & N T P

Je tiens à remercier mon promoteur Monsieur F. CHIGARA pour son aide et conseils ainsi que tous qui d'une manière ou d'une autre m'ont aidé à l'élaboration de ce travail.

Ce travail doit également beaucoup à LAHCENE qui prépare une thèse de Magister dans le même domaine et qui m'a permis de faire évoluer ce travail. Je souhaite que ce mémoire soit l'expression de ma profonde Amitié.

Que Messieurs les membres du Jury trouvent ici l'expression de mon profond respect.

A tous les enseignants ayant contribué à ma formation, ma profonde gratitude.

بسم الله الرحمن الرحيم

ملخص

يتطرق هذا البحث الى استعمال صيغة جديدة لتصغير النماذج المركبة المسماة صيغة «الانجازات المتوازنة» المتمثلة في فضاء الحالات. تحتوي هذه الطريقة على مرحلتين اساسيتين، الأولى مرحلة «التوازن»، أين تكون المصفوفتان "Wo" و "Wg" متساويتين الى مصفوفة قطرية Σ التي تستعمل في المرحلة الثانية قصد تصغير النماذج المركبة استناداً بمعيار هنكل « Critère de Hankel ».

في الأخير، أعطيت بعض الأمثلة لتصغير نماذج «متواصلة» و«عددية» بهدف تبين مدى فعالية وجودة الطريقة المستعملة.

Abstract :

A recent method of system reduction is presented in this study, this method is called "balanced realisations"

In the state space representation, this method is performed in two steps. The first step is the balance of the system, where the two gramians "Wo" and "Wg" must be diagonal and equal to a matrix Σ Which will be used, later, in the second step in order to reduce the system according to the Hankel Criterion.

At the end of this study, some examples of the reduction of "Continuos" and "Discret-time" systems are given in order to illustrate the problem of reduction and particularly to show the performance and the efficiency of the adopted method.

Résumé :

Une récente méthode de réduction de systèmes complexes est présentée dans cette étude, appelée " Méthode des Réalisations Equilibrées". Dans l'espace d'état, cette méthode comprend essentiellement deux parties fondamentales .

La première, étape d'équilibre où les deux grammiens "Wo" et "Wg" doivent être diagonaux et égaux à une matrice Σ , qui, par la suite, dans la deuxième partie va être utilisée pour la réduction des systèmes complexes, selon le critère de Hankel.

Pour mieux illustrer le problème de réduction et afin de mettre en évidence la performance, ainsi que l'efficacité de la méthode adoptée, nous proposons à la fin de cette étude quelques exemples concernant la réduction des systèmes "continus" et "discrets".

Mots clés : Systèmes complexes, espace d'état, grammiens (de gouvernabilité, d'observabilité), valeurs singulières, équilibre, réduction du modèle.

- INTRODUCTION.	1
- Chapitre I. GENERALITES SUR LES SYSTEMES.	
I.1. Notion de systèmes.	2
I.2. Notion de grands systèmes.	2
I.3. Classification des systèmes.	3
I.4. Differentes représentations et propriétés des systèmes.	4
I.4.1. Cas Continu.	4
I.4.2. Cas Discret.	10
I.4.3. Passage des systèmes continus aux systèmes discrets et vis-versa.	14
I.5. Robustesse.	16
I.6. Conclusion.	16
- Chapitre II. PRINCIPALES METHODES DE REDUCTION.	
II.1. Introduction.	17
II.2. Qualités d'un modèle réduit.	17
II.3. Méthodes de réduction.	18
II.3.1. Approximation de type Padé.	18
II.3.2. Réduction optimale.	18
II.3.3. Méthodes modales.	19
II.3.4. Perturbations singulières.	21
II.4. Conclusion.	23
- Chapitre III. L'EQUILIBRE.	
III.1. Généralités.	24
III.2. Principe de l'équilibre.	24
III.2.1. Calcul des Gramiens.	25
III.2.2. Base équilibrée et maximisation d'énergie.	27
III.3. Algorithme d'équilibre.	31
III.4. Applications.	35
III.5. Conclusion.	40

- Chapitre IV. LA REDUCTION.	
IV.1. Introduction.	41
IV.2. Critère énergétique de réduction.	41
IV.3. Principe de réduction.	42
IV.4. Propriétés du système réduit.	44
IV.5. Conclusion.	46
- Chapitre V. RESULTATS ET INTERPRETATION.	
V.1. Introduction.	48
V.2. Cas Discret.	48
V.3. Cas Continu.	69
V.4. Interpretation.	83
V.5. Conclusion.	83
- CONCLUSION GENERALE.	85
- REFERENCE BIBLIOGRAPHIQUE.	86

ANNEXES :

- Annexe 1. Rappel sur le calcul matriciel.
- Annexe 2. Algorithme S.V.D.
- Annexe 3. Algorithme de JAMESON.
- Annexe 4. Algorithme de CHOLESKY.
- Annexe 5. Démonstration.
- Listing du programme.

PRESENTATION DU TRAVAIL

L'objet de cette étude est l'analyse des systèmes complexes et leur réduction par la méthode des réalisations équilibrées. La procédure de cette réduction de modèle introduite par Moore [1] est basée sur la représentation équilibrée des systèmes linéaires asymptotiquement stables et invariants dans le temps.

La troncature des modèle [1] enveloppe un compromis entre l'ordre du modèle réduit et le degré pour lequel les caractéristiques du processus sont fidèlement conservés.

Un logiciel a été développé, des programmes en Fortran ainsi que le logiciel Matlab [5] ont été utilisés.

INTRODUCTION.

La plupart des processus réels existant en industrie s'avèrent des systèmes complexes, cette complexité provient de la dimension de sa structure, de son environnement, limite les approches globales classiques d'analyse et de commande.

Il est indéniable qu'une analyse exacte de ces processus soit extrêmement coûteuse, la nécessité de représenter le système par un modèle d'ordre réduit qui assure de suivre le comportement du système initial avec une certaine tolérance s'impose.

Dans la mesure où la complexité est liée à la grande dimension des modèles, un important travail a été fait sur les méthodes de réduction. Des efforts considérables sur les techniques de Perturbations singulières, d'Agrégation et la méthode des Réalisations Equilibrées, ont été investis. La réduction dans ces cas peut s'opérer soit en ne conservant que les modes les plus lents, soit en remplaçant le vecteur d'état x par un vecteur $z=Tx$ soit sur une base énergétique.

L'étude présentée dans ce mémoire porte sur la réduction de systèmes par la méthode des réalisations équilibrées. Elle s'étalera essentiellement en deux volets.

Dans la première partie, après avoir caractérisé les systèmes en mettant en évidence leurs propriétés les plus importantes, nous résumons les principales méthodes de réduction présentées respectivement dans les chapitres I et II.

La seconde partie enveloppera principalement les chapitres III et IV développant l'équilibre du système et sa réduction du à partir de sa réalisation équilibrée.

Le chapitre V sera consacré à quelques applications de cette méthode aux systèmes continus et discrets permettant d'illustrer la performance de la méthode suivies de leurs interprétations. Les graphes présentés seront tracés par le logiciel Grapher [23].

Une conclusion générale mettant en relief l'aboutissement de notre travail sera présentée.

En Annexes, seront donnés les différents algorithmes et outils mathématiques ainsi que le listing du programme élaboré à l'issue de cette étude.

CHAPITRE I GENERALITES SUR LES SYSTEMES

I.1 Notion de systèmes.

Un système est un arrangement ou une combinaison ordonnée de parties dans un tout [2], le mot système implique aussi que la combinaison des parties est faite selon un principe rationnel et représente un arrangement méthodique des parties. Le vocable sous-système sera réservé à un ensemble dont la structure physique est inconnue et qui n'est accessible que par ses entrées-sorties.

On définit aussi un système [3] comme un groupe indépendant ou un ensemble d'interactions régulières d'éléments formant un tout unifié.

I.2 Notion de Grands Systèmes [8].

Les problèmes des systèmes complexes (systèmes d'ordre élevé comprenant des sous-systèmes interconnectés) posent des difficultés énormes de point de vue analyse et commande; de tels systèmes sont rencontrés dans l'industrie et dans le domaine socio-économique.

Bien qu'il soit possible d'entâmer directement la phase des systèmes d'ordre réduit (définition des entrées-sorties, contrôle, construction du modèle, estimation des paramètres, définition du critère, etc...) ainsi que la phase de commande (synthèse et implémentation des algorithmes), ce qui n'est pas le cas de la plus grande majorité des systèmes dont l'ordre est élevé; les difficultés peuvent être purement théoriques (mauvaise convergence ou divergence de l'algorithme) ou peuvent découler des considérations économiques, de même le traitement des systèmes complexes exige:

i/ de nouvelles méthodes analytiques étant donné que la dimension est élevée,

2i/ la nécessité d'obtenir un état intermédiaire avant le contrôle; cet état doit consister :

- soit en la réduction de la dimension du problème par une procédure d'agrégation et l'application de techniques standards du problème réduit,
- soit en la décomposition du système en définissant des sous-systèmes adéquats, dans le but d'utiliser des méthodes de décomposition-coordination et la structure de commande à multiniveaux .

3i/ La synthèse des algorithmes de contrôle utilisant les principes de décomposition-coordination.

I.3 Classification des Systèmes.

Un modèle est une idéalisation d'un processus physique, il est utilisé pour réduire le calcul dans l'analyse et la conception des systèmes.

De tels systèmes peuvent être classés comme suit:

- Systèmes Déterministes-Systèmes Aléatoires [3].

Dans les systèmes déterministes, tous les paramètres sont décrits d'une manière exacte alors que pour les systèmes aléatoires, quelques (ou tous les) paramètres sont décrits d'une façon probabilistique.

- Systèmes Continus- Systèmes Discrets [4].

Un système continu s'il traite des signaux qui sont définis à des instants (t) continus, par contre un système est dit discret s'il traite des signaux discrets, c-à-d; convenablement définis à des instants multiples de la période d'échantillonnage T_0 .

- Systèmes Simples-Systèmes Complexes [18].

Dans la nomination de systèmes "complexes" concourent plusieurs paramètres telsque: incertitude structurelle, contraintes à structure informationnelle et grande dimension.

En contrast, on dit qu'un système est simple s'il ne possède pas les propriétés sus-citées.

- Systèmes S.I.S.O-Systèmes M.I.M.O [3].

Un système est dit S.I.S.O s'il possède une entrée et une seule sortie on l'appelle Système Monovariabale ou "Système Scalaire". Un système M.I.M.O est un système ayant r entrées et p sorties, il est dit Multivariable, on l'appelle aussi "Système Vectoriel".

Un système S.I.S.O est un cas simplifié du système M.I.M.O (r=p=1) comme le montrent les Fig.1 et Fig.2.

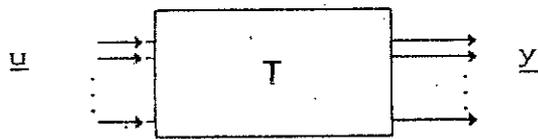


Fig.1

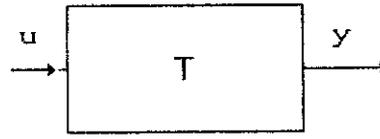


Fig.2

I-4 Différentes Représentations et Propriétés des Systèmes.

On traitera dans ce paragraphe, le cas général, c-à-d; les systèmes M.I.M.O possédant des couplages internes [2]. On distinguera le cas des systèmes continus puis le cas discret.

I.4.1 Systèmes Continus.

i/ Représentations.

Parmi les différentes représentations on en citera :

- Système d'équations différentielles (équation différentielle).

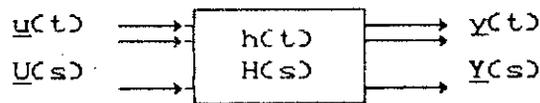
Par suite des couplages internes [2], chaque équation différentielle peut faire intervenir tout ou des parties des diverses entrées et sorties, ainsi que leur dérivées. Pour alléger l'écriture, en faisant intervenir l'opérateur de différentiation $D = \partial/\partial t$, on met le système sous forme matricielle:

$$LCD)y = MCD)u \tag{1.1}$$

où :

$LCD)$ est une matrice de dim. (p,p),
 $MCD)$ est une matrice de dim. (p,m).

- Matrice des Réponses impulsionnelles (Réponse Impulsionnelle).



Par définition [3], la réponse impulsionnelle d'un système est la réponse d'un système à une impulsion de Dirac [4]:

$$u_i(t) = \delta(t) = \begin{cases} \infty & t=0, \\ 0 & t \neq 0. \end{cases} \quad (1.2)$$

$$U_i(s) = L(\delta(t)) = 1 \quad (1.3)$$

où L est l'opérateur de Laplace.

La matrice des réponses impulsionnelles $h(t)$ de dim. (p,r) est telle que:

$$y(t) = \int_0^{\infty} h(\tau) u(t-\tau) d\tau \quad (1.4)$$

-Matrice de transfert (Fonction de transfert).

Si l'on prend la transformée de Laplace [3] de l'équation (1.1) avec l'hypothèse que les conditions initiales sont nulles quelque soit $t < 0$, on obtient donc [4]:

$$Y(s) = \int_0^{\infty} y(t) \text{Exp}(-st) dt \quad (1.5)$$

$$Y(s) = H(s) U(s). \quad (1.6)$$

où :

$Y(s) = L(y(t))$:Laplacien du vecteur de sortie $y(t)$ de dim. (m) ;

$H(s) = L(h(t))$:Laplacien de la matrice des réponses impulsionnelles $h(t)$ de dim. (p,m) ;

$U(s) = L(u(t))$:Laplacien du vecteur d'entrée $u(t)$ de dim. (r) .

- Matrice de la réponse fréquentielle.

Cette matrice est obtenue en évaluant la matrice de transfert sur l'axe imaginaire ($s = j\omega$) [4].

C-a-d :

$$H(\omega) = H(s) \Big|_{s=j\omega} \quad (1.7)$$

$$H(\omega) = \int_0^{\infty} h(t) \exp(-j\omega t) dt. \quad (1.8)$$

- Représentation d'Etat.

Dans le cas général où le système différentiel représentant le fonctionnement dynamique du système physique considéré est linéaire, à coefficients constants, d'ordre n , on peut lui associer une représentation d'état de la forme [2]:

$$\begin{cases} \dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t) \\ \underline{y}(t) = C\underline{x}(t) + D\underline{u}(t) \end{cases} \quad (1.9)$$

où :

- $\underline{x}(t)$ est le vecteur d'état de dim. (n);
- $\underline{u}(t)$ est le vecteur d'entrée (commande) de dim. (r),
- $\underline{y}(t)$ est le vecteur de sortie de dim. (p),
- A : Matrice d'évolution de dim. (n,n),
- B : Matrice de commande de dim. (n,r),
- C : Matrice d'observation de dim. (p,n),
- D : Matrice de couplage entrées/sorties de dim. (p,r).

Il est très intéressant de signaler que cette représentation est un outil mathématique puissant [2] puisqu'il jouit des principales qualités, soient:

- * Les conditions initiales apparaissent explicitement comme dans le système d'équations différentielles (équation différentielle),
- * Les matrices qui interviennent sont des matrices à coefficients constants, elles relèvent de l'Algèbre Linéaire.

- Robustesse. La conception des systèmes de commandes est basée sur des modèles simplifiés.

Remarques.

Ces propriétés restent aussi valables en discret.

2i/ Structure Canonique de Kalman.

Dans le cas général, un système S comporte de sous-systèmes aux différentes propriétés de Gouvernabilité et d'Observabilité:

$$S = S_1 \cup S_2 \cup S_3 \cup S_4 .$$

possède la structure canonique de Kalman dans laquelle:

- S₁ est le sous-syst. gouvernable, non observable,
- S₂ est le sous-syst. gouvernable, observable,
- S₃ est le sous-syst. non gouvernable, non observable,
- S₄ est le sous-syst. non gouvernable, observable.

$$\begin{bmatrix} dx_1/dt \\ dx_2/dt \\ dx_3/dt \\ dx_4/dt \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & A_{22} & 0 & A_{24} \\ 0 & 0 & A_{33} & A_{34} \\ 0 & 0 & 0 & A_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} \underline{U} \quad (1.13)$$

$$\underline{Y} = [0 \ C_2 \ 0 \ C_4] \underline{X}$$

3i/ Critères de Kalman Gouvernabilité et d'Observabilité.

Considérons le système de dimension n à r entrées et p sorties, soit:

$$dx(t)/dt = Ax(t) + Bu(t) \quad \text{ou} \quad x(k+1) = Ax(k) + Bu(k). \quad (1.14)$$

Le système est dit gouvernable ssi:

$$\text{rang}[B, AB, \dots, A^{n-1}B] = n \quad [A1] \quad (1.15)$$

* Le système libre représenté par:

$$\begin{cases} dx(t)/dt = Ax(t) \\ y(t) = Cx(t) \end{cases} \quad \text{ou} \quad \begin{cases} x(k+1) = Ax(k) \\ y(k) = Cx(k) \end{cases} \quad (1.16)$$

est dit observable ssi:

$$\text{rang}[C^T, A^T C^T, \dots, A^{(N-1)T} C^T] = n \quad (1.17)$$

* On notera l'intéressante propriété de Dualité :

	DUALITE	
Gouvernabilité	↔	Observabilité.
(A, B) gouvernable	↔	(A ^T , C ^T) observable.
(A, C) observable	↔	(A ^T , B ^T) gouvernable.

Il vient qu'au lieu d'étudier les deux propriétés séparément, on se limitera à celle de la gouvernabilité par exemple, l'autre se déduira par Dualité.

1.4.2 Cas Discret.

1/ Représentations des systèmes discrets.

Les techniques mathématiques pour l'analyse des systèmes invariants dans le temps, linéaires, discrets, sont appelés "Techniques du domaine à séquences" car les entrées, sorties et le modèle du système sont décrits par des séquences de nombres [6]. Parmi les diverses représentations de ces systèmes, on distingue:

- Matrice des Réponses impulsionnelles.

La réponse impulsionnelle $h_{ij}(k)$ d'un système est la $j^{\text{ème}}$ réponse du système lorsque la $i^{\text{ème}}$ entrée est excitée par l'impulsion de Kröneckner $\delta(k)$ définie par [4]:

$$\delta(k) = \begin{cases} 1 & \text{si } k=0 \\ 0 & \text{sinon} \end{cases} \quad (1.18)$$

c-à-d;

$$h_{ij}(k) = T(u_i = \delta(k)). \quad (1.19)$$

- Systèmes d'équations aux différences (équation récurrente).

Le système sera décrit par l'équation ci-dessous [3]:

$$\sum_{n=0}^N a_{in}(k) y_i(k-n) = \sum_{m=0}^M b_{jm}(k) u_j(k-m) \quad (1.20)$$

- où :
- u_j : est la $j^{\text{ème}}$ entrée du système ($j=\overline{1,r}$),
 - y_i : est la $i^{\text{ème}}$ sortie du système ($i=\overline{1,p}$),
 - N et M sont des entiers positifs,
 - N est l'ordre du système,
 - $a_{in}(k)$, $b_{jm}(k)$ sont des constantes indépendantes de k
(pour des systèmes linéaires stationnaires).

- Matrice de transfert.

A partir des équations aux différences d'ordre N , on passe à la transformée en Z [3]:

$$\sum_{k=-\infty}^{+\infty} \left\{ \sum_{n=0}^N a_{in} y_i(k-n) \right\} Z^{-k} = \sum_{k=-\infty}^{+\infty} \left\{ \sum_{m=0}^M b_{jm} u_j(k-m) \right\} Z^{-k} \quad (1.21)$$

avec $i=\overline{1,r}$ et $j=\overline{1,p}$.

En utilisant les propriétés de linéarité, décalage, de la transformée en Z , on peut mettre (1.21) forme:

$$Y_i(Z) \sum_{n=0}^N a_{in} Z^{-n} = U_j(Z) \sum_{m=0}^M b_{jm} Z^{-m} \quad (1.22)$$

$$H_{ij}(Z) = \frac{Y_i(Z)}{U_j(Z)} = \frac{\sum_{m=0}^M b_{im} Z^{-m}}{\sum_{n=0}^N a_{jn} Z^{-n}} \quad (1.23)$$

avec: $\left\{ \begin{array}{l} - i = \overline{1, p} , \\ - j = \overline{1, r} , \\ - Y_i \text{ est le signal requis à la } i^{\text{ème}} \text{ sortie quand la } j^{\text{ème}} \text{ entrée est excitée.} \end{array} \right.$

- Matrice des réponses fréquentielles $H(\omega)$.

C'est la transformée de Fourier [4] de la matrice des réponses impulsionnelles ($h(k)$), avec:

$$\left\{ \begin{array}{l} i = \overline{1, r} , \\ j = \overline{1, p} . \end{array} \right.$$

La matrice des réponses fréquentielle est la matrice de transfert évaluée sur le cercle unité:

$$H(\omega) = H(Z) \Big|_{Z = \exp(j\omega)} \quad (1.24)$$

- Représentation dans l'Espace d'Etat :

C'est une description donnée en termes des états du système linéaire discret rassemblés dans le vecteur d'état \underline{x} [2].

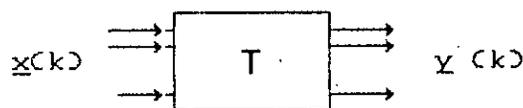
Elle se base sur l'équation d'état et l'équation d'observation:

$$\begin{cases} \underline{x}(k+1) = A\underline{x}(k) + B\underline{u}(k) \\ \underline{y}(k) = C\underline{x}(k) + D\underline{u}(k) \end{cases} \quad (1.25)$$

- Où:
- $\underline{x}(k)$ est le vecteur d'état de dim. (n),
 - $\underline{u}(k)$ est le vecteur d'entrée (commande) de dim.(r),
 - $\underline{y}(k)$ est le vecteur de sortie de dim. (p),
 - A : Matrice d'évolution de dim. (n,n),
 - B : Matrice de commande de dim. (n,r),
 - C : Matrice d'observation de dim.(p,n),
 - D : Matrice de couplage entrées/sorties de dim. (p,r).

21/ Propriétés.

Un système discret multivariable est une transformation (T) reliant les signaux discrets d'entrées \underline{u} à ceux de sorties \underline{y} [6].



Un tel système est représenté par l'équation :

$$\underline{y} = T(\underline{u}) \quad (1.26)$$

Les valeurs de la sortie \underline{y} à l'instant discret k est donnée par:

$$\underline{y}(k) = T(\underline{u}(k)) \quad (1.27)$$

- Linéarité. Le système est dit linéaire si pour tous signaux $\underline{u}_1(k), \underline{u}_2(k)$ et les constantes a, b on a:

$$\begin{aligned} T(a\underline{u}_1(k) + b\underline{u}_2(k)) &= aT(\underline{u}_1(k)) + bT(\underline{u}_2(k)) \\ &= a\underline{y}_1(k) + b\underline{y}_2(k) \end{aligned} \quad (1.28)$$

On dira qu'un tel système vérifie la contrainte de superposition (homogénéité et additivité).

- Invariance temporelle. Un système est dit invariant dans le temps si un décalage à l'entrée n'introduit pas une distorsion du signal de sortie.

C-à-d; si $\underline{y}(k) = T(\underline{u}(k))$

$$\text{alors } T(\underline{u}(k+m)) = \underline{y}(k+m), \forall m \in \mathbb{Z}. \quad (1.29)$$

- **Causalité.** Un système discret causal est caractérisé par le fait que sa réponse ne précède jamais son excitation, c-à-d; un système est causal s'il satisfait l'une des conditions suivantes:

- * $h(k)=0$ pour $k < 0$
- * le degré du dénominateur de la fonction de transfert $H(Z)$ du système est supérieur à celui du numérateur.

-**Stabilité.** Un système est dit stable s'il satisfait l'une des conditions suivantes [4]:

-Sa réponse impulsionnelle $h(k)$ satisfait :

$$\sum_{k=-\infty}^{+\infty} |h(k)| < \infty \quad (1.30)$$

-Si les pôles de la matrice de transfert appartiennent au cercle unité .

-Si les valeurs propres de la matrice (A) appartiennent au cercle unité.

1.4 Passage des Systèmes Continus aux Systèmes Discrets et vis-versa [27].

Soit un système continu, représenté dans l'espace d'état : (A, B, C, D) et ayant pour fonction de transfert:

$$G(s) = D + C(s.I - A)^{-1}B$$

Considérons la transformation bilinéaire [6], reliant la variable complexe de Laplace (s) du plan fréquentiel continu, à celle du plan fréquentiel discret (Z), donnée par:

$$s = \beta \frac{Z-1}{Z+1} \quad \longrightarrow \quad Z = \frac{\beta+s}{\beta-s}$$

où :
$$\beta = \frac{2}{T_0}; T_0: \text{période d'échantillonnage.}$$

En général, on prend $T_0 = 2$ unités de temps $\rightarrow \beta = 1$

La fonction de transfert $G_d(Z)$ du système discret correspondant est alors donnée par:

$$G(z) = G(s = \frac{z-1}{z+1}) = J + H.(zI - \Phi^{-1}).\Gamma$$

où (Φ, Γ, H, J) est la représentation d'état du système discret,

avec:

$$\begin{cases} - \Phi = (I + A).(I - A)^{-1} \\ - \Gamma = \sqrt{2} . (I - A)^{-1} . B \\ - H = \sqrt{2} . C . (I - A)^{-1} \\ - J = D + C . (I - A)^{-1} . B \end{cases}$$

ce qui implique aussi:

$$\begin{cases} A = (\Phi + I)^{-1} (\Phi - I) \\ B = \sqrt{2} (\Phi + I)^{-1} \Gamma \\ C = \sqrt{2} (\Phi + I)^{-1} \\ D = J - H (\Phi + I)^{-1} \Gamma \end{cases}$$

Remarques:

1/ Si P est solution de l'équation matricielle:

$$A.P + P.A^T + B.B^T = 0,$$

P est aussi solution de l'équation matricielle:

$$\Phi.P.\Phi^T - P + \Gamma.\Gamma^T = 0$$

2i/ Si Q est solution de l'équation matricielle:

$$A^T.Q + Q.A + C^T.C = 0$$

Q est aussi solution de l'équation matricielle:

$$\Phi^T Q \Phi - Q + H^T H = 0$$

De 1, 2i et du fait que la transformation bilinéaire conserve la stabilité (Demi plan gauche (s) transformée, l'intérieur du cercle unité du plan (z)), le système continu (A,B,C) est gouvernable et observable si et seulement si le système discret (Φ, Γ, H) et gouvernable et observable.

1.5 Robustesse [25].

C'est la sensibilité du système aux perturbations introduites à cause de la tolérance des composants car la conception des systèmes de commandes est basée sur des modèles simplifiés.

1.6 Conclusion.

Dans ce chapitre, ont été introduites la notion de systèmes, avec leurs classification, importantes propriétés ainsi que leurs diverses représentations. Il sera suivi d'un chapitre dans lequel on discutera sur les principales méthodes de réduction.

CHAPITRE II PRINCIPALES METHODES DE REDUCTION

II-1 Introduction.

Les contraintes numériques représentent le plus grand problème que doit faire face le concepteur lors des phases d'analyse et de conception de grands systèmes. Ces contraintes sont d'autant plus importantes dans les applications en temps. La dimension n du système est élevée (>10): le temps de calcul devient prohibitif (n^2), l'espace mémoire (n^2) trop important [7].

Il devient donc nécessaire de faire un compromis entre la complexité du modèle qui représente plus la réalité que la précision et la nécessité d'avoir un modèle simple pour le commander [8].

La réduction de la dimension d'un système s'accompagne toujours d'une perte d'informations qui reste un problème majeur. Pour préciser les qualités essentielles que doit envelopper un modèle réduit, on doit définir certaines de ses propriétés.

II-2 Qualités d'un modèle réduit [7].

* Le modèle réduit doit présenter les mêmes caractéristiques de stabilité que le modèle initial.

* Pour une classe d'entrées donnée, les sorties mesurables du modèle doivent se rapprocher le plus possible des sorties réelles.

* Conservation du gain statique.

-Le modèle réduit doit satisfaire le compromis "Simplicité, Précision". La "structure simplifiée" importante propriété, résultera d'un choix d'ordre minimal du modèle réduit.

* Bonne approximation de la réponse fréquentielle ($H(\omega)$).

* Conserver les relations entre les états du modèle initial et ceux du modèle réduit, c-à-d; garder une signification physique aux variables d'état.

La réduction de modèle doit :

- * Chiffrer l'écart entre modèle réduit et modèle complet,
- * Limiter le volume du calcul,
- * Etre robuste,
- * Guider le choix de l'ordre du modèle.

II-3 Méthodes de réduction

Se basant sur le domaine dans lequel le modèle d'un système est représenté, les méthodes de réduction sont groupées en deux catégories [3]:

- domaine temporel,
- domaine fréquentiel.

Les méthodes dans le domaine temporelle sont associées à l'espace d'état et celles dans le domaine fréquentiel sont associées à la fonction de transfert.

Le problème de réduction a été considéré pour les systèmes continus, on étendra les résultats aux systèmes discrets.

II-3.1 Approximation de type Padé.

Cette approche est basée sur des développements limités de type Taylor dans le domaine (s). Elle s'adapte parfaitement à l'opérateur de transfert et garantie généralement une conservation des gains statiques [7]. De nombreuses variantes ont été proposées avec notamment [3] l'approximation de Routh mais n'assure pas la stabilité, elle n'est de fait, pas adaptée à la régulation.

II-3.2 Réduction Optimale.

C'est une technique basée sur la minimisation de l'erreur de sortie par un critère quadratique pour des entrées d'un type déterminé [7]. Le critère s'écrit généralement:

$$J = \sum_{i=1}^p \text{tr} \left(\int_0^{+\infty} [y_i(t) - \hat{y}_i(t)][y_i(t) - \hat{y}_i(t)]^T dt \right) \quad (2.1)$$

où: p est le nombre de sorties et \hat{y} la sortie du modèle réduit. La recherche de l'optimum se traduit par une programmation non linéaire et peut se résoudre par des équations de Lyapunov. Les calculs sont lourds et la convergence délicate. Si cette méthode présente l'avantage de donner le modèle optimal pour un critère et un modèle donnés (applicable en multivariable), elle ne permet pas de fixer l'ordre du modèle réduit à priori et surtout, il n'existe pas de relation entre l'état des deux systèmes, ce qui empêche toute signification physique des états du modèle réduit. Le choix du critère est délicat et la mise en oeuvre est généralement lourde.

II-3.3 Méthodes Modales.

Ces méthodes sont basées sur la méthode d'Agrégation introduite par Aoki ([7],[3]). Le modèle réduit est obtenu par séparation des modes du système en deux groupes, les modes dominants instables ou lents et les modes rapides avec conservation uniquement des modes dominants. Partant d'un système initial, continu, supposé complètement commandable et observable,

$$\begin{cases} dx(t)/dt = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (2.2)$$

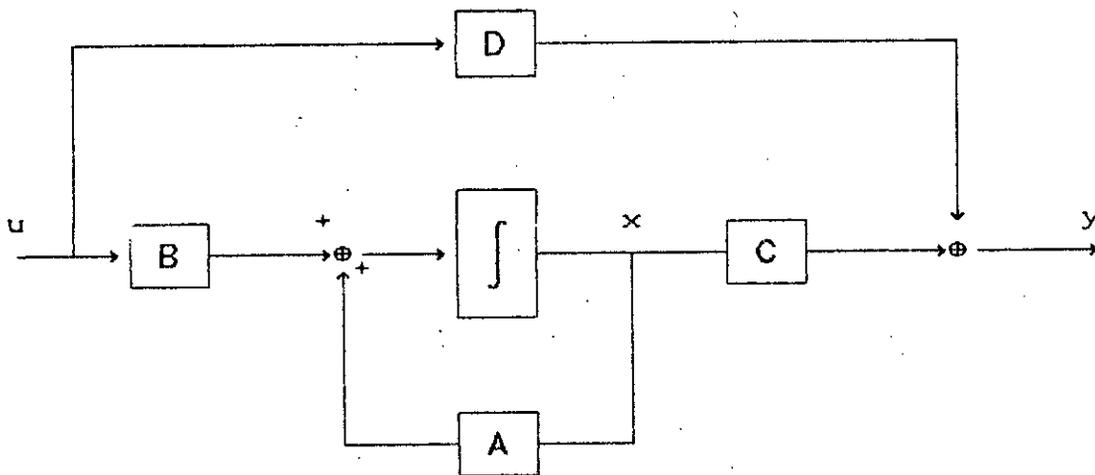


Fig.2.1. Système Initial.

un modèle réduit de dimension m ($r \leq m \leq n$) d'équations:

$$\begin{cases} dz(t)/dt = Fz(t) + Gu(t) \\ \hat{y}(t) = Hz(t) + Eu(t) \end{cases} \quad (2.3)$$

est dit " Modèle agrégé ", si:

- les états $z(t)$ et $x(t)$ des modèles (1) et (2) sont reliés par une relation linéaire du type:

$$Z(t) = L x(t) \text{ où } L \text{ est une matrice de dim. } (m,n),$$

- les matrices F et G vérifient les conditions suivantes:

$$\begin{cases} FL = LA \\ G = LB \end{cases}$$

- La matrice L est supposée de rang maximum.

La stabilité du modèle simplifié est assurée et les états du modèle réduit sont reliés à ceux du modèle initial par une relation linéaire.

Cependant, elle présente l'inconvénient de nécessiter le calcul, long pour des systèmes complexes, des valeurs propres du modèle et pose le problème d'un bon choix des modes pour le modèle réduit.

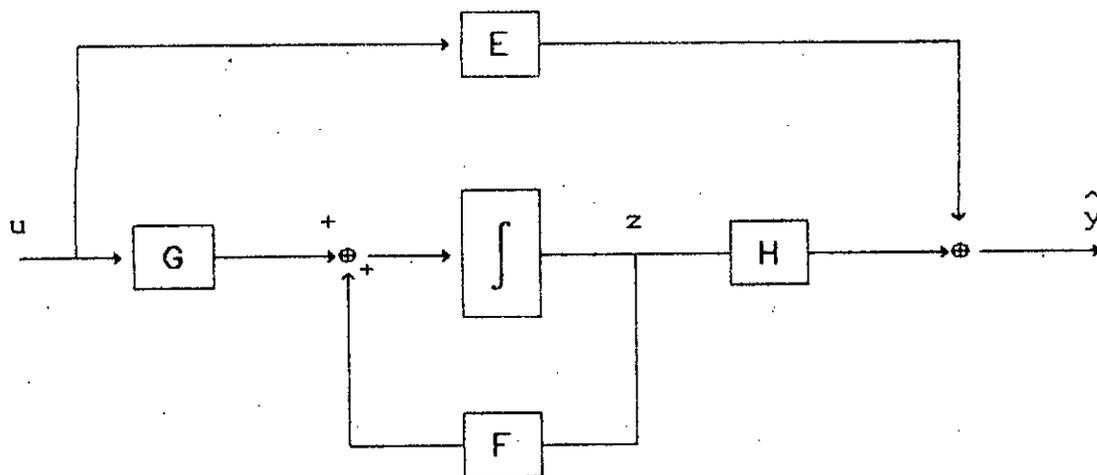


Fig. 2.2. Système Réduit.

II-3.4 Perturbations Singulières.

Cette approche a été introduite par Kokotovic [7] pour les systèmes continus à plusieurs échelles de temps, étendue aux systèmes discrets, possédant des variables qui varient rapidement. Ce type de systèmes peuvent se mettre sous la forme:

$$\begin{cases} dw(t)/dt = Aw(t) + Bz(t) & , w(t_0) = w_0 \\ \mu dz(t)/dt = Cw(t) + Dz(t) & , z(t) = z_0 \end{cases} \quad (2.4)$$

L'inconvénient de cette technique est qu'il n'est pas toujours aisé de mettre un système sous la forme (2.4). Il n'existe pas de relation exacte entre le système complet et son modèle réduit:

$$\hat{z}(t_0) \neq z(t_0). \quad (2.5)$$

II-3.5 Troncature dans la Base d'Equilibre (M.R.E).

Cette méthode a été introduite par Moore [1]. Elle se fonde sur la recherche des valeurs singulières dans la base d'équilibre [7], où les deux gramiens sont égaux. Pour un système continu de la forme:

$$\begin{cases} dx(t)/dt = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (2.6)$$

Les degrés de commandabilité et d'observabilité à l'instant T sont obtenus par décomposition en valeurs singulières des gramiens de commandabilité et d'observabilité donnés respectivement par:

$$W_g = \int_0^T \text{Exp}(At) B B^T \text{Exp}(A^T t) dt. \quad (2.7)$$

$$W_o = \int_0^T \text{Exp}(A^T t) C^T C \text{Exp}(At) dt. \quad (2.8)$$

En effectuant un changement de base, la représentation équilibrée du système s'obtiendra en se plaçant dans celle où $W_g = W_o = \Sigma$ est une matrice diagonale.

En notant par $\Sigma_g = V_g \Sigma_g^T$ et $\Sigma_o = V_o \Sigma_o^T$ la décomposition en valeurs singulières des matrices W_g et W_o respectivement, la matrice H définie par:

$$H = \Sigma_o V_o V_g \Sigma_g^T \quad (2.9)$$

est dite " *Matrice de Hankel* " [1], admet pour éléments des valeurs singulières invariantes par un changement de base qui sont les modes du second ordre du système continu. Il est possible de classer les modes les plus gouvernables et les plus observables et de ne conserver que les significatifs.

La troncature dans la base d'équilibre est indépendante du type d'entrée.

Une variante a été proposée par KABAMBA ([13],[16]), c'est une technique de troncature basée sur les gains équilibrés. Tout comme la M.R.E elle n'assure pas un modèle réduit optimal [13].

Nous développerons la méthode des réalisations équilibrées dans les chapitres suivants.

II.4 Conclusion.

Dans le présent chapitre ont été brièvement exposées les plus importantes techniques de réduction, la plus récente, celle basée sur l'équilibre, opération qui fera objet du chapitre suivant.

CHAPITRE III

L'EQUILIBRE

III-1 Généralités.

On sait qu'une représentation dans l'espace d'état n'est pas unique, ses divers modèles sont reliés entre eux par une transformation régulière [A1] de similarité, c-à-d; les représentations sont toutes équivalentes via cette transformation [9], mais ceci n'implique pas l'invariance de toutes les propriétés du systèmes (l'invariance des modes du système conduit à la conservation de la stabilité) sous une transformation de coordonnées telles que les propriétés de gouvernabilité et d'observabilité qui dépendent du choix de la base de coordonnées [12], elles sont appelées respectivement *Gramiens de gouvernabilité*, *d'observabilité* et sont notées W_g, W_o .

Equilibrer une réalisation [11] revient à rendre symétrique une certaine propriété d'entrée (gouvernabilité) avec une certaine propriété de sortie (observabilité) en choisissant une certaine base de coordonnées. Il est alors nécessaire de trouver une nouvelle représentation dans laquelle ces deux grammiens soient égaux et diagonaux.

Un choix d'un gramien d'entrée particulier avec un gramien de sortie particulier est dit " *Paire Equilibrée* " si le spectre du produit des deux quantités est indépendant du système de coordonnées (les pôles de la fonction de transfert) :

$$\begin{array}{ccc} (A, B, C, D) & \xleftrightarrow{T_b} & (A_b, B_b, C_b, D) \\ (W_o, W_g) & \xleftrightarrow{\quad} & (\Sigma, \Sigma) \end{array}$$

III-2 Principe de l'Equilibre.

Le principe général de l'équilibre est d'associer à une classe particulière de systèmes un ensemble d'équations de Ricatti qui sont reliées d'une façon intrinsèque aux propriétés particulières des systèmes.

Ainsi la classe des systèmes asymptotiquement stables est associée à l'ensemble des équations de Lyapunov [9], une question qui peut se poser est ce que réalisation équilibrée signifie pour une telle classe de systèmes, ce sera l'objet des paragraphes suivants.

III-2.1 Calcul des gramiens.

On traitera tout d'abord le cas d'un système continu puis celui d'un système discret.

- Cas Continu [1]

Dans le contexte des systèmes continus linéaires décrits par:

$$\begin{cases} dx/dt = Ax(t) + Bu(t) & \text{avec } x(0) = x_0 \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (3.1)$$

où:

$$\begin{cases} x(t) \in \mathbb{R}^n, \\ u(t) \in \mathbb{R}^r, \\ y(t) \in \mathbb{R}^p. \end{cases}$$

Le système (3.1) est supposé gouvernable, observable et asymptotiquement stable.

La paire (A,B) est utilisée pour le calcul de la gouvernabilité, tandis que la paire (A,C) assure le calcul de l'observabilité.

- Définition [13].

Soit le système (3.1). Supposons que les valeurs propres de la matrice d'évolution A appartiennent au demi plan gauche du Plan de (s), le gramien de gouvernabilité du système est défini par :

$$W_g = \int_0^{t_f} \text{Exp}(At) B B^T \text{Exp}(A^T t) dt \quad (3.2)$$

et le gramien d'Observabilité du système est défini par :

$$W_o = \int_0^{t_f} \text{Exp}(A^T t) C^T C \text{Exp}(A t) dt \quad (3.3)$$

Où t_f est le temps final.

Quand $t_f \rightarrow +\infty$, les quantités W_g et W_o (matrices définies positives) sont solutions uniques des équations algébriques de Lyapunov:

$$\begin{cases} AW_g + W_g A^T + BB^T = 0 \\ A^T W_o + W_o A + C^T C = 0 \end{cases} \quad (3.4)$$

L'algorithme de JAMESON [A3] permettant de résoudre les équations de Lyapunov, est utilisé pour le calcul des deux gramiens (W_o, W_g).

- Cas Discret.

Le modèle est décrit par la représentation suivante:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) & \text{avec } x(0) = x_0 \\ y(k) = Cx(k) + Du(k) \end{cases} \quad (3.5)$$

où:

$$\begin{cases} x(k) \in \mathbb{R}^n, \\ u(k) \in \mathbb{R}^r, \\ y(k) \in \mathbb{R}^p. \end{cases}$$

Par définition [15], on posera:

$$W_g = \sum_{k=0}^{+\infty} A^k B B^T (A^T)^k \quad (3.6)$$

$$W_o = \sum_{k=0}^{+\infty} (A^T)^k C^T C A^k \quad (3.7)$$

Les matrices W_o et W_g représentent respectivement les gramiens d'Observabilité et de Gouvernabilité du système (3.1).

Si le système (3.1) est stable, autrement dit les valeurs propres de A sont à parties réelles négatives, les deux sommes (3.6) et (3.7) convergent pour un nombre fini de N termes. Les deux grammiens sont aussi solutions uniques des équations de Lyapunov (3.8) [9]:

$$\begin{cases} AW_g A^T - W_g + BB^T = 0 \\ A^T W_o A - W_o + C^T C = 0 \end{cases} \quad (3.8)$$

III-2.2 Base Equilibrée et Maximisation d'Energie.

Pour illustrer le principe d'équilibre il est nécessaire de mettre en évidence la relation des grammiens avec les relations d'entrées-sorties, pour cela considérons le problème de commande suivant [11]:

Supposant que le système est à l'état initial à l'instant t_0 et soit une fonction d'entrée $u(t)$ à énergie finie.

Si le système est complètement gouvernable, alors le gramien de gouvernabilité est inversible et l'énergie minimale E_1 nécessaire pour commander l'état dans la direction $e_1 = [0 \dots \hat{e}_1 \dots 0]$ telle que $\hat{e}_1 = 1$ est donné par:

$$E_1 = \int_{-\infty}^t ||u(\tau)||^2 d\tau \quad (3.9)$$

où:

$$u(\tau) = B^T(\tau) \Phi^T(t, \tau) W_g^{-1}(t) e_1$$

$$\int_{-\infty}^t ||u(\tau)||^2 d\tau = \int_{-\infty}^t u^T(\tau) u(\tau) d\tau$$

$$W_g(t) = \int_{-\infty}^t \Phi(t, \tau) B(\tau) B^T(\tau) \Phi^T(t, \tau) d\tau$$

$\Phi(t, \tau)$: matrice de transition [2].

Dans le cas des systèmes invariants dans le temps :

Par résolution des valeurs propres [A1], on décompose:

$$W_g = U_g^T \Sigma_g U_g \quad (3.10)$$

$$W_g = \begin{bmatrix} U_{g1} & U_{g2} & \dots & U_{gn} \end{bmatrix} \begin{bmatrix} \sigma_{g1} & & & 0 \\ & \sigma_{g2} & & \\ & & \dots & \\ 0 & & & \sigma_{gn} \end{bmatrix} \begin{bmatrix} U_{g1} \\ U_{g2} \\ \vdots \\ U_{gn} \end{bmatrix} \quad (3.11)$$

où:

- $\Sigma_g = \text{diag} \{ \sigma_{g1}, \sigma_{g2}, \dots, \sigma_{gn} \}$,
 - σ_{gi} est la $i^{\text{ème}}$ valeur propre de W_g ,
 - U_{gi} est le $i^{\text{ème}}$ vecteur colonne de U_g , vecteur propre associé à la valeur propre σ_{gi} ,
 - U_g est une matrice unitaire [A1],
- $i = \overline{1, n}$

En injectant l'équation (3.11) dans (3.9), on obtient:

$$E_1 = \sigma_{g1}^{-1} \quad (3.12)$$

La puissance de sortie E_o qu'on retrouve à la sortie de l'état e_1 est définie par :

$$E_o = \int_{-\infty}^t \|y(\tau)\|^2 d\tau \quad (3.13)$$

où:

$$\int_{-\infty}^t \|y(\tau)\|^2 d\tau = \int_{-\infty}^t y^T(\tau)y(\tau) d\tau$$

$$W_o(t) = \int_{-\infty}^t \Phi^T(t, \tau) C^T(t) C(\tau) \Phi(t, \tau) d\tau.$$

Dans le cas des systèmes invariants dans le temps et par résolution du problème des valeurs propres [14], on décompose:

$$W_o = U_o^T \Sigma_o U_o \quad (3.14)$$

$$W_o = \begin{bmatrix} U_{o1} & U_{o2} & \dots & U_{on} \end{bmatrix} \begin{bmatrix} \sigma_{o1} & & 0 \\ & \ddots & \\ 0 & & \sigma_{on} \end{bmatrix} \begin{bmatrix} U_{o1} \\ \vdots \\ U_{on} \end{bmatrix} \quad (3.15)$$

où :

- $\Sigma_o = \text{diag} \{ \sigma_{o1}, \sigma_{o2}, \dots, \sigma_{on} \}$,
 - σ_{oi} est la $i^{\text{ème}}$ valeur propre de W_o ,
 - U_{oi} est le $i^{\text{ème}}$ vecteur colonne de U_o , c'est le vecteur propre associé à la valeur propre σ_{io} ,
 - U_o est une matrice unitaire [A1],
- $i = \overline{1, n}$

En injectant l'équation (3.15) dans (3.13), on obtient

$$E_o = \sigma_{oi}^{-1} \quad (3.16)$$

Se basant sur les équations (3.12) et (3.16), on peut voir que si dans une certaine représentation d'état σ_{gi} est faible, les états dans la direction e_i sont faiblement couplés avec l'entrée (sont très difficile à contrôler) et demandent une énergie σ_{oi}^{-1} relativement grande pour pouvoir commander l'état dans la direction e_i .

Si l'on observe une énergie σ_{io} à la sortie de l'état e_i , ceci indique que le couplage entre les états et la sortie dans cette direction est très fort (facilement observables).

On recherchera la base de coordonnées dans laquelle les états qui demandent une forte énergie d'entrée (pour être gouvernés), sont les états lents à observer (dans le temps).

Une intéressante interprétation qui en découle des deux grammiens peut être illustrée [14] dans un problème de commande. En effet, l'énergie nécessaire à l'entrée est proportionnelle à $x^T W_g^{-1} x$. Le gramien de gouvernabilité indiquant le degré de gouvernabilité donne une idée sur la notion quantitative du problème de commande. Et si la sortie est à commander alors la quantité $x^T W_o x$ serait la mesure de l'énergie à la sortie du système autonome qui est faible si les états sont difficiles à contrôler donc nécessite une grande énergie d'entrée).

La réalisation (A,B,C) dans la nouvelle base sera dite " réalisation d'équilibre interne " [1].

Rendement énergétique.

Posons:

$$I = \frac{E_0}{E_1} \quad (3.17)$$

Selon Bettayeb [14], ce rapport peut avoir l'interprétation suivante :

- Débutant l'expérience à $t=-\infty$ avec l'état initial noté "état zéro".
- Appliquer la commande $u(t)$ à l'instant $-\infty < t < 0$ pour mener l'état zéro à l'état $x_0 = x(0)$, avec une énergie de commande minimale E_1 :

$$E_1 = x_1^T \cdot W_g^{-1} \cdot x_1 \quad (3.18)$$

-Supprimer la commande à l'instant $t=0$, c-à-d; $u(t)=0$ pour $t>0$. le système évolue donc en régime autonome (libre) avec la capacité x_0 ; il en résulte que l'énergie E_0 disponible à la sortie due à l'état initial x_0 est telle que:

$$E_0 = x_0^T W_o x_0 \quad (3.19)$$

Le rapport I exprimera le rendement qui donne l'énergie emmagasinée.

Une maximisation du rendement I est manifestement plus que souhaitable et elle n'est réalisable que si le système est aussi gouvernable qu'observable.

Le maximum de I est atteint (voir démonstration [A5]) dans la base équilibrée et prend la valeur:

$$I = \sigma_i^2 \quad (3.20)$$

On peut interpréter l'équation (3.19) comme suit :

Si on utilise une unité d'énergie de commande pour guider l'état de zéro vers un autre état x_0 , le maximum d'énergie disponible à la sortie serait de σ_i^2 unités.

On appellera σ_i^2 " les modes du second ordre du système " [13], alors que leurs racines carrées seront dites " valeurs singulières ".

III-3 Algorithme d'Equilibre .

Il est très important de signaler que la procédure d'équilibre est aussi valable en Continu qu'en Discret.

Elle consiste en la détermination de la transformation de la transformatin d'équilibre pour une paire (W_o, W_g) .

Cette procédure nous donne une paire équilibrée (Σ, Σ) qui est dite aussi " rendre symétriques les propriétés d'Entrée-Soties du système ".

La transformation d'équilibre T peut être établie via deux algorithmes, celui de Laub (1980) ou celui de Glover (1984). Pour cette étude, nous avons adopté l'algorithme de Glover. L'autre algorithme donnerait des résultats équivalents.

Algorithme de Glover [24].

Ayant la représentation d'état (A,B,C) , le système équilibré s'obtient en suivant la procédure suivante:

- étape1. - Calculer les gramiens de gouvernabilité et d'observabilité (W_g, W_o) [A3].
- étape2. - Calculer la factorisation de Cholesky [A4] de W_o ,

$$W_o = RR^T$$
 où R est la matrice triangulaire inférieure.
- étape3. - Former le produit $M = R^T W_g R$,
- étape4. - Résoudre le problème des valeurs propres [A1]

$$M = U \Sigma^2 U^T$$

où:

$$\begin{cases} - U: \text{matrice unitaire [A1]}, \\ - \Sigma^2 = \text{diag} \left\{ \sigma_1^2, \sigma_2^2, \dots, \sigma_n^2 \right\} \\ \sigma_i \geq \sigma_{i+1} \geq 0. \end{cases}$$

- étape 5. - Former la transformation d'équilibre T telle que:

$$T = \Sigma^{-1/2} U^T R$$

- étape 6. - Former le système équilibré (A_b, B_b, C_b, D_b) .

où:

$$\begin{cases} - A_b = T^{-1} A T, \\ - B_b = T^{-1} B, \\ - C_b = C T, \\ - D_b = D. \end{cases}$$

Eventuel test de vérification.

-Former les gramiens transformés et vérifier l'égalité:

$$\begin{aligned} W_{ob} &= (C T^T)^{-1} W_o T^{-T} \\ &= \Sigma^{1/2} U^T (R^T)^{-1} R^T R R^{-1} U \Sigma^{1/2} \\ &= \Sigma \\ W_{gb} &= T W_g T^T \\ &= \Sigma^{-1/2} U^T R W_g R^T U \Sigma^{-1/2} \\ &= \Sigma \end{aligned}$$

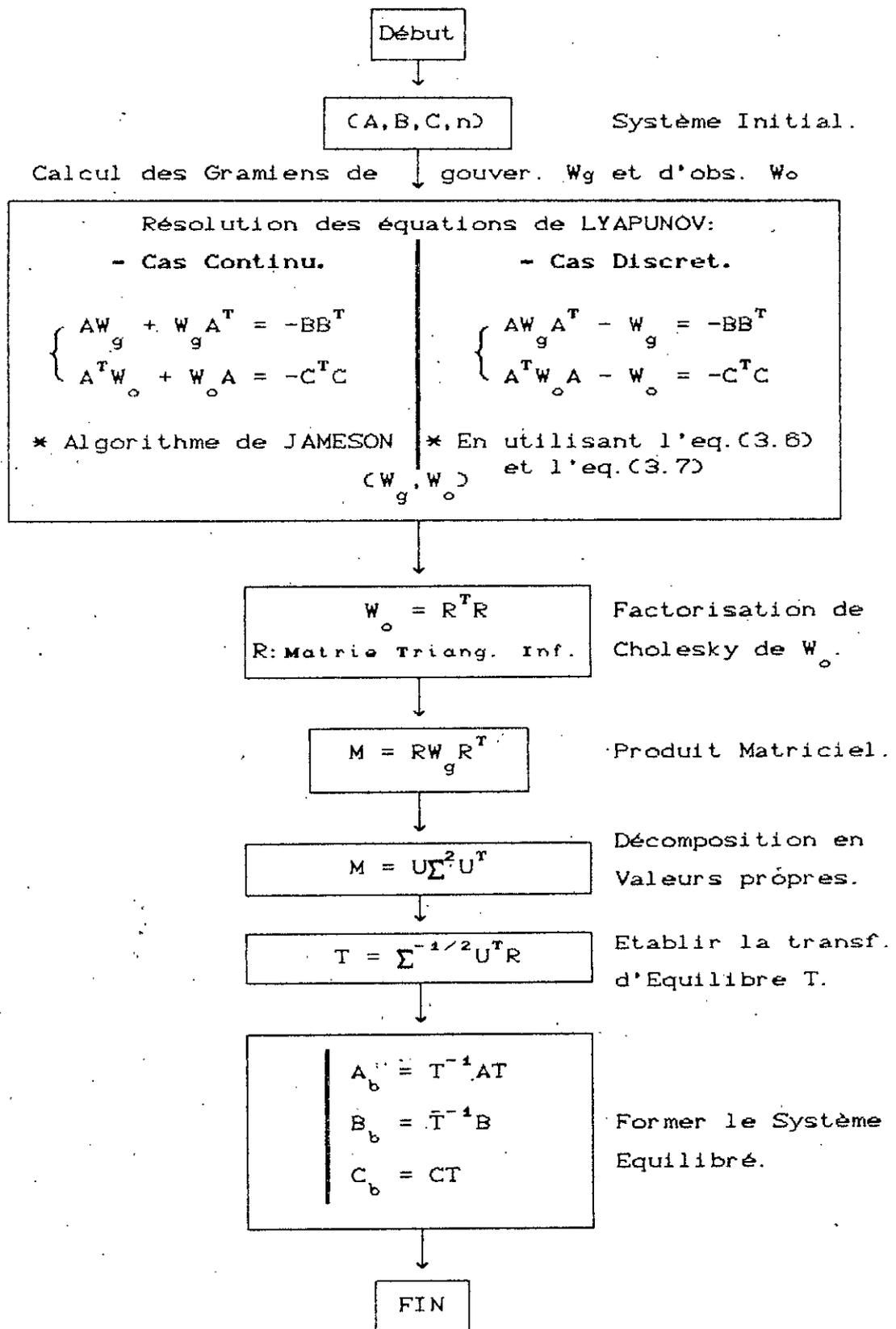
Proposition [16].

Soit \mathcal{P} tel que $\mathcal{P} \subseteq \mathbb{R}(m, n, p)$. \mathcal{P} est la classe de systèmes à paramètres équilibrés

Si (A, B, C) appartient à \mathcal{P} alors (A, B, C) est minimal et A est asymptotiquement stable.

Cette propriété garantie stabilité asymptotique et minimalité de tout triplet équilibré. C'est un grand avantage que nous offre la réalisation équilibrée puisque les autres représentations des systèmes multivariables n'assurent, pas simultanément ces importantes propriétés.

La forme équilibrée toutefois intéressante dans le problème de synthèse des systèmes [17], présente de bonnes propriétés numériques ([9],[17],[18]). La théorie des réalisations équilibrées a été jugée utile dans la réduction de Modèle [1], technique récente qui sera étudiée dans le chapitre suivant.



Organigramme de l'Equilibre des Systèmes.

III.4 Applications.

i/ Exemple d'un système continu (d'ordre 4).

Soit un système continu donné par sa représentation d'état suivante:

- La matrice A initiale:

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & -50.0000 \\ 1.0000 & 0.0000 & 0.0000 & -75.0000 \\ 0.0000 & 1.0000 & 0.0000 & -33.0000 \\ 0.0000 & 0.0000 & 1.0000 & -5.0000 \end{bmatrix}$$

- Le vecteur B initial:

$$B = \begin{bmatrix} 1.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}$$

- Le vecteur C initial:

$$C = \begin{bmatrix} 0.0000 & 1.0000 & 10.0000 & -33.0000 \end{bmatrix}$$

- Son gramien de Gouvernabilité est donné par:

$$W_g = \begin{bmatrix} 0.9973 & 0.3709 & 0.0582 & 0.0100 \\ 0.3709 & 0.1664 & 0.0270 & 0.0049 \\ 0.0582 & 0.0270 & 0.0045 & 0.0008 \\ 0.0100 & 0.0049 & 0.0008 & 0.0002 \end{bmatrix}$$

= Celui d'Observabilité est le suivant:

$$W_o = \begin{bmatrix} 0.4727 & 0.0000 & -2.1909 & 0.5000 \\ -0.0000 & 2.1919 & -0.5000 & -46.1636 \\ -2.1909 & -0.5000 & 36.1636 & -50.0000 \\ 0.5000 & -46.1636 & -50.0000 & 1126.3545 \end{bmatrix}$$

= Le gramien Equilibre Sigma est alors:

$$\Sigma = \begin{bmatrix} 0.5885 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.1452 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0797 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0230 \end{bmatrix}$$

= La transformation d'Equilibre Tb correspondante est:

$$T_b = \begin{bmatrix} -1.2302 & 0.0902 & 0.5424 & 0.2247 \\ -0.4387 & 0.2867 & 0.4575 & 0.6772 \\ -0.0677 & 0.0615 & 0.1053 & 0.0580 \\ -0.0112 & 0.0090 & 0.0305 & 0.0305 \end{bmatrix}$$

= La matrice inverse de Tb est:

$$T_b^{-1} = \begin{bmatrix} -0.8220 & -0.7687 & 1.9234 & 19.4967 \\ -1.1412 & 2.3463 & 18.6474 & -79.2586 \\ 0.2632 & -2.9545 & 5.5764 & 53.1384 \\ -0.2271 & 1.9811 & -10.4158 & 10.2919 \end{bmatrix}$$

- La Matrice A Equilibrée est donnée par:

$$A_b = \begin{bmatrix} -0.5212 & 1.1188 & 0.6186 & 0.3640 \\ -1.9187 & -2.3719 & -8.8584 & -1.6989 \\ 0.2935 & 2.4189 & -0.8158 & -1.1604 \\ -0.3081 & -0.8377 & 2.1042 & -1.2677 \end{bmatrix}$$

- Le vecteur B Equilibré est donné par:

$$B_b = \begin{bmatrix} -0.8220 \\ -1.1412 \\ 0.2632 \\ -0.2335 \end{bmatrix}$$

- Le vecteur C Equilibrée est donnée par:

$$C_b = \begin{bmatrix} -0.7460 & 0.6034 & 0.5038 & 0.2504 \end{bmatrix}$$

2i/ Exemple d'un système discret (Système d'ordre 4).

- La matrice A initiale est:

$$A = \begin{bmatrix} 4.3000e-001 & -2.6140e-001 & 3.8059e-001 & -4.5718e-002 \\ 1.0000e+000 & 0.0000e+000 & 0.0000e+000 & 0.0000e+000 \\ 0.0000e+000 & 1.0000e+000 & 0.0000e+000 & 0.0000e+000 \\ 0.0000e+000 & 0.0000e+000 & 1.0000e+000 & 0.0000e+000 \end{bmatrix}$$

- Le vecteur B initial est:

$$B = \begin{bmatrix} 1.0000e+000 \\ 0.0000e+000 \\ 0.0000e+000 \\ 0.0000e+000 \end{bmatrix}$$

- Le vecteur C initial est:

$$C = \begin{bmatrix} 0.0000e+000 & -2.9500e-001 & 2.5075e-001 & -7.2570e-003 \end{bmatrix}$$

- Le gramien de Gouvernabilite est:

$$W_g = \begin{bmatrix} 1.3144 & 0.4379 & 0.0109 & 0.3704 \\ 0.4379 & 1.3144 & 0.4379 & 0.0109 \\ 0.0109 & 0.4379 & 1.3143 & 0.4379 \\ 0.3704 & 0.0109 & 0.4379 & 1.3143 \end{bmatrix}$$

- Le gramien d'Observabilite est:

$$W_o = \begin{bmatrix} 0.1308 & -0.0893 & -0.0079 & 0.0015 \\ -0.0893 & 0.1834 & -0.0945 & 0.0041 \\ -0.0079 & -0.0945 & 0.0833 & -0.0042 \\ 0.0015 & 0.0041 & -0.0042 & 0.0003 \end{bmatrix}$$

- Le gramien Equilibre Sigma est:

$$\Sigma = \begin{bmatrix} 0.4483 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.3961 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0360 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0077 \end{bmatrix}$$

- La Transformation d'Equilibre Tbal est:

$$T_b = \begin{bmatrix} -0.5427 & -1.4932 & -3.0773 & -0.9145 \\ 1.2034 & -0.4963 & -4.0849 & -1.1091 \\ -0.1815 & 1.0168 & -5.1304 & -0.9184 \\ -0.5515 & -0.0373 & -3.1102 & 8.6242 \end{bmatrix}$$

- La transformation Inverse de Tbal:

$$T_b^{-1} = \begin{bmatrix} -0.3883 & 0.6202 & -0.2671 & 0.0101 \\ -0.3874 & -0.1315 & 0.3496 & -0.0208 \\ -0.0548 & -0.0517 & -0.1065 & -0.0238 \\ -0.0463 & 0.0205 & -0.0540 & 0.1079 \end{bmatrix}$$

- La matrice A équilibrée est:

$$A_b = \begin{bmatrix} -0.1876 & -0.0851 & -0.2324 & -0.4273 \\ 0.1498 & -0.1842 & 0.1829 & 9.8215 \\ 0.3165 & 0.1415 & 0.8716 & -0.3360 \\ -0.5010 & -0.6539 & 0.2892 & -0.0696 \end{bmatrix}$$

- Le vecteur B équilibré est:

$$B_b = \begin{bmatrix} -3.5343 \\ -4.6406 \\ -1.3351 \\ 1.1075 \end{bmatrix}$$

- Le vecteur C équilibré est:

$$C_b = \begin{bmatrix} -0.4569 & 3.4125 & 1.2694 & 1.2232 \end{bmatrix}$$

III.5 Conclusion.

Dans ce chapitre, ont été exposées la notion d'équilibre, une étroite relation liant gramiens et propriétés d'entrées-sorties de point de vue énergétique mise en relief et un algorithme d'équilibre a été établi permettant de former la réalisation équilibrée à partir de laquelle on construira le modèle réduit, technique qui sera détaillée dans le chapitre suivant.

CHAPITRE IV LA REDUCTION

IV-1 Introduction.

Une fois l'opération d'équilibre achevée, ayant obtenu:

- Wg_b : gramien de gouvernabilité équilibré,
- Wo_b : gramien d'observabilité équilibré,
- Avec $Wg_b = Wo_b = \Sigma$

où Σ est définie par:

$$\Sigma = \text{diag} \left\{ \sigma_1, \sigma_2, \dots, \sigma_n \right\} :$$

Σ est aussi appelée matrice des valeurs singulières du système.

La réduction consiste en l'élimination, par troncature, des modes les moins observables et les moins gouvernables du système, selon le critère de la norme de Hankel, en imposant à priori une certaine erreur [1].

Les valeurs singulières du système, autrement dit les éléments de Σ , reflètent les énergies de gouvernabilité et d'observabilité de chaque état du système, le principe de la réduction est donc d'éliminer les valeurs singulières les plus faibles.

IV-2 Critère énergétique de Réduction.

De point de vue mathématique, Σ est aussi solution des équations de Lyapunov [12] (que ce soit en continu ou en discret).

$$\begin{cases} A_b \Sigma + \Sigma A_b^T = -B_b B_b^T \\ A_b^T \Sigma + \Sigma A_b = -C_b^T C_b \end{cases} \text{ ou bien } \begin{cases} A_b \Sigma A_b^T - \Sigma = -B_b^T B_b \\ A_b^T \Sigma A_b - \Sigma = -C_b^T C_b \end{cases} \quad (4.1)$$

où:

- $A_b = T_b^{-1} A T_b$,
- $B_b = T_b^{-1} B$,
- $C_b = C T_b$.

Dans un but de simplifier la réduction, la matrice Σ doit être partitionnée comme suit:

$$\Sigma = \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array} \right] \quad \text{où: } \begin{cases} - \Sigma_1 = \text{diag} \{ \sigma_1, \dots, \sigma_k \} \\ - \Sigma_2 = \text{diag} \{ \sigma_{k+1}, \dots, \sigma_n \} \end{cases} \quad (4.2)$$

Les éléments de Σ doivent être ordonnés en décroissance, c-à-d; ($\sigma_1 \geq \sigma_2 \dots \geq \sigma_n \geq 0$).

Le classement est établi pour garder les valeurs singulières les plus importantes tout en éliminant les plus faibles. Dans ce processus, la réduction s'opère sur une base énergétique.

IV-3 Principe de Réduction.

Le problème qui se pose est de trouver un ordre (k) de troncature qui donnerait sous une erreur imposée à priori, un sous-système optimal pour cet ordre.

Le critère de réduction utilisé est celui de "la norme de Hankel":

$$\left[\sum_{i=1}^k \sigma_i^4 \right]^{1/2} \gg \left[\sum_{i=k+1}^n \sigma_i^4 \right]^{1/2} \quad \text{avec } \begin{cases} k \neq 0 \\ k \neq n \end{cases} \quad (4.3)$$

où:

$$\varepsilon_k = \frac{\left[\sum_{i=k+1}^n \sigma_i^4 \right]^{1/2}}{\left[\sum_{i=1}^k \sigma_i^4 \right]^{1/2}} \ll 1 \quad \text{avec } (k=0, n) \quad (4.4)$$

Par souci d'efficacité, on introduit une autre erreur ξ telle que:

$$\xi_i = | \varepsilon_i - \varepsilon_{i+1} | \leq \xi \quad \text{avec } (i=1, 2, \dots, n-1). \quad (4.5)$$

où ξ est l'erreur fixée.

Le système complet est donc réorganisé avec une transformation interne de coordonnées, [1] ce qui introduit la notion de " Sous-système dominant ".

En introduisant la matrice Σ partitionnée dans les équations de Lyapunov, on aura:

$$\left[\begin{array}{c|c} \left[\begin{array}{c|c} A_{b11} & A_{b12} \\ \hline A_{b21} & A_{b22} \end{array} \right] \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array} \right] + \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array} \right] \left[\begin{array}{c|c} A_{b11}^T & A_{b12}^T \\ \hline A_{b21}^T & A_{b22}^T \end{array} \right] = - \left[\begin{array}{c} B_{b1} \\ B_{b2} \end{array} \right] \left[\begin{array}{c|c} B_{b1}^T & B_{b2}^T \end{array} \right] \\ \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array} \right] \left[\begin{array}{c|c} A_{b11} & A_{b12} \\ \hline A_{b21} & A_{b22} \end{array} \right] + \left[\begin{array}{c|c} A_{b11}^T & A_{b12}^T \\ \hline A_{b21}^T & A_{b22}^T \end{array} \right] \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array} \right] = - \left[\begin{array}{c} C_{b1}^T \\ C_{b2}^T \end{array} \right] \left[\begin{array}{c|c} C_{b1} & C_{b2} \end{array} \right] \end{array} \right. \quad (4.6)$$

En posant:

$$\begin{array}{l} A_r = A_{b11} \\ B_r = B_{b1} \\ C_r = C_{b1} \\ \Sigma_r = \Sigma_1 \end{array}$$

Autrement dit; le système réorganisé peut se partitionné comme suit:

$$\left[\begin{array}{c} \frac{dx_1(t)}{dt} \\ \frac{dx_2(t)}{dt} \end{array} \right] = \left[\begin{array}{c|c} A_r & A_{b11} \\ \hline A_{b21} & A_{b22} \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \end{array} \right] + \left[\begin{array}{c} B_r \\ B_2 \end{array} \right] \underline{u}(t) \quad (4.7)$$

$$y(t) = \left[\begin{array}{c|c} C_r & C_{b2} \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \end{array} \right]$$

où: (A_r, B_r, C_r, k) est la partie la plus gouvernable et la plus observable du système complet (A, B, C, n) .

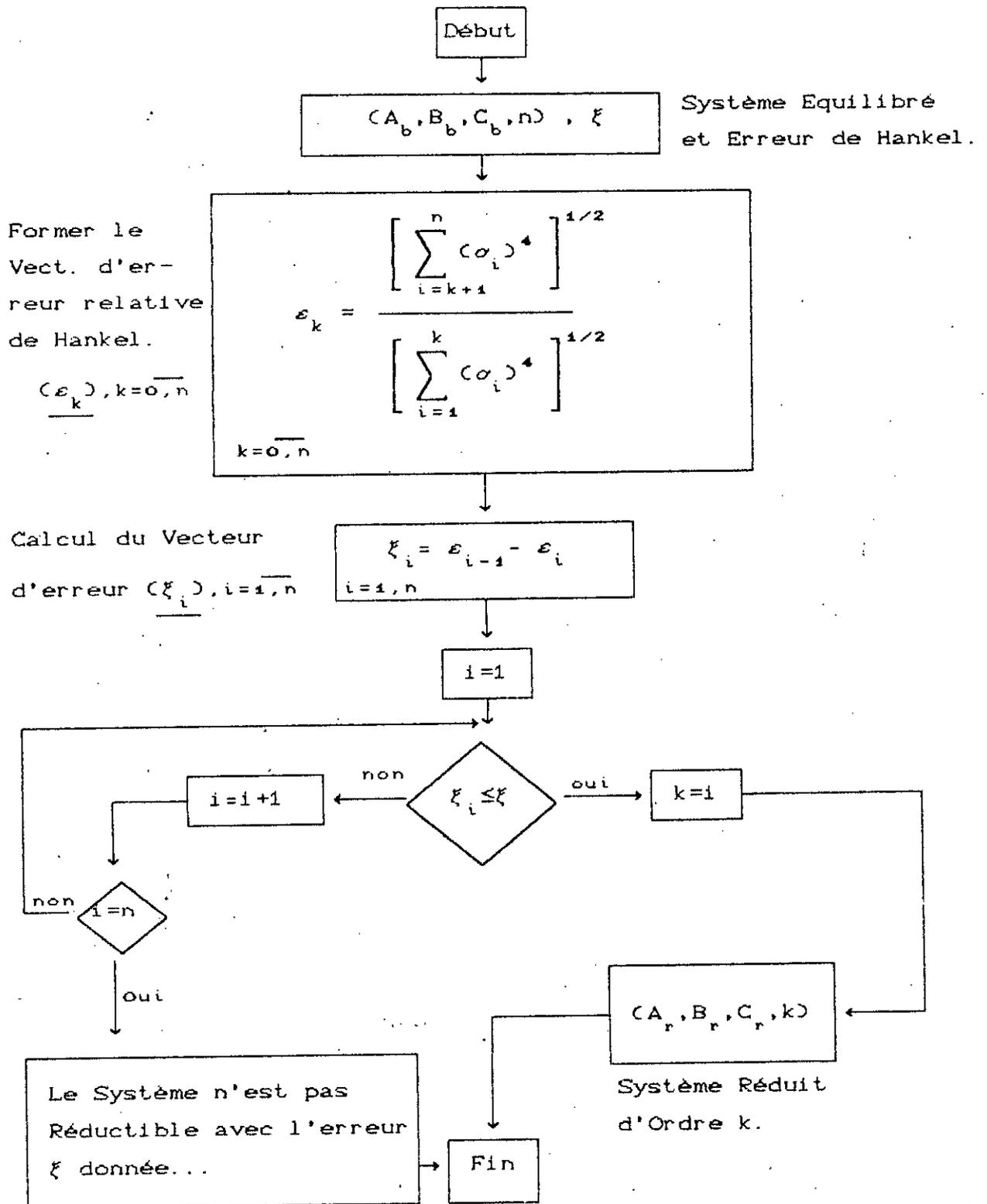
Il reste très important de signaler que la réduction par troncature dans la base d'équilibre préserve la signification physique des états du système réduit.

Définition [1].

Le système réduit (A_r, B_r, C_r) est un système à dominance interne si et seulement le modèle complet dans le système de coordonnées équilibré est organisé de telle façon qu'il vérifie la relation (4.3).

IV-3 Propriétés du système réduit.

Tout système réduit (A_r, B_r, C_r) engendré par un système équilibré asymptotiquement stable, satisfera lui aussi les deux équations de Lyapunov, cette importante remarque induit la conservation des propriétés telle la stabilité du système initial [1], [9], [10].



Organigramme de Réduction.

IV-5 Conclusion.

Dans ce chapitre la technique de réduction de modèles a été étudiée. Un des problèmes essentiels au niveau des applications reste sans doute le mal conditionnement de l'équilibre. En effet, la procédure de réduction du modèle de Moore de point de vue utilisation s'est avérée limitée, puisqu'elle présente des difficultés numériques (un calcul compliqué est demandé pour construire la transformation d'équilibre) et une sensibilité aux erreurs numériques (Inversion matricielle, problème d'arrondi...). Les difficultés numériques associés à T_b ont été contournées par l'utilisation d'algorithmes d'équilibres plus performants tel que l'algorithme de Laub [12], dont l'algorithme consiste à calculer directement la transformation d'équilibre T_b et son inverse et les utilise pour construire le modèle réduit. Cependant; un sérieux problème se pose, lorsque les systèmes possèdent des modes presque ingouvernables et presque inobservables (valeurs singulières très faibles); dans ce cas, l'équilibre devient intrinsèquement mal conditionné.

La transformation d'équilibre peut avoir un mauvais conditionnement quand les modes sont presque ingouvernables et inobservables [21], autrement dit, il y a tendance de la matrice T_b vers une singularité [A1] lorsque les modes du système sont presque inobservables et/ou ingouvernables. Un autre problème surgit lorsque les valeurs singulières du système sont de même ordre de grandeur, dans ce cas la réduction s'avère très difficile. C'est pourquoi, il serait préférable de générer une réalisation pour la fonction de transfert de modèle réduit de Moore:

$$G(s) = C_1 (sI - A_{11})^{-1} B_1 + D \quad (4.8)$$

sans avoir du tout à équilibrer et ce via des "Projections dans l'espace d'état " ([20], [21]).

Se basant sur le principe des projections, de nouvelles techniques de réduction de modèles ont été élaborées, on distinguera notamment le travail de Derras [20] dans lequel le modèle réduit obtenu est optimal et stable.

L'approche de Safonov et Chiang [21] présente le grand avantage d'un algorithme insensible à la tendance des modes des systèmes vers l'ingouvernabilité et/ou la non observabilité.

La performance de la M.R.E sera illustrée et discutée par quelques applications qui feront l'objet du chapitre suivant.

CHAPITRE V

RESULTATS ET INTERPRETATIONS

V.1 INTRODUCTION

L'objet de ce chapitre est l'évaluation des performances de la MRE. On propose quatre exemples de réduction (deux en continu, deux en discret). Pour chaque cas, on tracera quelques fonctions essentielles (réponse impulsionnelle, spectre d'amplitude, spectre de phase) ainsi que le plan de stabilité, cela pour le système initial et celui du réduit. La performance de la réduction sera reflétée par l'évaluation de l'erreur relative de Hankel.

V.2 Cas Discret.

Exemple 1.

Soit un filtre de Butherworth numérique d'ordre 15 [24] obeissant au gabarit suivant:

- Fréquence de coupure normalisée : $f_c = 0.25$
- Fréquence d'atténuation normalisée : $f_r = 0.258$
- Atténuation dans la B.P : $A = 0.1$

- Le système initial est donné par la réalisation (A,B,C), soient:

* La matrice A initiale :

$$A = \begin{bmatrix} 0.0000 & -2.0320 & 0.0000 & -1.5146 & 0.0000 \\ 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

0.5231	0.0000	-8.7E-2	0.0000	-6.7E-3
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000	0.0000
0.0000	0.0000	0.0000	1.0000	0.0000
0.0000	0.0000	0.0000	0.0000	1.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000

0.0000	-2.0E-4	0.0000	-1.0E-6	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	1.0000	0.0000

* Les vecteurs B et C correspondants sont:

$$B = \begin{bmatrix} 1.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}$$

$$C^T = \begin{bmatrix} 2.3640 \\ 1.6225 \\ 7.1701 \\ 0.2149 \\ 0.4723 \\ 0.7886 \\ 1.0141 \\ 1.0140 \\ 0.8872 \\ 0.4732 \\ 0.2151 \\ 7.170E-2 \\ 1.654E-2 \\ 2.364E-3 \\ 1.580E-4 \end{bmatrix}$$

- Les Pôles (valeurs propres de A) du système initial sont :

Partie réelle	Partie Imaginaire
0.0000	0.9004
0.0000	0.7265
0.0000	0.5773
0.0000	0.4452
0.0000	0.3249
0.0000	0.2125
0.0000	0.1051
0.0000	0.0000
0.0000	-0.1051
0.0000	-0.2125
0.0000	-0.3249
0.0000	-0.4452
0.0000	-0.5773
0.0000	-0.7265
0.0000	-0.9004

- Le garmien équilibré est alors :

$$\sum = \left\{ 9.9925E-1, 9.8575E-1, 8.8784E-1, 5.9307E-1, 2.3083E-1, \right. \\ 4.6507E-2, 5.1157E-3, 3.4328E-4, 1.4933E-5, 4.1031E-7, \\ \left. 6.6179E-9, 6.2207E-11, 3.1956E-13, 5.4555E-18, -4.6013E-16 \right\}$$

- Le système équilibré est donné par (Ab, Bb, Cb) :

* La matrice équilibrée A est :

$A_b =$

3.2415E-001	-5.9614E-001	-1.5923E-001	-4.1034E-001	5.4541E-002
5.7593E-001	5.2450E-001	4.9077E-001	-1.1045E-001	6.0391E-002
-1.7654E-001	-5.6321E-001	8.1604E-001	7.6023E-002	-2.6765E-002
8.8774E-001	-2.4732E-001	-1.4834E-001	1.5605E-001	-4.1942E-001
2.3619E-001	-2.7070E-001	-1.0454E-001	8.3957E-001	2.6867E-002
-2.4250E-001	1.0198E-001	5.5713E-002	-7.6095E-002	-5.7587E-001
2.2801E-002	-2.3018E-002	-9.2801E-003	5.3225E-002	1.0723E-002
1.3932E-002	-6.4479E-003	-3.3877E-003	6.1686E-003	2.6458E-002
2.2689E-003	-1.9667E-003	-8.2634E-004	4.1620E-003	1.8979E-003
6.9302E-004	3.6571E-004	1.8180E-004	-4.6222E-004	-1.1932E-003
-9.5805E-005	7.0130E-005	3.0971E-005	-1.3139E-004	-1.1314E-004
-1.5219E-005	9.0753E-006	4.3155E-006	-1.3719E-005	-2.3641E-005
-1.5427E-006	1.0390E-006	4.7952E-007	-1.7829E-006	-2.1941E-006
-3.6835E-007	2.3922E-007	1.1362E-007	-3.9025E-007	-5.7773E-007
4.6762E-009	-2.9488E-009	-1.3416E-009	4.8579E-009	6.2852E-009

5. 2748E-002	2. 2569E-002	-1. 0570E-002	1. 6185E-000	4. 7654E-004
2. 1432E-002	2. 2011E-002	-4. 7262E-003	1. 3554E-003	2. 4295E-004
-1. 3436E-002	-1. 0184E-002	2. 8498E-003	-6. 5356E-004	-1. 3860E-004
-3. 5809E-002	-1. 1397E-001	1. 0125E-002	-6. 4230E-003	-6. 8763E-004
5. 4246E-001	4. 5964E-002	-8. 6931E-002	5. 8629E-003	3. 5532E-003
-5. 5980E-002	1. 0970E+000	-3. 1533E-002	5. 3253E-002	3. 6997E-003
-2. 4107E-001	-1. 0328E-001	-4. 1193E-001	1. 5420E-002	1. 5243E-002
-9. 0402E-003	5. 3740E-001	-1. 3610E-001	-4. 1597E-001	-1. 8925E-002
-1. 6238E-002	2. 1397E-002	4. 4242E-001	-1. 6653E-001	3. 7785E-001
1. 1703E-003	-2. 1942E-002	-2. 0878E-002	-3. 9198E-001	-1. 9314E-001
4. 7711E-004	-1. 7383E-003	-1. 1932E-002	-2. 0787E-002	3. 4662E-001
4. 2567E-005	-4. 1562E-004	-9. 3489E-004	-6. 7183E-003	1. 7951E-002
6. 2586E-006	-3. 5874E-005	-1. 4654E-004	-5. 1471E-004	3. 7563E-003
1. 3586E-006	-9. 5880E-006	-3. 0015E-005	-1. 4486E-004	7. 0553E-004
-1. 5536E-008	1. 0914E-007	3. 7745E-007	1. 6656E-006	-8. 9854E-006

-6. 2517E-005	9. 7022E-006	-9. 4536E-007	2. 4932E-008	2. 3949E-008
-4. 4212E-005	5. 5898E-006	-6. 1782E-007	1. 6236E-008	1. 2687E-008
2. 2411E-005	-3. 0499E-006	3. 2186E-007	-7. 7609E-009	-9. 0668E-009
1. 8551E-004	-1. 8931E-005	2. 3673E-006	-5. 9134E-008	-5. 3519E-008
-3. 1969E-004	6. 5206E-005	-5. 8378E-006	1. 8086E-007	1. 1271E-007
-1. 4314E-003	1. 2494E-004	-1. 7745E-005	4. 3535E-007	3. 0931E-007
-1. 1459E-003	2. 6755E-004	-2. 2107E-005	6. 5563E-007	5. 8506E-007
1. 0265E-002	-7. 8609E-004	1. 1873E-004	-2. 9197E-006	-2. 4309E-006
-1. 9001E-002	5. 9943E-003	-4. 4603E-004	1. 4751E-005	6. 7402E-006
-3. 2899E-001	1. 6741E-002	-3. 3711E-003	6. 5515E-005	8. 6360E-005
-2. 0692E-001	-2. 9204E-001	1. 3775E-002	-4. 6108E-004	-8. 0037E-004
2. 9912E-001	-2. 2327E-001	-2. 4903E-001	4. 9147E-003	-4. 7065E-005
1. 5183E-002	2. 5626E-001	-2. 5182E-001	-6. 5515E-002	1. 2221E-001
5. 4228E-003	3. 3882E-002	5. 5057E-001	-2. 9775E-001	7. 3475E-001
-5. 8409E-005	-4. 8767E-004	-4. 7937E-003	-2. 8054E-002	-2. 1289E-001

* Les vecteurs B et C équilibrés sont :

$$B_b = \begin{bmatrix} 3.3912E-001 \\ 2.3330E-001 \\ -1.2554E-001 \\ -5.1837E-001 \\ 5.4533E-001 \\ 2.8759E-001 \\ 4.7863E-002 \\ -1.4246E-002 \\ 2.9305E-003 \\ 4.4937E-004 \\ -5.0261E-005 \\ 3.9886E-006 \\ -2.1431E-007 \\ 1.7182E-008 \\ 1.4289E-011 \end{bmatrix}, \quad C_b^T = \begin{bmatrix} 5.4479E-001 \\ -3.8794E-001 \\ -1.8190E-001 \\ 3.8493E-001 \\ 2.0230E-001 \\ -1.0049E-001 \\ 7.6106E-002 \\ 1.7363E-002 \\ 3.3584E-003 \\ -4.9639E-004 \\ -5.2714E-005 \\ -4.0647E-006 \\ -2.1557E-007 \\ -7.5486E-010 \\ 4.4502E-009 \end{bmatrix}$$

- L'erreur relative de Hankel est donnée dans le tableau suivant :

Ordre du système réduit	Erreur de Hankel
1	1.4732
2	0.7790
3	0.3842
4	0.1335
5	0.0263
6	0.0029
7	0.0002
8	0.0000
9	0.0000
10	0.0000
11	0.0000
12	0.0000
13	0.0000
14	0.0000

* La matrice réduite A_r est telle que : ($\epsilon = 10^{-3}$)

$$A_r = \begin{bmatrix} 3.2415E-001 & -5.9614E-001 & -1.5923E-001 & -4.1034E-001 \\ 5.7593E-001 & 5.2450E-001 & 4.9077E-001 & -1.1045E-001 \\ -1.7654E-001 & -5.6321E-001 & 8.1604E-001 & 7.6023E-002 \\ 8.8774E-001 & -2.4732E-001 & -1.4834E-001 & 1.5605E-001 \\ 2.3619E-001 & -2.7070E-001 & -1.0454E-001 & 8.3957E-001 \\ -2.4250E-001 & 1.0198E-001 & 5.5713E-002 & -7.6095E-002 \\ 2.2801E-002 & -2.3018E-002 & -9.2801E-003 & 5.3225E-002 \end{bmatrix}$$

5.4541E-002	5.2748E-002	2.2569E-002
6.0391E-002	2.1432E-002	2.2011E-002
-2.6765E-002	1.3436E-002	-1.0184E-002
-4.1942E-001	3.5809E-002	-1.1397E-001
2.6867E-002	5.4246E-001	4.5964E-002
-5.7587E-001	5.5980E-002	1.0970E+000
1.0723E-002	2.4107E-001	-1.0328E-001

* Les vecteurs réduits B_r et C_r correspondants sont :

$B_r =$	3.39117956727E-0001	$C_r =$	5.44788852390E-0001
	2.33299828030E-0001		-3.87941875023E-0001
	-1.25537420998E-0001		-1.81897135648E-0001
	-5.18366780371E-0001		3.84925941107E-0001
	5.45326726685E-0001		2.02295071600E-0001
	2.87586848749E-0001		-1.00494190083E-0001
	4.78626037118E-0002		7.61062837009E-0002

= Les pôles du système réduit sont les suivants:

Partie Réelle	Partie Imaginaire
1.2630304339395E-002	9.2111710320180E-001
1.2630304339395E-002	-9.2111710320180E-001
1.6688001286647E-001	7.7161899517021E-001
1.6688001286647E-001	-7.7161899517021E-001
5.2385526797643E-001	0.00000000000000
4.0273554880592E-001	4.7401185927790E-001
4.0273554880592E-001	-4.7401185927790E-001

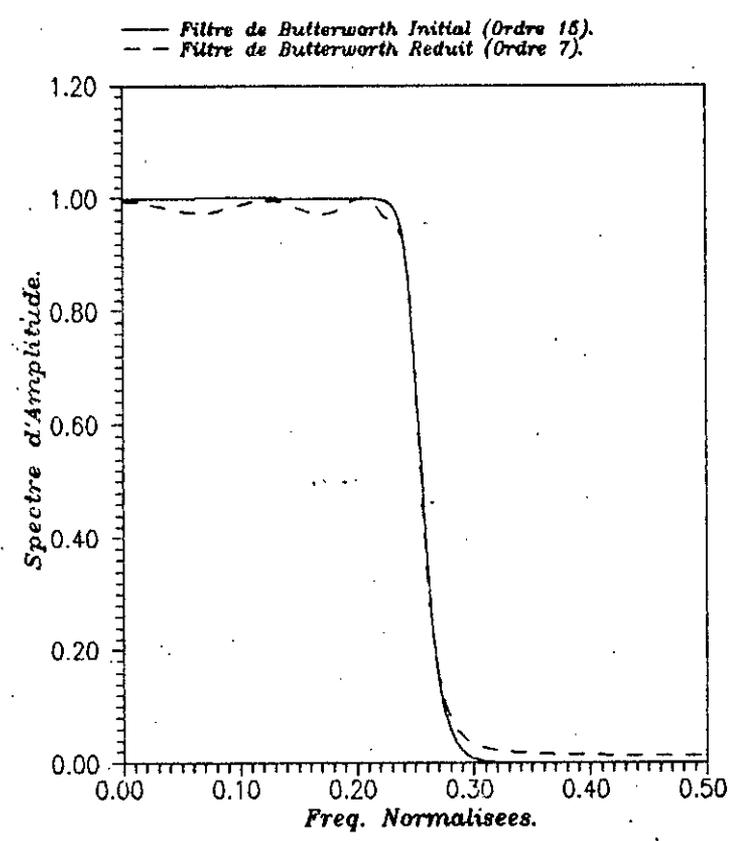
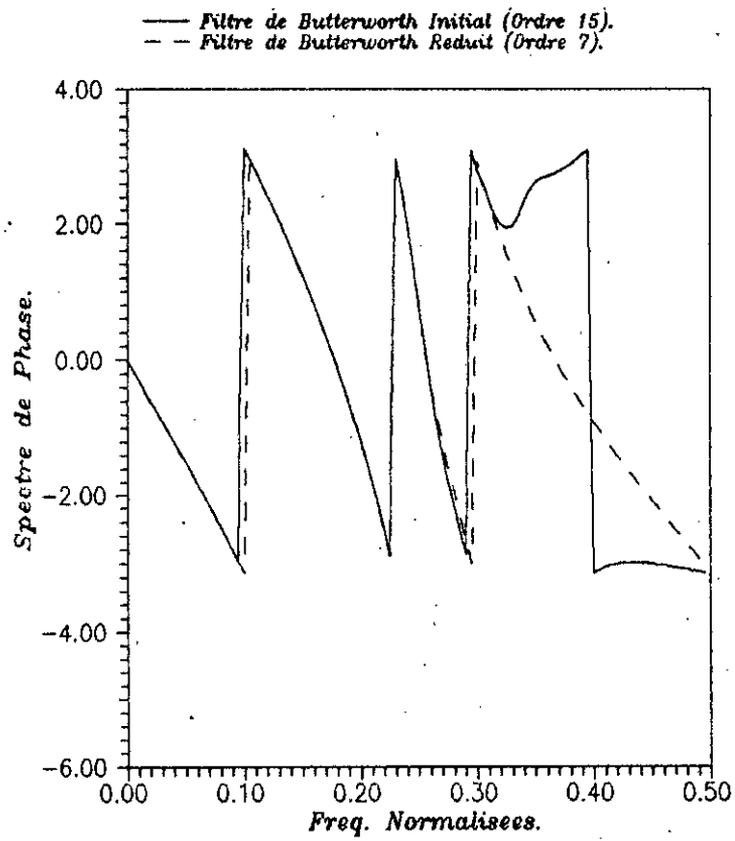


Fig. 5.1.a- Spectres d'amplitude et de phase.

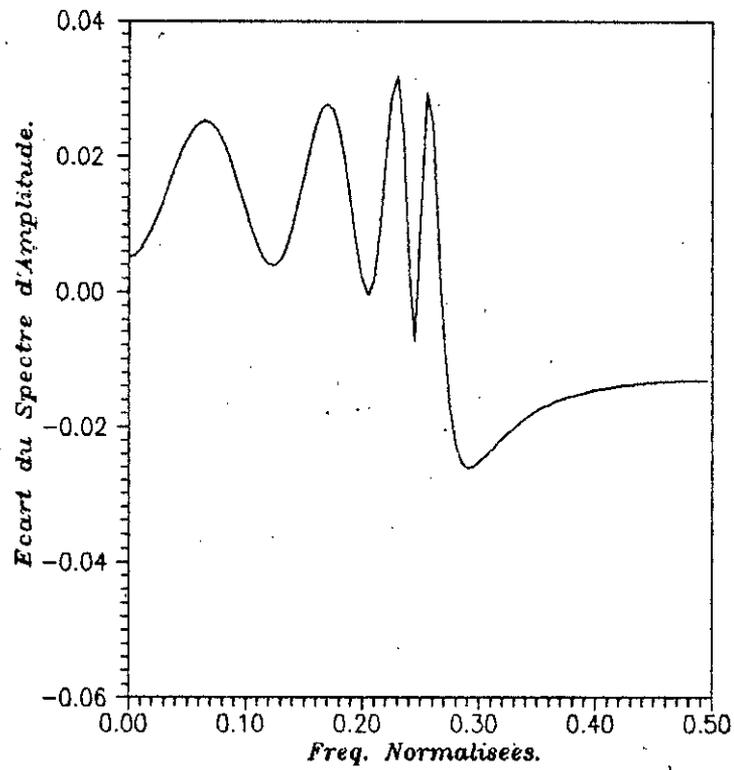
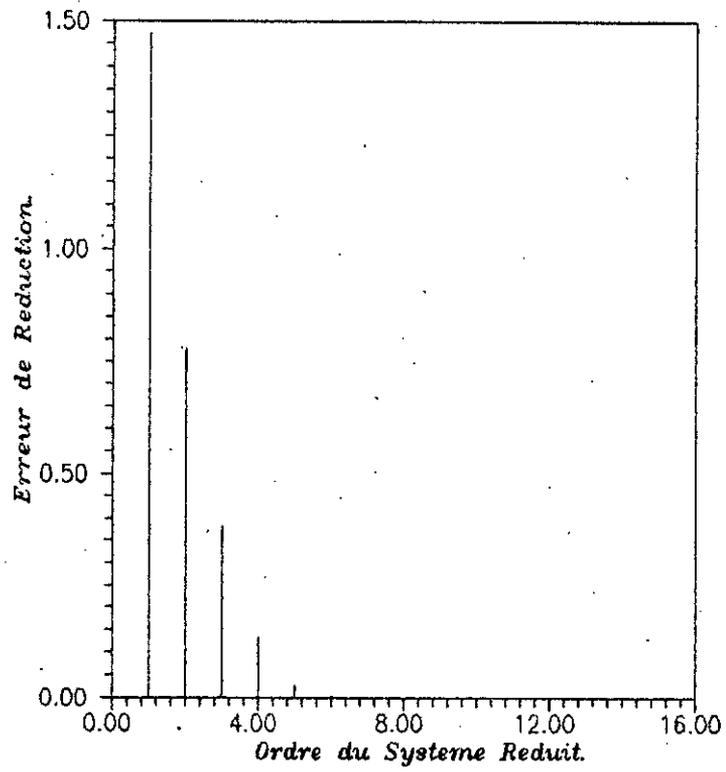


Fig. 5.1.b- Erreur de reduction et ecart du S.A.

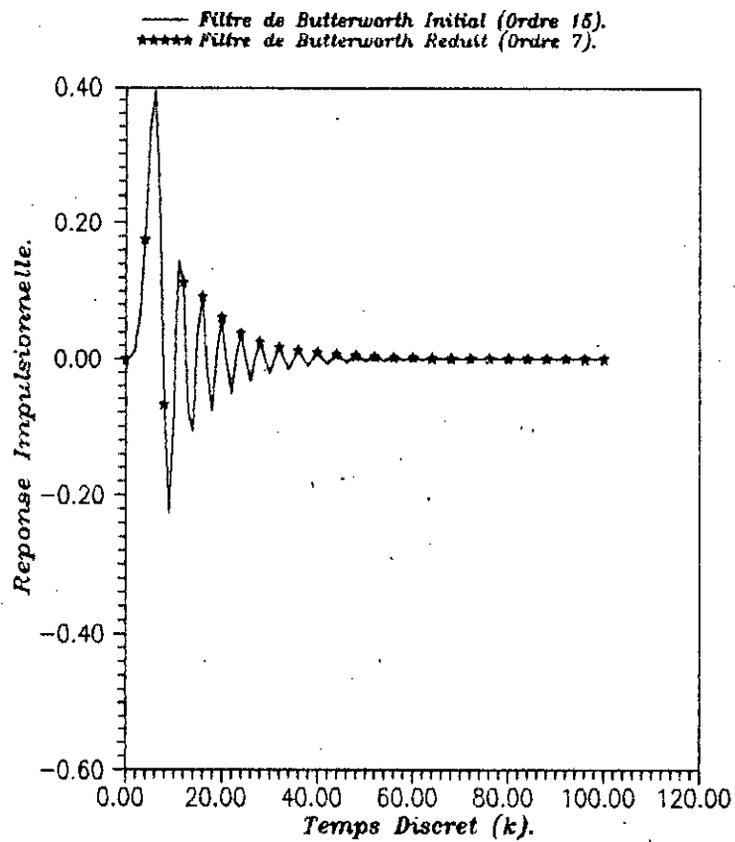
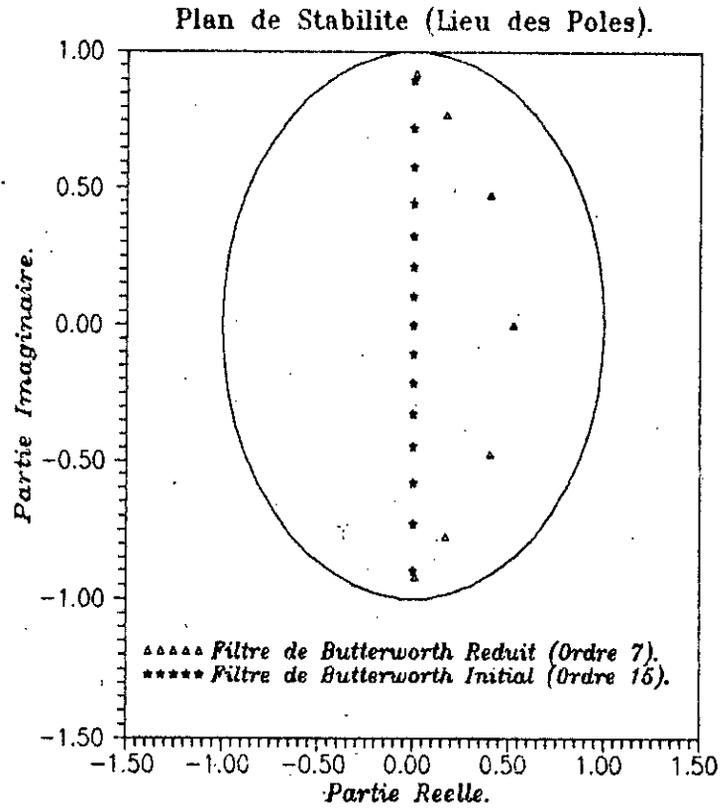


Fig. 5.1.c- Réponse Imp. et plan de stabilité.

Exemple2. Soit le système discret d'ordre huit [7].

- Le système initial est donné par la réalisation (A,B,C) suivante:

* La matrice A initiale:

$$A = \begin{bmatrix} -3.0000E-002 & -3.9970E-001 & 1.1647E-002 & 3.7524E-003 \\ 1.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 1.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 1.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 1.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \end{bmatrix}$$

$$\begin{bmatrix} 1.2690E-002 & 2.9473E-001 & 1.1200E-001 & 1.7601E-001 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 1.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 1.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 1.0000E+000 & 0.0000E+000 \end{bmatrix}$$

- Les vecteurs B et C initiaux sont respectivement :

$$B = \begin{bmatrix} 1.0000E+000 \\ 0.0000E+000 \end{bmatrix} \quad C^T = \begin{bmatrix} 0.0000E+000 \\ 1.0000E-003 \\ -2.5200E-003 \\ 3.7503E-003 \\ -3.6972E-003 \\ 2.5932E-003 \\ -8.2177E-004 \\ -3.0453E-004 \end{bmatrix}$$

- Les pôles (valeurs propres de A) du système initial sont :

Partie réelle	Partie Imaginaire
-4.200000E-001	6.400000E-001
-4.200000E-001	-6.400000E-001
4.700000E-001	7.300000E-001
4.700000E-001	-7.300000E-001
-1.000000E-001	5.700000E-001
-1.000000E-001	-5.700000E-001
8.700000E-001	0.000000E+000
-8.000000E-001	0.000000E+000

- Le gramien équilibré du système considéré est :

$$\Sigma = \text{diag} \left\{ 9.1405E-05, 3.6792E-05, 1.2740E-05, 1.3826E-06, \right. \\ \left. 7.8147E-07, 2.9769E-07, 2.1816E-007, 1.6853E-008 \right\}$$

- Le système équilibré est représenté par la réalisation (Ab, Bb, Cb), soient :

* La matrice A équilibrée :

$$A_b = \begin{bmatrix} -8.6076E-01 & -3.3499E-01 & -3.8137E-02 & -4.4580E-02 \\ 3.9006E-01 & -4.6164E-01 & -7.1605E-01 & 2.4302E-02 \\ -3.4063E-02 & 5.4927E-01 & -3.3767E-01 & -2.6469E-01 \\ 6.6531E-02 & 3.1148E-02 & 4.4227E-01 & 7.1062E-02 \\ 4.8472E-03 & -3.8381E-02 & 1.0744E-02 & -5.3383E-01 \\ -1.8096E-02 & 5.1265E-03 & -1.0409E-01 & 9.2565E-02 \\ 2.4361E-02 & 3.9911E-03 & 1.4614E-01 & -5.3190E-02 \\ -3.5320E-03 & -5.0968E-03 & -2.2927E-02 & -2.0556E-02 \end{bmatrix}$$

$$\begin{bmatrix} 6.7421E-03 & -4.1270E-02 & -1.7402E-02 & -5.6366E-03 \\ 6.2162E-02 & -1.3613E-02 & 3.3197E-03 & 9.4708E-03 \\ 1.3348E-02 & -2.1202E-01 & -9.3241E-02 & -3.2679E-02 \\ 1.1081E+00 & -3.1503E-01 & -5.6704E-02 & 4.8959E-02 \\ 1.8865E-01 & 4.8649E-01 & 2.5276E-01 & 6.0290E-02 \\ 2.9672E-01 & -1.1642E-01 & 3.2233E-01 & -1.6024E-01 \\ 4.9216E-01 & -1.0290E+00 & 6.0851E-01 & 1.1461E-01 \\ 5.2547E-02 & -2.2899E-01 & -5.1302E-02 & 8.7828E-01 \end{bmatrix}$$

- Les vecteurs B et C équilibrés:

$$B_b = \begin{bmatrix} -4.3157E-02 \\ -5.2341E-02 \\ -2.4580E-02 \\ -1.7317E-02 \\ 2.1811E-03 \\ 3.6123E-03 \\ -5.7970E-03 \\ 1.2352E-03 \end{bmatrix} \quad C_b = \begin{bmatrix} 3.8827E-02 \\ -4.0442E-02 \\ 2.4759E-02 \\ -1.0439E-02 \\ -2.7293E-03 \\ -7.4114E-03 \\ -3.7256E-03 \\ -1.7735E-03 \end{bmatrix}$$

- L'erreur de Hankel (en fonction de l'ordre) est:

Ordre du système réduit	Erreur de Hankel
1	0.4283
2	0.1304
3	0.0164
4	0.0087
5	0.0037
6	0.0022
7	0.0002

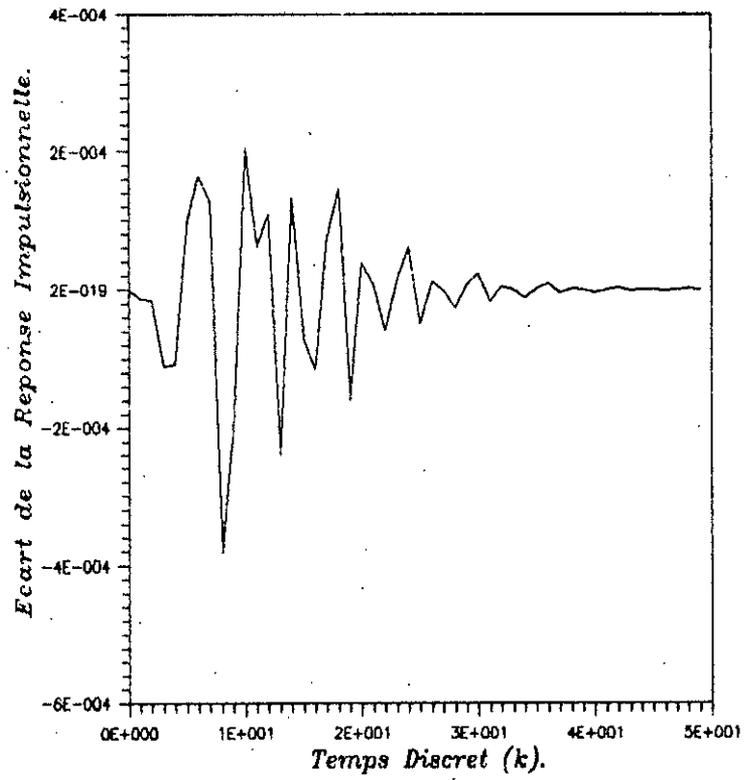
- Le système réduit est donné par la réalisation (A_r,B_r,C_r), soient: ($\zeta = 5 \cdot 10^{-3}$).

* La matrice A réduite :

$$A_r = \begin{bmatrix} -8.6076E-001 & -3.3500E-001 & -3.8138E-002 & -4.4580E-002 \\ 3.9007E-001 & -4.6164E-001 & -7.1606E-001 & 2.4303E-002 \\ -3.4064E-002 & 5.4927E-001 & -3.3768E-001 & -2.6470E-001 \\ 6.6531E-002 & 3.1149E-002 & 4.4227E-001 & 7.1062E-002 \end{bmatrix}$$

* Le vecteur B réduit :

$$B_r = \begin{bmatrix} -4.3157E-02 \\ -5.2341E-02 \\ -2.4580E-02 \\ -1.7317E-02 \end{bmatrix}$$



— Systeme Initial (Ordre 8).
 - - - Systeme Reduit (Ordre 4).

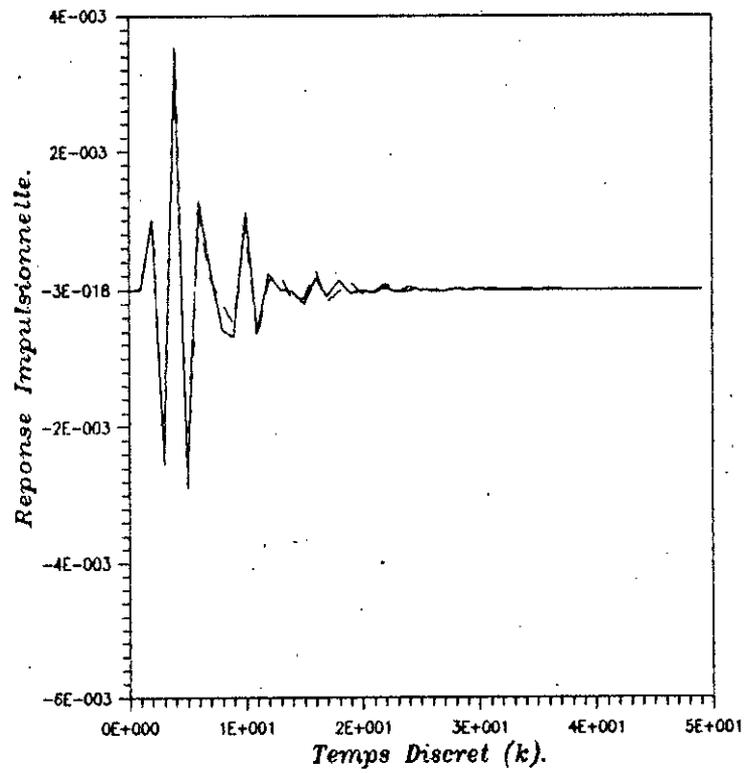


Fig. 5.2. a- Réponse Imp. et son ecart.

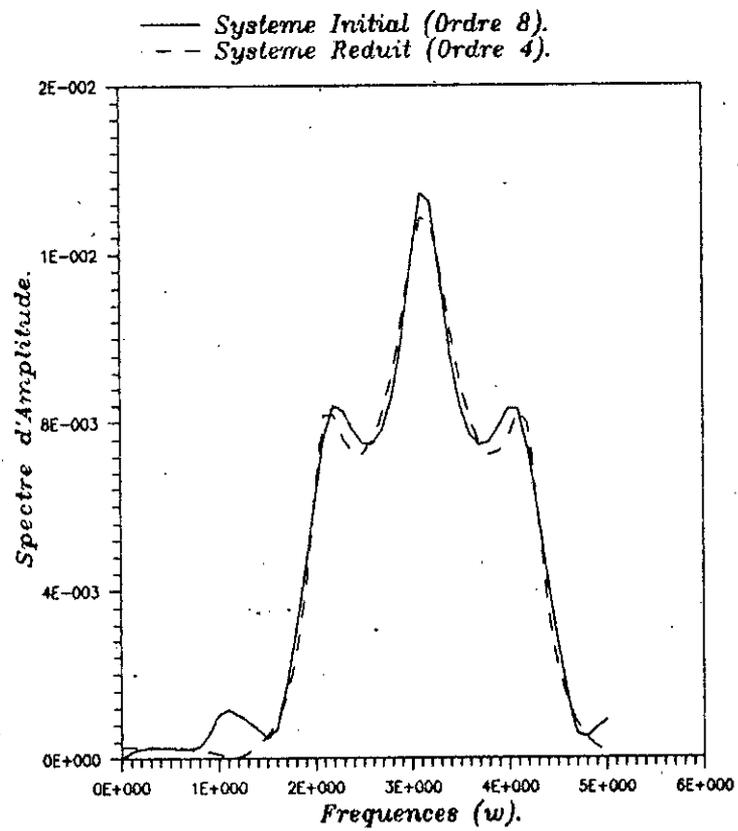
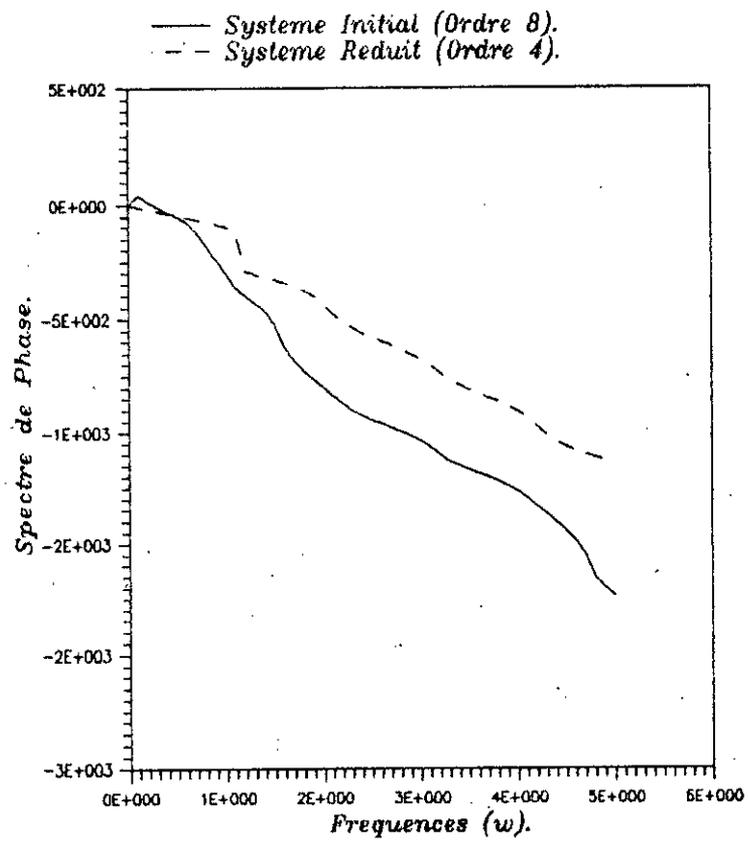


Fig. 5.2.b- Spectres d'Amplitude et de Phase.

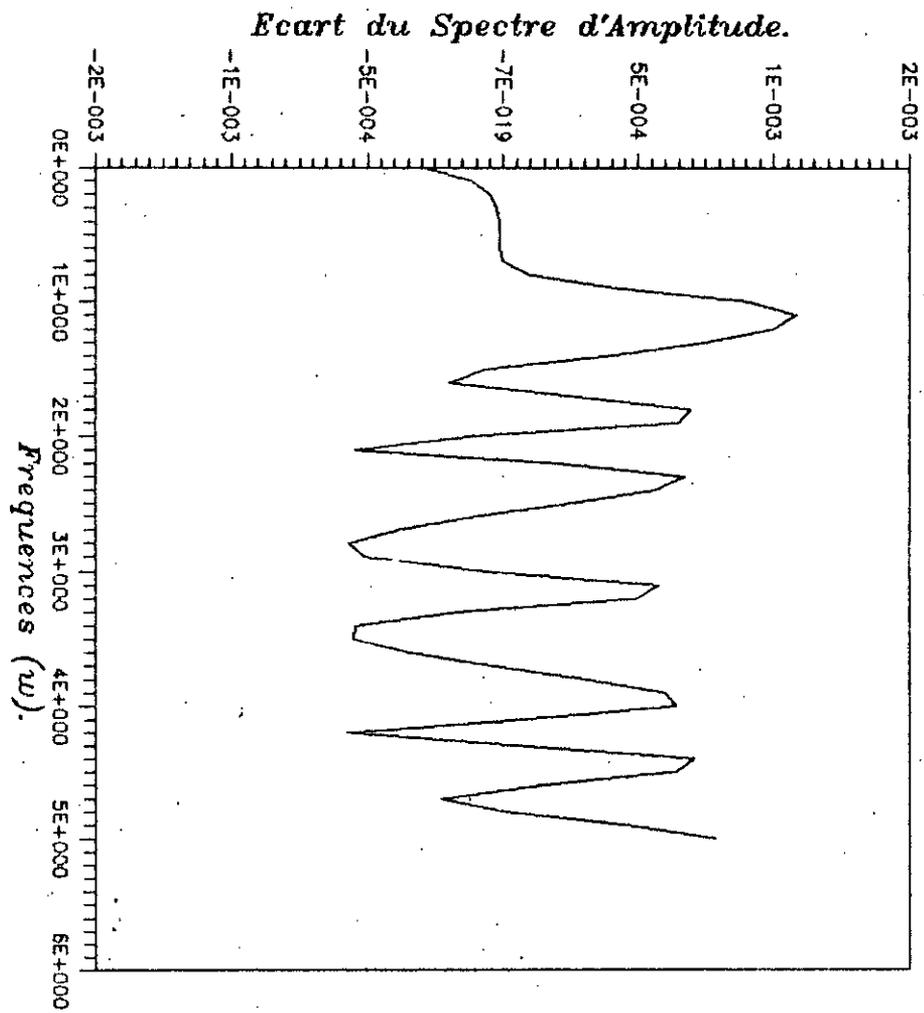
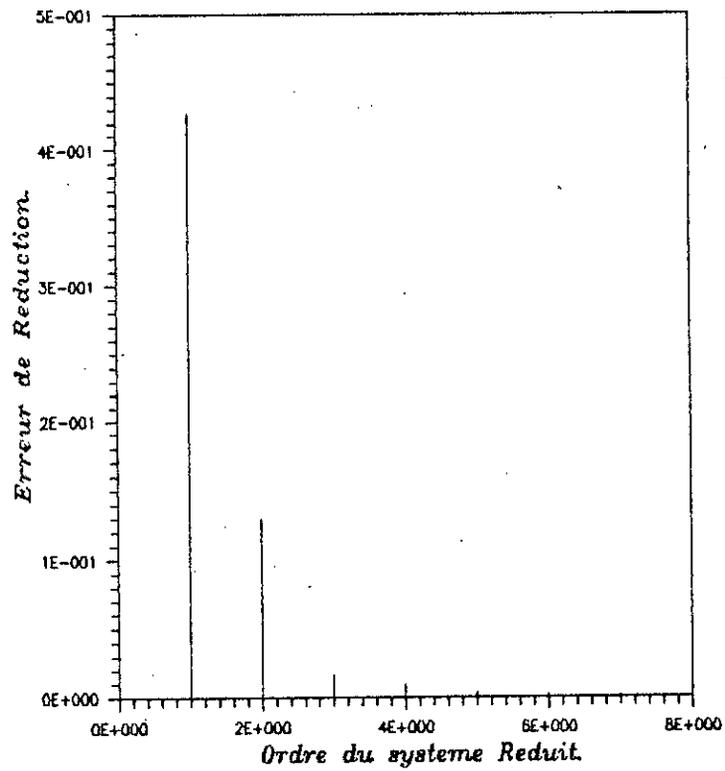


Fig. 5.2.c- Ecart du spectre d'Amplitude.



Plan de Stabilité (Lieu des POLES).

△△△△ Systeme Initial. (Ordre 8).
 ★★★★★ Systeme Reduit (Ordre 4).

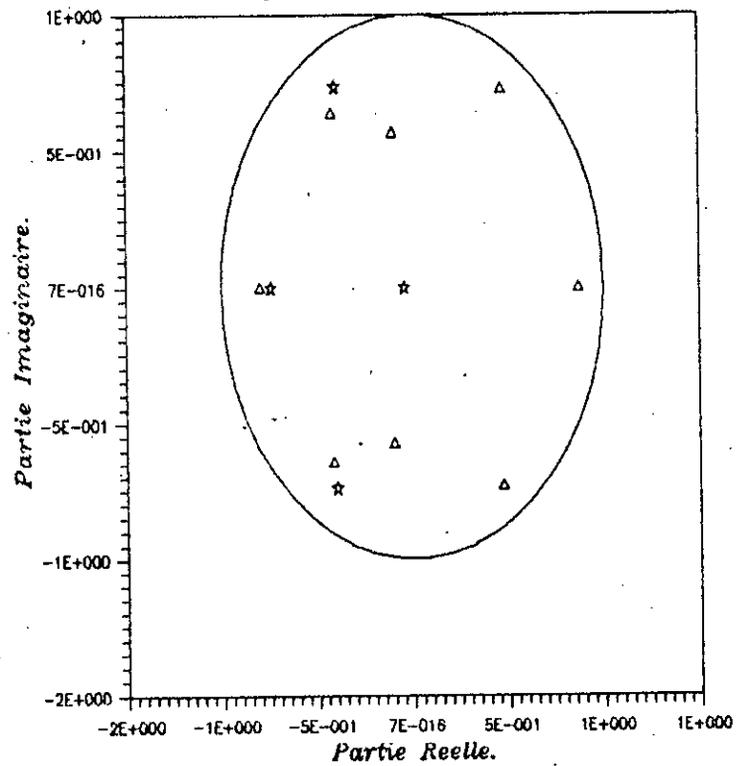


Fig. 5.2.d- Erreur de réduction et plan de stabilité.

* Le vecteur C réduit :

$$C_r = \begin{bmatrix} 3.8827E-02, & -4.0442E-02, & 2.4759E-02, & -1.0439E-02 \end{bmatrix}$$

- Les pôles (valeurs propres de A) du système réduit sont:

Partie réelle	Partie Imaginaire
-4.0146457862E-001	8.36572219898E-001
-4.0146457862E-001	-7.36572219898E-001
-7.4157339713E-001	0.000000000000
-4.4518284977E-002	0.000000000000

V.3 Cas Continu.

Exemple1.

Soit un modèle (M.I.M.O) d'ordre 4 représentant les dynamiques longitudinales d'un avion type F-8 [28].

- Le système initial est donné par la réalisation (A,B,C), avec:

* La matrice A initial :

$$A = \begin{bmatrix} -6.560E-3 & -7.577E-2 & 7.390E-4 & 3.564E-3 \\ 7.577E-2 & -8.383E-3 & 9.204E-4 & 4.445E-3 \\ -9.171E-4 & 1.142E-3 & -9.219E-2 & -3.086 \\ 3.597E-3 & -4.486E-3 & 3.136 & -1.816 \end{bmatrix}$$

* Le vecteur B initial :

$$B = \begin{bmatrix} -4.713 \\ 4.831 \\ -3.293E-1 \\ 1.292 \end{bmatrix}$$

* La matrice C initiale est donnée par:

$$C = \begin{bmatrix} 5.530E-4 & 1.241E-3 & -1.951E-1 & -1.743E-1 \\ 4.713 & 4.831 & -2.653e-1 & -1.281 \end{bmatrix}$$

- Les pôles du système initial correspondant sont:

Partie Réelle	Partie Imaginaire
-7.4711266992216E-003	7.5762156425065E-002
-7.4711266992216E-003	-7.5762156425065E-002
-9.5409537330078E-001	2.98911752573390
-9.5409537330078E-001	-2.98911752573390

- Le gramien équilibré correspondant est donné par :

$$\Sigma = \text{diag} \{ 1629.74, 1392.24, 0.59, 0.46 \}$$

- Les paramètres du système équilibré sont donnés par:

* La matrice A équilibrée:

$$A_b = \begin{bmatrix} -6.562E-3 & -7.577E-2 & 7.395E-4 & -3.562E-3 \\ 7.577E-4 & -8.382E-3 & 9.209E-4 & -4.443E-3 \\ -9.165E-4 & 1.142E-3 & -9.215E-2 & 3.082 \\ -3.600E-3 & 4.489E-3 & -3.139 & -1.816 \end{bmatrix}$$

* Le vecteur B équilibré:

$$B_b = \begin{bmatrix} 4.713 \\ -4.831 \\ 3.291E-1 \\ 1.293 \end{bmatrix}$$

* La matrice C équilibrée:

$$C_b = \begin{bmatrix} -5.530E-4 & -1.241E-3 & 1.982E-1 & -1.741E-1 \\ -4.713 & -4.831 & 2.655E-1 & -1.280 \end{bmatrix}$$

- Le système réduit est donné par la réalisation (A_r, B_r, C_r), soient:
($\xi = 10^{-5}$)

* La matrice A réduite :

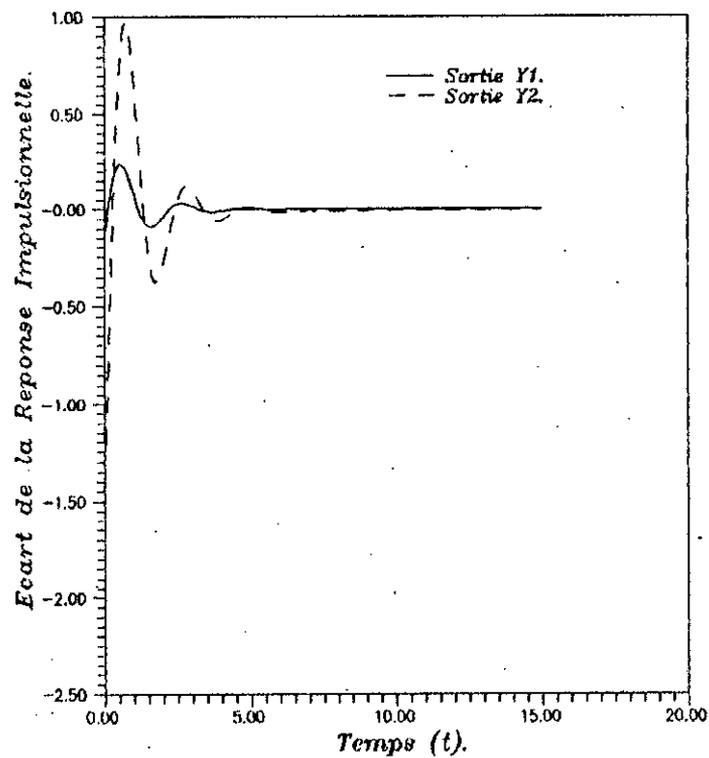
$$A_r = \begin{bmatrix} -6.561E-3 & -7.577E-2 \\ 7.577E-2 & -8.382E-3 \end{bmatrix}$$

* Le vecteur B réduit correspondant est :

$$B_r = \begin{bmatrix} 4.713 \\ -4.831 \end{bmatrix}$$

* La matrice C réduite est donnée par:

$$C_r = \begin{bmatrix} -5.530E-4 & -1.241E-3 \\ -4.713 & -4.831 \end{bmatrix}$$



i: Systeme Initial (Ordre 4).
r: Systeme Reduit (Ordre 2).

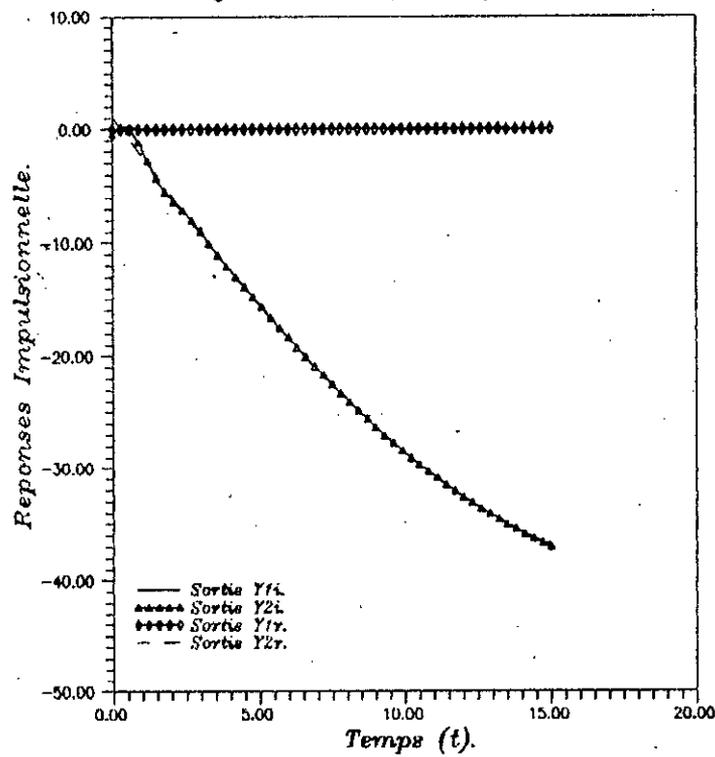


Fig. 5.3.a- Réponse Impulsionnelle et son écart.

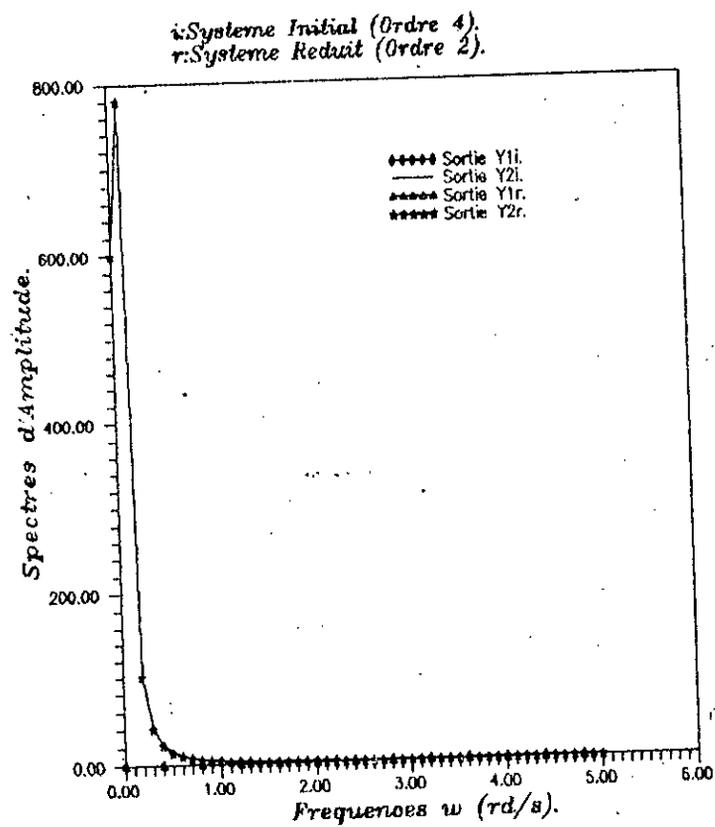
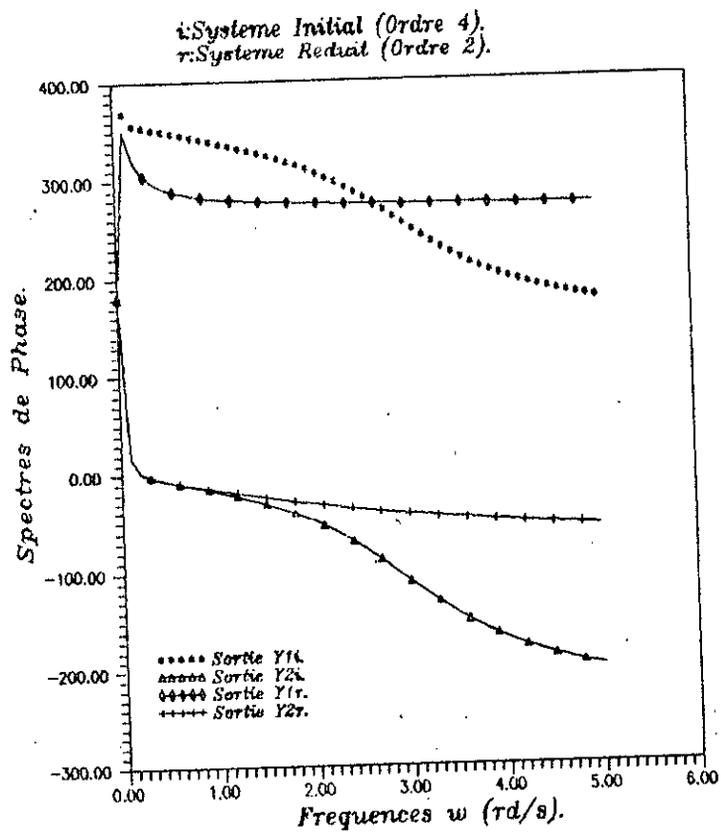


Fig. 5.3.b- Spectres d'Amplitude et de Phase.

Plan De Stabilité (lieu des Poles).

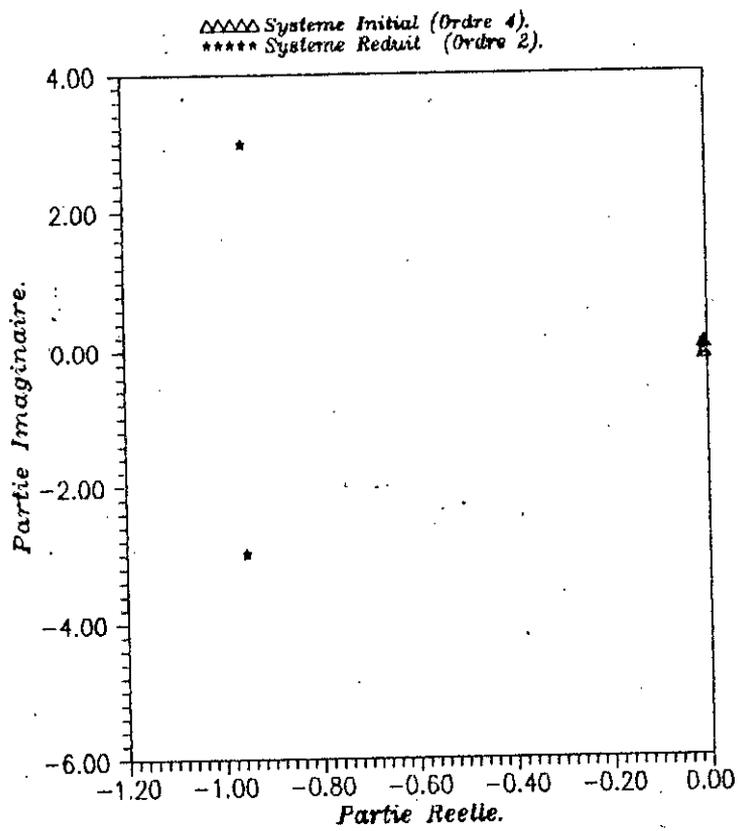


Fig. 5.3.c- Plan de stabilité.

= Les pôles du système réduit sont donnés dans le tableau ci-dessous:

Partie Réelle	Partie Imaginaire
-7.4714999966000E-003	7.5764517658614E-002
-7.4714999966000E-003	-7.5764517658614E-002

Exemple2.

L'exemple continu qu'on propose est un système d'ordre 8 donné dans [29].

= Le système initial est donné par la réalisation (A,B,C) suivante:

* La matrice A initiale:

$$A = \begin{bmatrix} -3.3000E+001 & -4.3700E+002 & -3.0170E+003 & -1.1870E+004 \\ 1.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 1.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 1.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 1.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \end{bmatrix}$$

$$\begin{bmatrix} -2.7470E+004 & -3.7492E+004 & -2.8880E+004 & -9.6000E+003 \\ 0.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 1.0000E+000 & 0.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 1.0000E+000 & 0.0000E+000 & 0.0000E+000 \\ 0.0000E+000 & 0.0000E+000 & 1.0000E+000 & 0.0000E+000 \end{bmatrix}$$

= Les vecteurs B et C initiaux sont respectivement:

$$B = \begin{bmatrix} 1.0000E+000 \\ 0.0000E+000 \end{bmatrix} \quad C^T = \begin{bmatrix} 1.0000E+000 \\ 3.1020E+001 \\ 3.7929E+002 \\ 2.3510E+003 \\ 7.9345E+003 \\ 1.4568E+004 \\ 1.3724E+004 \\ 5.5201E+003 \end{bmatrix}$$

= Les Pôles (valeurs propres de A) du système initial sont:

Partie réelle	Partie Imaginaire
-1	1.00
-1	-1.00
-3	0.00
-4	0.00
-5	0.00
-8	0.00
-10	0.00
-1	0.00

= Le gramien équilibré du système considéré est:

$$\Sigma = \left\{ 8.4774E-02, 5.1573E-04, 3.4508E-05, 2.4292E-06, 7.1072E-09, \right. \\ \left. -1.1666E-10, 1.8805E-12, -2.2929E-12 \right\}$$

= Le système équilibré est représenté par la réalisation (Ab, Bb, Cb) telle que:

* La matrice A équilibrée est:

$$A_b = \begin{bmatrix} -1.6434E+00 & 3.7069E-01 & -2.6696E-01 & -3.3030E-01 \\ -1.0811E+00 & -6.8412E-01 & 1.4634E+00 & 1.3915E+00 \\ -6.5831E-01 & -1.2361E+00 & -1.4306E+00 & -2.9218E+00 \\ -7.1005E-01 & -1.0382E+00 & -2.4571E+00 & -8.1273E+00 \\ -6.2876E-01 & -8.5758E-01 & -2.6708E+00 & -1.3281E+01 \\ -1.8299E-03 & -2.4634E-03 & -9.2147E-03 & -8.2149E-02 \\ -7.6457E-04 & -1.0504E-03 & -3.2533E-03 & -1.7663E-02 \\ -8.9012E-06 & -1.2241E-05 & -3.7817E-05 & -2.0582E-04 \end{bmatrix}$$

$$\begin{bmatrix} 5.4828E-02 & 5.4168E-01 & -7.5852E-01 & -1.9294E+00 \\ -3.1501E-01 & -1.5763E+00 & 3.6612E+00 & 3.1963E+01 \\ -1.5644E-02 & 9.7584E+00 & -3.2510E+00 & 1.6697E+02 \\ -2.5475E+00 & 5.1516E+01 & 9.1211E+00 & 1.2823E+03 \\ -1.5559E+01 & 1.7362E+02 & 1.5254E+02 & 6.5721E+03 \\ -4.3706E-01 & -6.4160E-01 & 8.8055E+00 & 1.6323E+02 \\ -4.1356E-02 & -1.7565E-01 & -2.8062E+00 & -2.2084E+01 \\ -5.0383E-04 & -3.0888E-03 & 5.8911E-02 & -2.1067E+00 \end{bmatrix}$$

- Les vecteurs B et C équilibrés:

$$B_b = \begin{bmatrix} -1.0184208568E+00 \\ -3.0952972782E-01 \\ -2.0796242100E-01 \\ -2.2101610861E-01 \\ -1.9479836134E-01 \\ -5.6605029369E-04 \\ -2.3695788758E-04 \\ -2.7588798003E-06 \end{bmatrix}$$

$$C_b^T = \begin{bmatrix} -9.7745253267E-01 \\ 1.0174352156E-01 \\ -8.1246692827E-02 \\ -1.0007296339E-01 \\ 1.5042043216E-02 \\ 1.7616587307E-01 \\ -2.1297110640E-01 \\ -8.0356225371E-03 \end{bmatrix}$$

- L'erreur de Hankel (en fonction de l'ordre k) est:

Ordre du système réduit	Erreur de Hankel
1	0.0061
2	0.0004
3	0.0000
4	0.0000
5	0.0000
6	0.0000
7	0.0000

- Le système réduit est donné par la réalisation (A_r, B_r, C_r), soient: ($\xi = 10^{-3}$).

* La matrice A réduite :

$$A_r = \begin{bmatrix} -1.6434E+000 & 3.7069E-001 & -2.6697E-001 \\ -1.0812E+000 & -6.8413E-001 & 1.4634E+000 \\ -6.5832E-001 & -1.2362E+000 & -1.4306E+000 \end{bmatrix}$$

* Le vecteur B réduit :

$$B_r = \begin{bmatrix} -1.0184208568E+00 \\ -3.0952972782E-01 \\ -2.0796242100E-01 \end{bmatrix}$$

* Le vecteur C réduit :

$$C_r = \left[-9.7745253267E-01, 1.0174352156E-01, -8.1246692827E-02 \right]$$

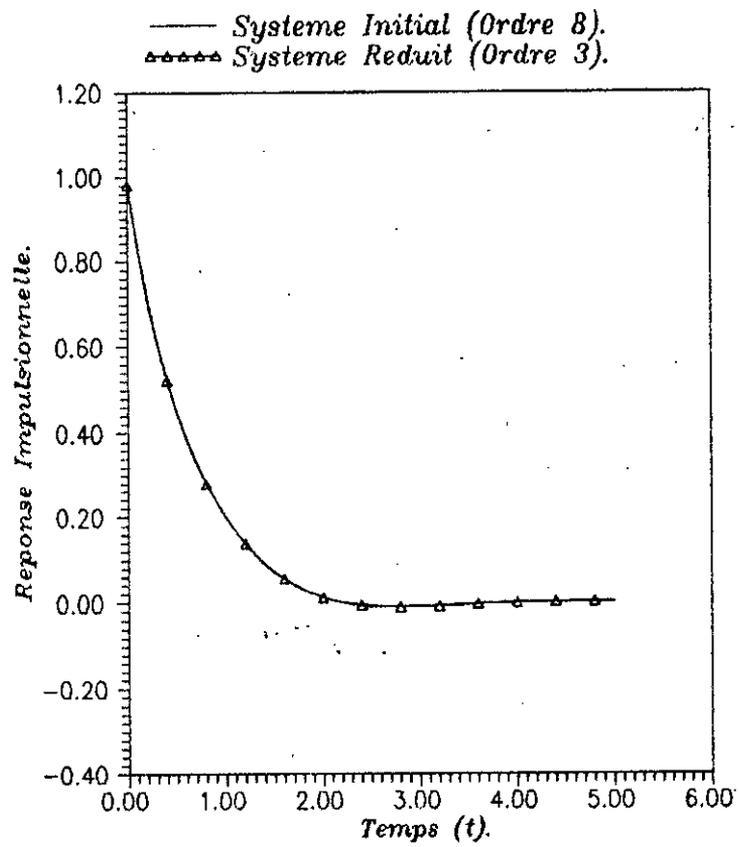
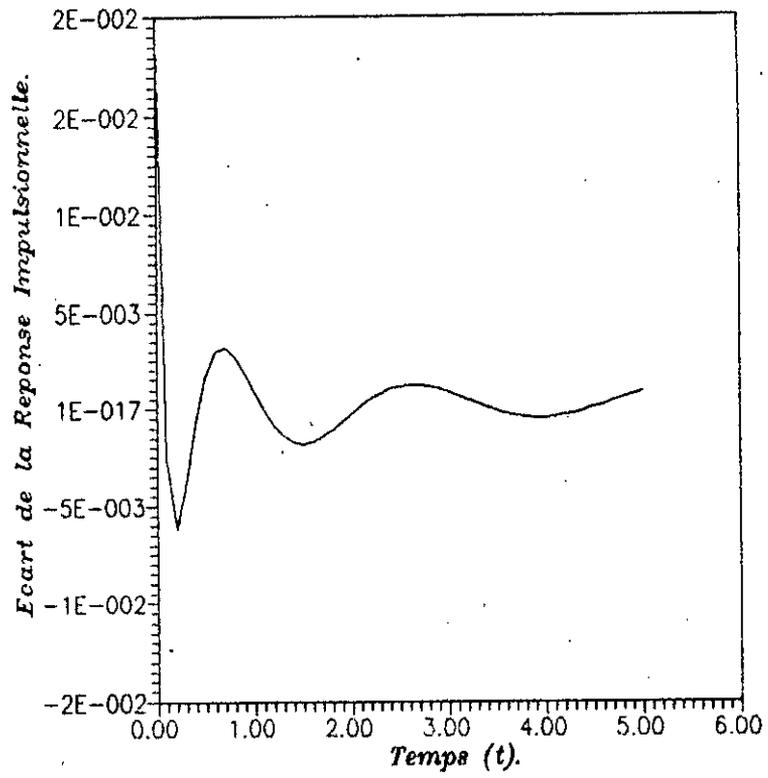


Fig. 5.4.a- Réponse Impulsionnelle et son écart.

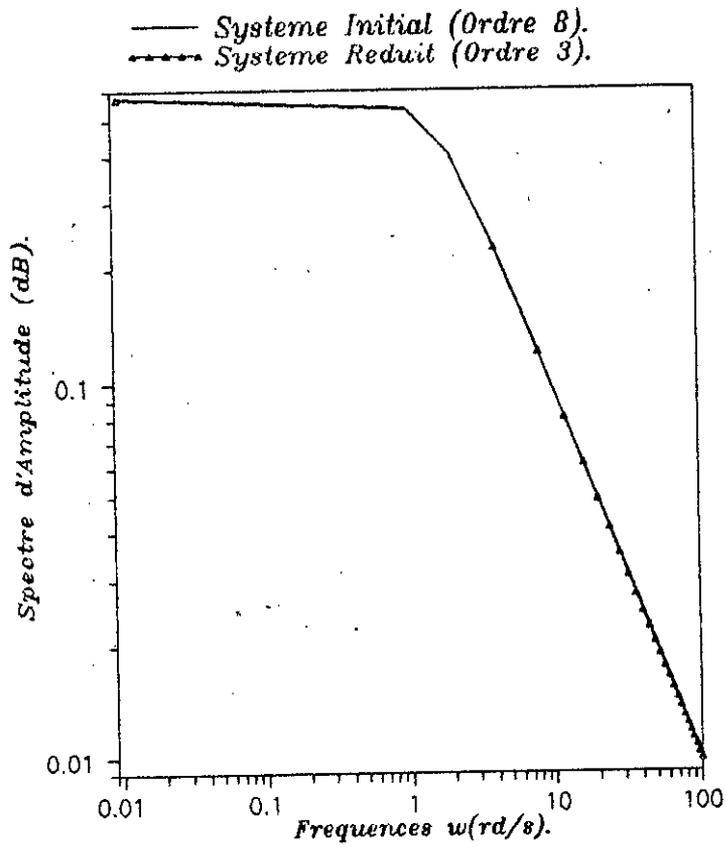
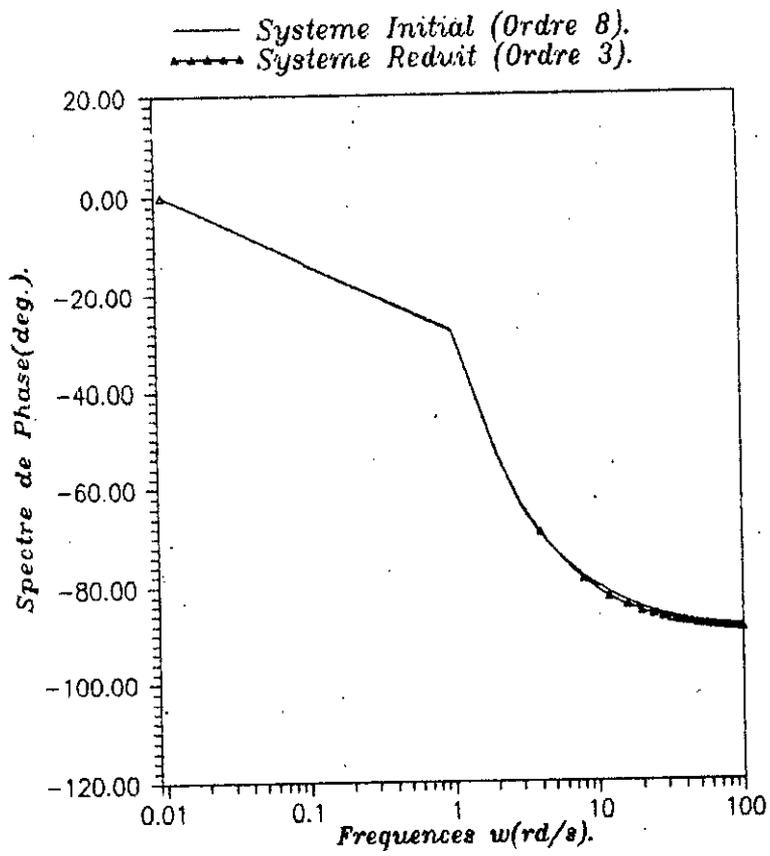
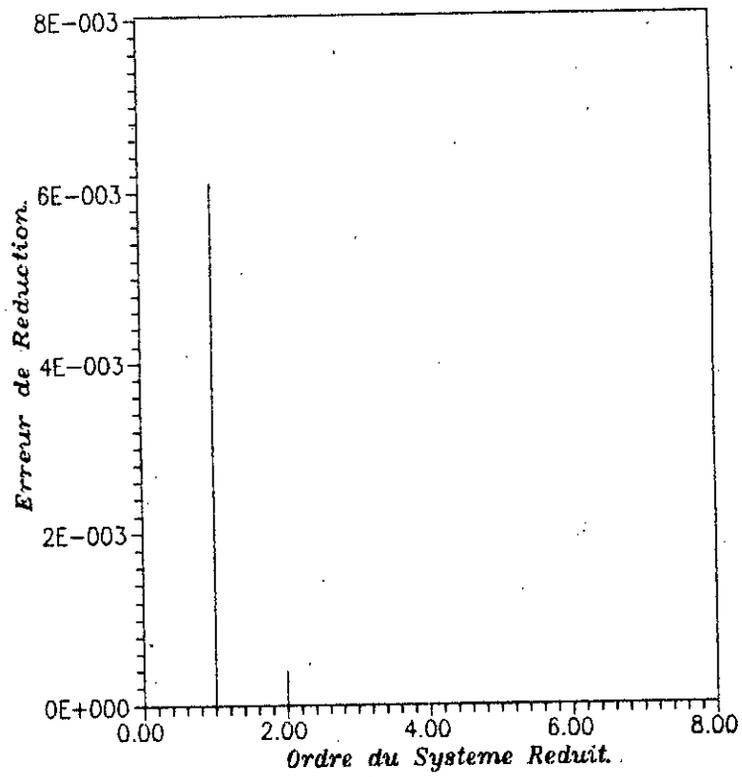


Fig. 5.4.b- Spectres d'Amplitude et de Phase.



***** Systeme Initial (Ordre 8).
 ▲▲▲▲▲ Systeme Reduit (Ordre 3).

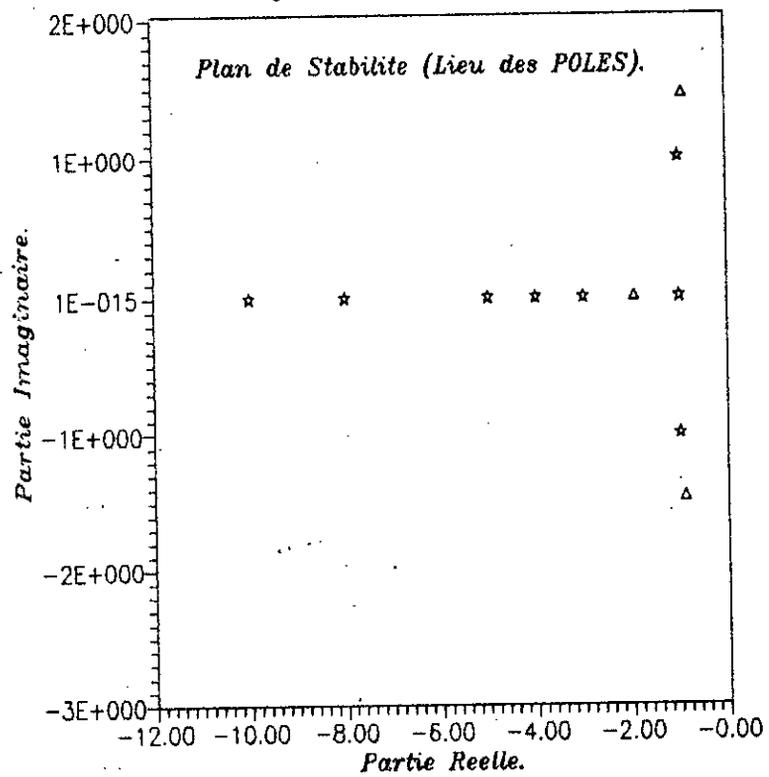


Fig. 5.4.c- Erreur de réduction et plan de stabilité.

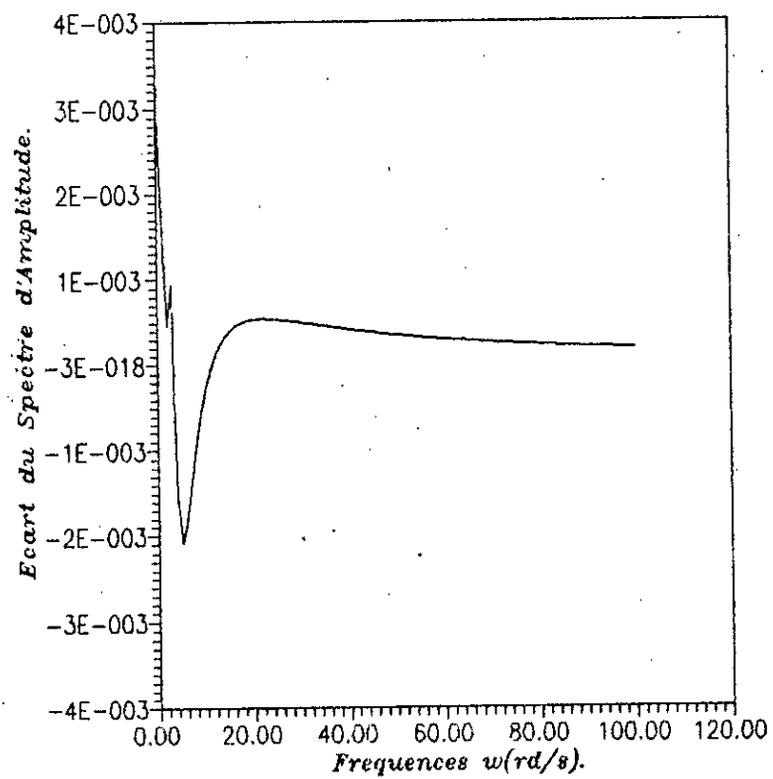


Fig. 5.4.d- Ecart du spectre d'Amplitude.

- Les modes (valeurs propres de A) du système réduit sont:

Partie réelle	Partie Imaginaire
-1.94130937244320	0.000000000000000
-9.0841031377E-01	1.46148447377890
-9.0841031377E-01	-1.46148447377890

V.3 Interprétations.

La technique de troncature de Moore donne un sous-système qui présente une bonne approximation de la réponse impulsionnelle réelle du système initial (sous-système à dominance interne). Elle présente une erreur de réduction non nulle aux basses fréquences, qui tend à s'annuler aux fréquences élevées. Le choix de l'ordre du système, fonction de la donnée de l'erreur, est fixé par le concepteur.

On remarque le déplacement des pôles vers l'axe imaginaire (cas continu), vers le cercle unité (cas discret), comme on note la manifestation de faibles ondulations dans la bande passante du système discret (filtre de BUTTERWORTH numérique) réduit.

Un filtre de même ordre que celui du réduit, synthétisé directement présenterait une B.P plate.

Le système est minimal pour une erreur fixée.

De même, la stabilité et causalité sont retenues dans le modèle réduit.

V.4 Conclusion.

Dans ce chapitre, des applications pour la réduction de systèmes ont été proposées. Les résultats montrent l'intéressante abilité de la méthode dans la conservation des propriétés clé du système initial telles la stabilité et la causalité.

Le fait que la méthode se base essentiellement sur la stabilité du système initial et sur le calcul des gramiens fait la restriction de la méthode aux systèmes asymptotiquement stables, gouvernables et observables.

CONCLUSION GENERALE

Dans le travail présenté, une des plus récente méthode de réduction de systèmes complexes a été développée et un logiciel de réduction de l'ordre de tels systèmes a été élaborer.

On peut conclure d'après les résultats obtenus que la méthode de réduction de systèmes via leur réalisations équilibrées, est optimale par rapport a l'ordre du système réduit choisi.

Il est très intéressant que le système réduit garde presque les mêmes caractéristiques que le système initial.

En effet, la réduction de l'ordre d'un système n'affecte que peu sa réponse impulsionnelle, sa réponse fréquentielle (Amplitude et Phase), les écarts de ces derniers en témoignent.

Il faut aussi remarquer que la réduction conserve la minimalité, stabilité, causalité, gouvernabilité et observabilité du système initial, le système obtenu est donc qualifié de physiquement réalisable.

Bien que les résultats obtenus soient satisfaisantes, cette approche reste déconseillée dans le cas où le système à réduire présente des valeurs singulières de même ordre de grandeur, l'élimination de l'une de ces dernières donnera un système réduit qui ne suivra que peu le comportement du système initial.

La simplification de modèles complexes reste un domaine vaste de recherche et de développement, une continuité de ce travail sera très envisageable.

Cette étude nous a permis d'une part d'approfondir nos connaissances en matière d'AUTOMATIQUE, surtout en ce qui concerne réduction de modèles, d'autre part de s'adapter à un environnement informatique et maîtriser la programmation en PASCAL.

Référence Bibliographique

- [1] B.C. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability and Model Reduction ", *IEEE Trans. on Automatic Control*, Vol. AC-26, pp 17-31, Feb., 1981.
- [2] A. Fossard, *Commandes des systèmes multidimensionnels*, 1972.
- [3] M. Jamshidi & M.M. Zavarei, *Linear Control Systems*, 1987.
- [4] R.A. Gabel & R.A Roberts, *Signals and linear systems*, New York, 1987.
- [5] Matlab , *The Math Works, Inc.*, June 17, 1989.
- [6] R. Boite & M. Kunt, *Traitement de la parole*, Presses polytechniques romandes, Lausanne, 1987.
- [7] F. Parant, "Modélisation et identification d'un système fortement oscillant. Application à l'entraînement de bande de métal", Thèse de Doctorat de L'I.N.P.L, Nancy, Juillet 1990.
- [8] M.S. Singh & A. Titli, *Systems: Decomposition, Optimisation and Control*, Laboratoire d'Automatique et d'Analyse des Systèmes du C.N.R.S.
- [9] R. J. Ober, "Balanced realizations : Canonical form, Parametrization ", *Int. J. C.*, Vol:46, NO.2, pp.643-670, 1987.
- [10] L. Pernebo & L. Silverman, "Model reduction via balanced state space representations", *IEEE, Trans. on Auto. Control*, Vol. Ac-27, NO.2, April 1982.
- [11] S. Shokoohi & L. Silverman, "Linear time-variable systems : balancing and model reduction", *IEEE, Trans on Auto. Control*, Vol. Ac-28, NO.8, August 1984.
- [12] A. Laub, M.T. Heath, C.C. Paige & R.C. Ward, "Computation of systems balancing transformations and other applications of simultaneous diagonalization algorithms ", *IEEE Trans. on Autom. Control*, Vol. AC.32, NO.2, February 1987.
- [13] J.E. Mason, "Identification using low order systems", Ph.D Dissertation University of California, Berkely, 1988.

- [14] M. Bettayeb, "New interpretation of balancing state space representation as an input-output energy minimization problem", *Int. J. Systems SCI*, 1991.
- [15] N. J. Young, "Balanced realization via operators", *Int. J. Control*, 1984, Vol. 42, NO.2, 369-389.
- [16] P. T. Kabamba, "Balanced Forms: Canonicity and Parametrization", *IEEE, Trans. on Automatic Control*, Vol. AC-30, NO.1, November, 1985.
- [17] C. Mullis & R. Roberts, "Synthesis of minimum Roundoff noise fixed point digital filters", *IEEE*, Vol. Cas-23, NO.9, September, 1976.
- [18] D. D. Siljak, "Complex dynamic systems, dimensionality, structure and uncertainty", *IFAC/IFORS. Symposium, Warsaw, Poland*, 11-15 July, 1983.
- [19] P. A. Löf, T. Smed, G. Anderson & D. J. Hill, "Fast calculation of a voltage stability index", *Trans. of Power Systems*, Vol.7, NO.1, February, 1992.
- [20] B. Derras, "Model reduction via a state-space based criterion" *Departement d'Electronique, ENP, Alger*, 1993.
- [21] M. G. Safonov & R. Y. Chiang, "A shur method for Balanced model reduction", *IEEE, Transactions On Automatic Control*, Vol.34, No.7, July 1989.
- [22] D. Rothschild & A. Jameson, "Comparaison of four numerical algorithms for solving the Lyapunov matrix equation", *Int. J. Control*, 1970, Vol.11, NO.2, 181-198.
- [23] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ error bounds", *Int. J. Control*, Vol. 39, pp. 1115-1193, 1984.
- [24] A. Antoniou, *Digital filters: Analysis and design*, Mc Graw-Hill, New York, 1958.
- [25] K. J. Astrom & B. Wittlenmark, *Computer, controlled systems*, USA, 1990.
- [26] Grapher, Golden Software Inc, Colorado, USA, 1988.

- [27] U. M. Al-Saggaf & G. F. Franklin, "Model reduction via balanced realizations : An extension and frequency weighting techniques", *IEEE, Trans. on Auto. Control*, Vol. 33, NO. 7, July 1988.
- [28] K. Vincenza Fernando & H. Nicholson, "Singular perturbational model reduction of balanced systems", *IEEE, Trans. on Auto. Control*, AC-27, NO. 2, April 1982.
- [29] P. Gutman, C. F. Mannerfelt & P. Molander, "Contributions to the model reduction problem", *IEEE, Trans. on Auto. Control*, AC-27, NO. 2, April 1982.
- [30] S. Tin, J. Lam & D. Ho, "Comments on New algorithm for minimal balanced realization of transfer function matrix", *Int. J. Control*, 1991, Vol. 53, NO. 5, 1255-1259.

Annexe # 1

Rappel sur le Calcul Matriciel

Notion de matrice: Une matrice est un tableau rectangulaire dont les éléments peuvent être des êtres mathématiques quelconques.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \dots a_{1n} \\ a_{21} & a_{22} & a_{23} \dots a_{2n} \\ \dots & \dots & \dots \dots \dots \\ a_{m1} & a_{m2} & a_{m3} \dots a_{mn} \end{bmatrix}$$

Où: a_{ij} est l'élément de la matrice A, se situant à la $i^{\text{ième}}$ ligne et la $j^{\text{ième}}$ colonne.

Si $m=n$; la matrice A est dite " Carrée ".

Notation.

$$A = [a_{ij}] \quad \left| \begin{array}{l} i = \overline{1, n} \\ j = \overline{1, m} \end{array} \right.$$

Matrice diagonale: C'est une matrice carrée dont tous les éléments non situés sur la diagonale principale sont nuls, c-à-d;

$$a_{ij} = 0 \quad \forall \quad i \neq j$$

Matrice Unité: La matrice unité I est un cas particulier d'une matrice diagonale. Dans ce cas, les éléments diagonaux sont égaux à 1.

Matrice nulle: La matrice dont tous les éléments sont nuls appelée est la matrice nulle, au lieu de reproduire la matrice des éléments a_{ij} nuls, on écrira

$$A = \underline{0}.$$

Matr e Unitaire: Une matrice A est dite unitaire ssi elle vérifie:

$$AA^t = I.$$

Matrice transposée: La matrice A^t transposée de A est obtenue en interchangeant les lignes et les colonnes de cette dernière.

Si $A = [a_{ij}]$ et $B = A^t$ alors $B = [b_{ij}] = [a_{ji}]$.

Remarque

Si A est une matrice (n, m) , alors A^t est une matrice (m, n) .

Trace d'une matrice: La trace d'une matrice A est donnée par la somme des éléments diagonaux:

$$\text{Tr}(A) = \sum_{i=1}^n a_{ii} = a_{11} + a_{22} + \dots + a_{nn}$$

Les mineurs: Le mineur M_{ij} de l'élément a_{ij} du déterminant $|A|$ est la matrice obtenue en supprimant la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne de A .

Cofacteur: Le cofacteur C_{ij} de a_{ij} s'obtient à partir de son mineur M_{ij} par la relation:

$$C_{ij} = (-1)^{i+j} M_{ij}$$

Matrice adjointe: C'est la transposée de la matrice obtenue en remplaçant chaque élément par son cofacteur.

$$\text{Adj}(A) = [C_{ij}]^t = [C_{ji}]$$

Rang d'une matrice: Le rang (r) d'une matrice indique le nombre de lignes ou de colonnes linéairement indépendantes.

La matrice A possède le rang r_i , s'il existe au moins un sous-déterminant non nul avec r_i lignes et tous les sous-déterminants avec r_{i+1} lignes sont nuls.

Déterminant: Seule une matrice carrée possède un déterminant:

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \dots & \dots & a_{nn} \end{vmatrix} = |A|$$

Le déterminant d'une matrice carrée (n,n) , $A=(a_{ij})$, est la somme suivante qui est obtenue par la somme sur toutes les permutations possibles $\sigma = j_1 j_2 \dots j_n$ de S_n :

$$|A| = \sum_{\sigma} (\text{sgn } \sigma) a_{1\sigma(1)} a_{2\sigma(2)} \dots a_{n\sigma(n)}$$

Matrice définie-positive: Soit S une matrice carrée, symétrique, on dit qu'elle est définie positive, si sa forme quadratique obéit à la condition suivante:

$$\forall x \neq 0 ; q = x^t S x > 0$$

Où: x est un vecteur quelconque.

Le test de Sylvester est souvent utilisé pour vérifier si une matrice S est définie positive: Tous les sous-déterminants de S doivent être positifs.

C-à-d:

si $S = [S_{ij}]$

$$S_{11} > 0; \begin{vmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{vmatrix}; \begin{vmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{vmatrix}; \dots; \det(S) > 0.$$

Matrice Régulière et Singulière: Le rang d'une matrice possède une importance particulière. Lorsqu'on a $r(A) = n$, on dit que la matrice A est régulière: elle possède donc n lignes et n colonnes indépendantes.

Dans le cas contraire, c-à-d; $r(A) < n$, la matrice A sera dite singulière.

Théorème de Cayley-Hamilton: Ce théorème dit que chaque matrice A satisfait sa propre équation caractéristique; c-à-d;

$$A^n + C_{n-1} A^{n-1} + \dots + C_2 A^2 + C_1 A + C_0 I_n = 0$$

Où:

C_i : $i=0, n-1$ sont les coefficients de l'équation caractéristique de A.

Valeurs Propres: Le scalaire λ est dit valeur propre de la matrice carrée A s'il existe un vecteur colonne non nul u tel que:

$$Au = \lambda u$$

Vecteurs Propres: Le vecteur propre x associé à une matrice carrée A est défini par la relation :

$$Ax = \lambda x$$

ou encore:

$$(\lambda I - A) \cdot x = 0$$

Cette équation matricielle possède en général plusieurs solutions.

Opérations sur les matrices:

* Egalité $A = B$ si $a_{ij} = b_{ij} \quad \forall i, j$

* Addition $C = A+B$ si $c_{ij} = a_{ij} + b_{ij} \quad \forall i, j$

* Produit Matriciel Soient $A_{(m,n)}$ et $B_{(n,k)}$ deux matrices quelconques soit C, la matrice résultante du produit $C=A \cdot B$

telque:

$$C = \sum_{p=1}^n a_{ip} \cdot b_{pj} \quad \forall \begin{cases} i=1, \dots, m \\ j=1, \dots, k \end{cases}$$

* Inversion Matricielle La matrice A^{-1} , inverse de A est donnée par:

$$A^{-1} = \frac{\text{Adj}(A)}{|A|}$$

Annexe # 2

ALGORITHME DE SVD [19]
(Singular Values Decomposition)

Soit une matrice réelle $A(n,n)$, sa décomposition en valeurs singulière (SVD) est donnée par:

$$A = U \Sigma V^t = \sum_{i=1}^n \sigma_i u_i v_i^t \quad (A2.1)$$

Où:

U, V sont des matrices orthogonales (n,n) ;
 u_i, v_i sont les vecteurs singuliers colonnes des matrices U
 et V respectivement;
 $\Sigma = \Sigma(A) = \text{diag} \left\{ \sigma_i \right\}_{i=1, n}$: matrice diagonale contenant les
 valeurs singulière de A avec $\sigma_i \geq 0 \forall i=1, n$.

Les éléments σ_i sont ordonnées par ordre décroissant, c-à-d;
 $\sigma_i \geq \sigma_j$ pour tout $i < j$.

Si A est de rang r (voir A1) ($r \leq n$), ses valeurs singulières sont les racines carrées des r valeurs propres positives de $A^t A$, qui sont aussi les r valeurs propres positives de AA^t .

U et V sont les matrices orthogonales de dimension (n,n) et leurs colonnes contiennent les vecteurs propres de $A^t A$ respectivement AA^t .

Si σ_i est la $i^{\text{ième}}$ valeur singulière de A , le vecteur u_i est le $i^{\text{ième}}$ vecteur singulier droit, d'où on peut écrire les relations liant les valeurs singulières et vecteurs singuliers gauche et droit de la matrice A comme suit:

$$\begin{cases} Av_i = \sigma_i u_i \\ A^t u_i = \sigma_i v_i \end{cases} \quad (A2.2)$$

Donc les valeurs singulières de la matrice A , par construction, leurs carrés sont les valeurs propres de AA^t (ou $A^t A$) et sont décrits par la relation suivante:

$$\lambda(A^t A) = \lambda(AA^t) = \Sigma^2(A) \quad (A2.3)$$

Où:

$$\lambda(X) = \text{diag} \left\{ \lambda_i(X) \right\} \quad (A2.4)$$

Avec:

- $i=1, \dots, n$ et λ_i : la $i^{\text{ème}}$ valeur propre de la matrice X
- $\Lambda(A^t A)$ est la matrice diagonale contenant les valeurs propres de la matrice AA^t ou $A^t A$;
- Σ est la matrice diagonale contenant les valeurs singulières de la matrice A .

Remarque: Si $A = U\Sigma V^t$ est la décomposition en valeurs singulières de A , alors les décompositions en valeurs propres de $A^t A$ et de AA^t sont données par les équations suivantes:

$$\begin{cases} - A^t A = V(\Sigma^t \Sigma)V^t = V(\Sigma^2)V^t \\ - AA^t = U(\Sigma \Sigma^t)U^t = U(\Sigma^2)U^t \end{cases} \quad (A2.5)$$

Annexe # 3

Algorithme de JAMESON [22]

" Résolution de l'équation de Lyapunov, Cas continu "

Soit à résoudre l'équation de Lyapunov, donnée par:

$$AX + XA^T = C \quad (A3.1)$$

Où:

- A: Matrice(n,n),
- B: Matrice(n,n),
- X: Matrice(n,n), inconnue.

Ayant les deux matrices A et C; la procédure de calcul de la matrice inconnue X, solution de (A3.1), est la suivante:

Etape Une. Détermination des coefficients du polynôme caractéristique de la matrice A.

On utilisera la méthode de SOURIAU :

Soient (a_i) , $(i=0, n)$, les $(n+1)$ coefficients du polynôme caractéristique:

$$P(A) = \sum_{i=0}^n a_i \lambda^{n-i} = a_0 \lambda^n + a_1 \lambda^{n-1} + \dots + a_{n-1} \lambda + a_n$$

Afin de calculer ces coefficients, on définit une suite de matrices (A_k) obéissant à la relation de récurrence suivante:

Conditions initiales:

$$A_0 = 0; a_0 = 1.$$

Boucle:

Pour $k=1, n$, faire:

$$A_k = (A_{k-1} + a_{k-1} I) \cdot A$$

$$a_k = (-1/k) \cdot \text{tr}(A_k)$$

où $\text{tr}(A_k)$ désigne la trace de la matrice A_k .

Les coefficients (a_i) $(i=1, n)$ sont alors obtenus de proche en proche.

Etape deux. Calcul de la matrice S.

Conditions initiales:

$$C_0 = 0$$

$$C_1 = C$$

$$S_0 = 0$$

Boucle

Début:

Pour $k=1, \overline{n}$, faire:

$$\begin{cases} C_k = A^{k-1} C - C_{k-1} A^t \\ S_k = S_{k-1} + (-1)^{-k} a_{(n-k)} C_k \end{cases}$$

Fin:

$$S = S_n$$

Etape Trois. Calcul de la matrice G^{-1}

1-Calcul des coefficients r_{ij} .

Pour:

avec:

$$\begin{cases} i=1, \overline{2} \\ j=1, n \\ k=0, n \end{cases} \quad r_{ij} = a_{ijk} \quad \begin{cases} r_{11}=a_0; r_{12}=a_2; \\ r_{13}=a_4; r_{14}=a_6; \dots \\ r_{21}=a_1; r_{22}=a_3; \\ r_{23}=a_5; r_{24}=a_7; \dots \end{cases}$$

Les coefficients r_{ij} restants sont calculés par la règle de Routh comme suit:

pour:

$$\begin{cases} i=3, \overline{n+1} \\ j=1, n \end{cases}$$

on a :

$$r_{(i,j)} = r_{(i-2,j+1)} \cdot r_{(i-1,j+1)} \cdot r_{(i-2,1)} / r_{(i-1,1)}$$

G^{-1} s'écrit alors comme suit:

$$G^{-1} = \frac{(-1)^n}{2 \cdot r_{(n+1,1)}} \cdot H_{(n+1)}$$

Où:

$$\begin{array}{l}
 H_1 = I \\
 H_2 = I \\
 \vdots \\
 H_{(n+1)} = H_{(n-1)} + \frac{r_{(n-1,1)}}{r_{(n,1)}} \cdot A \cdot H_n
 \end{array}$$

La solution de l'équation (A3.1) est donnée alors par:

$$X = G^{-1} S_n$$

Remarque: Le temps de calcul pour une matrice X de dimension (9,9) est de 0.9 sec.

Annexe # 4
Algorithme de CHOLESKY

Ayant une matrice A définie positive symétrique, on peut toujours la décomposer sous la forme:

$$A = SS^t$$

où: - A: matrice de dim. (n,n),
- S: matrice triangulaire inférieure.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{ni} & \dots & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} s_{11} & 0 & \dots & 0 \\ \vdots & s_{22} & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ s_{ni} & \dots & \dots & s_{nn} \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_{nn} \end{bmatrix}$$

Les coefficients S_{ij} ($i=1,\overline{n}$; $j=1,\overline{n}$) sont donnés par:

$$S_{ii} = \left(a_{ii} - \sum_{j=1}^{i-1} S_{ij}^2 \right)^{1/2} : \text{éléments diagonaux (} i=1,\overline{n} \text{)}$$

$$S_{ji} = \begin{cases} 0 & , i > j \\ \frac{1}{S_{ii}} \left(S_{ij} - \sum_{k=1}^{i-1} S_{jk} S_{ik} \right) & , j = \overline{i+1, n} \end{cases}$$

Annexe # 5
DEMONSTRATION

Question.

Qu'elle est la direction que doit prendre x_0 dans l'espace à n dimensions pour maximiser le rendement énergétique I ?

- La réponse est donnée par le théorème suivant.

Théorème.

$$I_{\max} = \max \frac{x_0^T W x_0}{x_0^T W_g^{-1} x_0}$$

est atteint pour x_0^* le vecteur propre maximal obtenu par résolution du problème des valeurs propres

$$W x_0^* = \lambda_{\max}^2 W_g^{-1} x_0^*$$

Une fois normalisé

$$x_0^{*T} W_g^{-1} x_0^* = \lambda_{\max}^{-1}, \quad \lambda_{\max} > 0$$

il vient que

$$I_{\max} = \lambda_{\max}^2$$

Preuve.

Considérons la décomposition en valeurs propres de W_g

$$W_g = U_g \Sigma_g U_g^T$$

alors

$$\begin{aligned} W_g^{-1} &= U_g \Sigma_g^{-1} U_g^T \\ &= \left[U_g \Sigma_g^{-1/2} \right] \left[\Sigma_g^{-1/2} U_g^T \right] \\ &= S^T S \end{aligned}$$

posons $z = Sx_0$, on aura:

$$I = \frac{x_0^T W x_0}{x_0^T W^{-1} x_0} = \frac{z^T S^{-T} W_0 S^{-1} z}{z^T z}$$

La maximisation de I est équivalente à

$$\max_{z \in \mathbb{R}^n} \frac{z^T S^{-T} W_0 S^{-1} z}{z^T z}$$

Il est évident que la maximisation est achevée lorsque z est le vecteur propre de

$$S^{-T} W_0 S^{-1}$$

correspondant à la valeur propre maximale $\lambda_{\max}^2 = \lambda_{\max}^2 (S^{-T} W_0 S^{-1})$ est normalisée par

$$z^T z = \lambda_{\max}^{-1}$$

On a alors,

$$S^{-T} W_0 S^{-1} z = \lambda_{\max}^2 z$$

ou

$$S^{-T} W_0 S^{-1} Sx_0 = \lambda_{\max}^2 Sx_0$$

Ainsi

$$W_0 x_0 = \lambda_{\max}^2 S^T Sx_0$$

$$W_0 x_0 = \lambda_{\max}^2 W^{-1} x_0$$

et

$$z^T z = S^T Sx_0 = x_0^T W^{-1} x_0 = \lambda_{\max}^{-1}$$

le théorème est démontré pour $z = x_0^*$. On a aussi:

$$I_{\max} = \frac{z^T \lambda_{\max}^2 z}{z^T z}$$

LISTING DU PROGRAMME

PROCEDURES UTILISEES

Procedure Identite (var Id; n, m).

Cette procédure donne en sortie la matrice Identité Id.

Procedure LireMat (var T; n, m).

Cette sous-routine effectue la lecture des éléments d'une matrice.

Procedure EcrireMat (var T; n, m).

Cette sous-routine permet l'affichage du contenu d'une matrice.

Procedure DataInFile (var T; n, m).

Le sous-programme permet la mise en fichier d'une matrice.

- Paramètres intermédiaires : nom du fichier contenant la donnée à extraire.

- Paramètres de sortie : La matrice T de dim. (n, m).

- Tests. La procédure envisage deux tests:

- Test d'erreur d'ouverture de fichier.

- Test d'erreur de fermeture de fichier.

Procedure SommeMat (A, B; n, n; S).

Cette procédure effectue la somme de deux matrices de dim. compatibles.

Procedure differenceMat (A, B; n, n; D).

Cette procédure effectue la différence de deux matrices de dim. compatibles.

Procedure ProduitMat (A, B; n, m, u; C).

Cette procédure effectue le produit de deux matrices de dim. compatibles.

Procedure InverseMat (A; n, var B).

Cette procédure effectue l'inverse d'une matrice donnée A.

- Paramètres d'entrée : La matrice A de dim. (n, n)

- Paramètres de sortie : La matrice B de dim. (n, n).

Procedure TransposeMat (T; n, u; Mt).

Cette procédure effectue la transposée d'une matrice donnée T.

- Paramètres d'entrée : La matrice T de dim. (n, u)

- Paramètres de sortie : La matrice Mt de dim. (u, n).

Procedure Scalaire (Scal; Matrice; n, n; Result).

Cette procédure effectue le produit d'une constante avec une matrice.

Procedure CalculCoef (A; n; Coef).

Cette procédure effectue le calcul des coefficients du polynôme caractéristique.

- Paramètres d'entrée : La matrice A de dim.(n,n)
- Paramètres de sortie : Le vecteur contenant les coefficients de dim. (n+1).
- Procédures utilisées : * Scalaire
* SommeMat
* ProduitMat

Procédure Opération (A; Ck; var K; n; sign).

C'est une sous-routine qui effectue des calculs des éléments d'une matrice K d'une façon itérative.

- Paramètres d'entrée : Le réel A, la matrice Ck de dim.(n,n)
- Paramètres de sortie : La matrice K d'ordre n.

Function Puissance (power).

Cette procédure effectue la puissance $n^{\text{ème}}$ de l'entier (-1).

Procédure CalculMatriceK (Cas, T, var K; n).

Cette procédure calcule les éléments de la matrice K utilisée dans la détermination des grammiens de l'algorithme de Jameson.

- Paramètres d'entrée : La sélection cas, la matrice T de rang n.
- Paramètres de sortie : Matrice K de dim. (n,n).
- Procédures utilisées : * TransposeMat
* ProduitMat
* DifferenceMat
* Operation

Procédure CalculMatriceRouth (Routh).

Ce sous-programme effectue le calcul des éléments de la matrice de Routh.

- Paramètres intermédiaires : Vecteur des coefficients.
- Paramètres de sortie :
- Tests : La procédure envisage un test et un message d'erreur quand $Routh(i-1)=0$.

Procédure CalculMatriceInverseDeG (var InvG; n).

Cette procédure effectue le calcul des éléments de G^{-1} .

- Sous-Programmes utilisés : * ProduitMat
* CalculMatriceRouth
* Scalaire
* SommeMat
- Tests : La procédure envisage un test d'arrêt lorsque $Routh(n+1,1)$ est nul.

Procédure CalculDeX (Cas; T; var X; n).

Cette procédure calcule le grammien contenu dans la matrice X

d'ordre n.

- Sous-Programmes utilisés : * CalculMatriceK
* CalculMatriceInverseDeG
* ProduitMat.

Procédure Cholesky (n; A; var L).

Cette sous-routine donne pour une matrice donnée A, sa matrice triangulaire inférieure de Cholesky, contenue dans la variable L.

Procédure OrdonneMat (var T; n).

Cette sous-routine permet l'ordonnement des éléments d'une matrice en décroissance.

Procédure SigmaPuissMoinsUnDemi (A; var B; n).

Cette procédure effectue les opérations d'inversion, racine carrée, sur les éléments d'une matrice diagonale contenue dans la variable A.

Procédure TestOrdreReduc.

Cette procédure effectue un test sur l'ordre du système.

Procédure PuissanceMat.

Cette procédure effectue les puissances successives de la matrice.

Procédure LireVecteursPropres (Matrice; n).

Cette procédure permet la lecture des vecteurs propres qui sont extraits d'un quelconque fichier ASCII provenant de EIGEN2*

Procédure LireValeursPropres (propres; n).

Cette procédure permet la lecture des valeurs propres qui sont extraits d'un quelconque fichier ASCII provenant de EIGEN2*

Procédure CalculGramNumer (Cas; T; var X; n).

Cette procédure calcule les gramiens dans le cas numérique.

- Sous-Programmes utilisés: * ProduitMat
* SommeMat
* DifferenceMat.

* Le sous-programme **Eigen** faisant partie du logiciel EISPACK a été transcrit selon la procédure exposée dans:

Num.Math.13,293-304 (1969) By Parlett & Reinsch
Handbook for Auto.Comp., Vol II-Linear Algebra,
315-326 (1971).


```
Begin (du programme principal)
```

```
20:clrscr;  
Gotoxy(5,1); Write('Ecole Nationale Polytechnique.');
```

```
Gotoxy(5,2); Write('Departement d'Electronique.');
```

```
Gotoxy(15,5); Write(' - MODEL REDUCTION VIA BALANCED REALISATION -');
```

```
Gotoxy(15,6); Write(' PRESENTED BY : ADAMOU AMEL B.H.');
```

```
Gotoxy(15,10);Write('SYSTEME CONTINU TAPPEZ SUR LA TOUCHE F1 :');
```

```
Gotoxy(15,12);Write('SYSTEME DISCRET TAPPEZ SUR LA TOUCHE F2 :');
```

```
Gotoxy(15,14);Write('POUR ANNULER TAPPEZ SUR ESC :');
```

```
Gotoxy(15,25);Write('VOTRE CHOIX S.V.P :');
```

```
x:=readkey;
```

```
If x=#27 Then
```

```
begin
```

```
clrscr;
```

```
exit;
```

```
end
```

```
Else
```

```
begin
```

```
if x=#59 then goto 21 ;
```

```
if x=#60 then goto 22 else goto 20 ;
```

```
end;
```

```
21:clrscr;
```

```
(***** Systemes Continus *****)
```

```
Begin  
Write('Entrer la valeur de n, nombre d'Etats < ',MaxElements, ':');  
Readln(n);  
Assign(FileName,'ordre.dat');  
Rewrite(FileName);  
Write(FileName,N);  
Close(FileName);  
Write('Entrer la valeur de m, nombre d'Entrees < ', MaxElements, ':');  
Readln(m);  
Write('Entrer la valeur de u, nombre de Sorties < ', MaxElements, ':');  
Readln(u);  
Identite ( Id , n );  
LireMat ( A,n,n );  
LireMat ( B,n,m );  
LireMat ( C,u,n );  
DataInFile ( A,n,n );  
DataInFile ( B,n,m );  
DataInFile ( C,u,n );  
(DataOutFile ( A,n,n );  
DataoutFile ( B,n,m );  
DataoutFile ( C,u,n );)  
CalculCoef(a,n,coef);  
TransposeMat ( B, n, m, GramG );  
ProduitMat ( B, GramG, n, m,n, GramG );  
Scalaire ( (-1), GramG, n, n, GramG );  
CalculDeX ( true, GramG, GramG, n );  
DataInFile ( GramG,n,n );  
Readln;  
Cholesky ( n, GramG, Lr);  
TransposeMat ( Lr,n,n,Lrt);  
TransposeMat ( C,u,n,GramO );  
ProduitMat ( GramO,C,n,u,n,GramO );  
Scalaire ( (-1), GramO,n, n, GramO );  
CalculDeX ( false, GramO, GramO, n );  
DataInFile ( GramO,n,n );  
TransposeMat ( A,n,n,A );
```

```

Cholesky ( n, GramO, T );
TransposeMat ( T, n, n, Lot );
ProduitMat ( Lot, Lr, n, n, n, R );
TransposeMat ( R, n, n, Rtranspose );
ProduitMat ( R, Rtranspose, n, n, n, RRt );
ProduitMat ( Rtranspose, R, n, n, n, R+R );
DataInfile( RRt, n, n );
DataInfile ( R+R, n, n );
Swapvectors ;
  Exec('eigen1.exe', '' );
Swapvectors;
LireVecteursPropres ( G, n );
LireValeursPropres ( Propres, n );
Diagonale( Sigma, n );
Writeln ( ' La Matrice sigma Diagonale ' );
EcrireMat( Sigma, n, n );
DataInfile( Sigma, n, n );
Readln;
SigmaPuissMoinsUnDemi ( Sigma, D, n );
TestOrdreReduc ( Sigma, n );
TransposeMat ( G, n, n, G );
ProduitMat ( D, G, n, n, n, It );
ProduitMat ( It, Lot, n, n, n, It );
InverseMat ( It, n, Tbal );
Writeln ( ' La Matrice Tbal est ' );
EcrireMat ( Tbal, n, n );
DataInfile ( Tbal, n, n );
ProduitMat ( It, A, n, n, n, A );
ProduitMat ( A, Tbal, n, n, n, A );
Writeln( ' Matrice A Reduite est ' );
EcrireMat ( A, k, k );
Readln;
ProduitMat ( It, B, n, n, m, B );
Writeln( ' Matrice B Reduite est: ' );
EcrireMat ( B, k, m );
Readln;
ProduitMat ( C, Tbal, u, n, n, C );
Writeln( ' Matrice C Reduite est: ' );
EcrireMat ( C, u, k );
Readln;
DataInFile ( A, n, n );
DataInFile ( B, n, m );
DataInFile ( C, u, n );
Delay(4000);
Clrscr;
Gotoxy(2,10);
Write(' Si vous voulez reduire un autre systeme taper sur une touche
      'sinon ESC' );
x:=readkey;
If x=#27 Then halt(1) Else goto 20;

(***** Systemes Discrets *****)

```

```
2:Clrscr;
```

```
begin
```

```

Write(' Entrer la valeur de n, nombre d' Etats < ', MaxElements, ' ');
Readln(n);
Assign(FileName, 'ordre.dat' );
Rewrite(FileName);
Write(' ');
Close(FileName);

```

```

Write('Entrer la valeur de m, nombre d'Entrees <', MaxElements,':');
Readln(m);
Write('Entrer la valeur de u, nombre de Sorties <', MaxElements,':');
Readln(u);
Identite ( Id , n );
LireMat ( Phi,n,n );
LireMat ( Gamma;n,m );
LireMat ( H,u,n );
DataInFile ( Phi,n,n );
DataInFile ( Gamma,n,m );
DataInFile ( H,u,n );
(DataOutFile ( Phi,n,n );
DataoutFile ( Gamma,n,m );
DataoutFile ( H,u,n );)
CalculCoef(Phi,n,coef);
TransposeMat ( Gamma, n, m,Tomp);
ProduitMat ( Gamma, Tomp, n, m,n, Tomp );
CalculGramNumer( True, Tomp, GramG, n);
DataInFile ( GramG,n,n );
Cholesky ( n, GramG, Lr);
TransposeMat ( Lr,n,n,Lrt);
TransposeMat ( H, u, n, Tamp );
ProduitMat ( Tamp, H, n, u, n, Tamp );
CalculGramNumer ( False, Tamp, GramO, n);
DataInFile ( GramO, n, n );
TransposeMat ( Phi, n, n, A );
Cholesky ( n, GramO, T );
TransposeMat ( T, n, n, Lot );
ProduitMat ( Lot, Lr, n, n, n, R );
TransposeMat ( R, n, n, Rtranspose );
ProduitMat ( R, Rtranspose, n, n, n, RRt);
ProduitMat ( Rtranspose, R, n,n, n, RtR );
DataInfile( RRt,n,n);
DataInfile ( RtR,n,n);
Swapvectors ;
Exec('eigen1.exe', '');
Swapvectors ;
LireVecteursPropres ( G,n );
LireValeursPropres ( Propres,n );
Diagonale( Sigma,n);
Writeln ( ' La Matrice sigma Diagonale ');
EcrireMat( Sigma,n,n );
DataInfile( sigma,n,n);
Readln;
SigmaPuissMoinsUnDemi ( Sigma, D, n );
TestOrdreReduc ( Sigma, n);
TransposeMat ( G, n, n, G );
ProduitMat ( D, G, n, n, n, It );
ProduitMat ( It, Lot, n,n,n, It );
InverseMat ( It, n, Tbal );
Writeln ( ' La Matrice Tbal est ');
EcrireMat ( Tbal, n, n );
DataInfile ( Tbal, n, n );
ProduitMat ( It, Phi, n, n, n, Phi);
ProduitMat ( Phi, Tbal, n, n, n, Phi);
Writeln(' Matrice Phi Reduite est ');
EcrireMat ( Phi, k, k );
Readln;
ProduitMat ( It, Gamma, n, n, m, Gamma );
Writeln(' Matrice Gamma Reduite est: ');

```

```
EcrireMat ( Gamma, k, m ) ;
Readln;
ProduitMat ( H, Tbal, u, n, n, H );
Writeln(' Matrice H Reduite est: ');
EcrireMat ( H, u; k ) ;
Readln;
DataInFile ( Phi, n, n);
DataInFile ( Gamma, n, m );
DataInFile ( H, u, n );
Readln;
Clrscr;
Gotoxy(2,10);
Write('Si Vous voulez reduire un autre Systeme taper sur une touche
      sinon ESC');
x:=readkey;
If x=#27 Then halt(1) Else goto 20;
```

nd.

UNIT Maths;

(* ----- *) INTERFACE (* ----- *)

(\$L-)

Const

MaxElements =15;
Max = MaxElements + 1;
Max2 = Max;

Type

mat= array[1..Max,1..Max] of real;
VecteurLigne=array[1..Max] of real;
VecteurColonne = array [1..MaxElements] of real;

Var

Selec : String[8];
X : Char;
n,m,u,k : Integer;
Id,A,B,C,GramO,R,Lr,T,Lot,Tbal,IT,Rtranspose,matu,matv,RtR,RRt,Phi,
Gamma,H,tomp,GramG,O,G,Sig,Lrt,tamp1,tamp,sigma,D,Result: Mat
V ,coef : VecteurLigne ;
choix : Boolean;
NomExterne : String[17];
Egval,PROPRES, ValeursPropres : VecteurColonne;
FileName : Text;

Procedure Identite (var Id : mat; n,m : integer);

Procedure AffectationMat (A:Mat;n,u: Integer; Var B:Mat);

Procedure Liremat(var t:mat ; n,u:integer);

Procedure EcrireMat(var t:mat ; n,u:integer);

Procedure SommeMat(a,b:mat; n,u: integer; VAR s : mat) ;

Procedure DifferenceMat(a,b:mat; n,u: integer; VAR d : mat) ;

Procedure CalculCoef (var A:mat; n:integer; Var Coef:VecteurLigne);

Procedure DataInFile(var T: Mat; n,u:integer);

Procedure DataOutFile(var T: Mat; n,u:integer);

Procedure ProduitMat(a,b:mat; n,m,u:Integer; VAR c: mat);

Procedure InverseMat(a:mat; n:Integer; VAR b: mat);

Procedure transposemat(t:mat;n,u:integer; VAR mt: mat);

Procedure Scalaire (scal : real; matrice : mat; n,u : integer; VAR result : m

c);

Procedure CalculDeX (Cas : boolean; T : mat; VAR X : mat; n: integer);

Procedure CalculGramNumer (Cas : boolean; T : mat; VAR X : mat; n: integer);

Procedure Cholesky(n :integer;a:mat;var L:mat);

Procedure Maximum (Propres: VecteurColonne; n:Integer; Start:Integer;
var Max: Real; Var Index : Integer);

Procedure LineVecteursPropres (Var MATRICE : Mat; n : Integer);

Procedure LineValeursPropres (Var Propres : VecteurColonne; n : Integer);

Procedure SigmaPuissMoinsUnDemi(A: Mat; Var B:Mat; n:Integer);

Procedure TestOrdreReduc (A:Mat; n:Integer);

Procedure Diagonale(Var D:mat; n:integer);

Procedure PuissanceMat (A:Mat; power,n:integer; Var P: Mat);

(* ----- *) IMPLEMENTATION (* ----- *)

Procedure Identite (var Id : mat; n,m : integer);

var

i,j : integer;

begin

for i:= 1 to n do

for j:= 1 to n do

if i=j

```

then
    Id[i,j] := 1
else
    Id[i,j] := 0;
end;
*****
procedure AffectationMat ( A:Mat; n,u: Integer; Var B:Mat);
var
    i,j:Integer;
begin
    for i:= 1 to n do
        begin
            for j:=1 to u do
                B[i,j]:=A[i,j];
            end;
        end;
end;
*****
procedure liremat(var t:mat ; n,u:integer);
var
    i,j : integer;
begin
    for i:=1 to n do
        for j:=1 to u do begin
            write('entrez les valeurs de la matrice : ');
            write(' t [',i,',',j,'] = ');
            readln(t[i,j]); writeln;
        end;
    end;
end;
*****
procedure EcrireMat(var t:mat ; n,u:integer);
var
    i,j : integer;
begin
    for i:=1 to n do
        begin
            for j:=1 to u do write(t[i,j]:10:10,' ');
            writeln;
        end ;
    end;
end;
*****
procedure DataInFile( var T: Mat; n,u: integer);
var
    Fic:Text;
    i,j:integer;
begin
    write(' entrer le NomExterne du fichier ');
    Readln ( NomExterne );
    (* Assignment *);
    Assign ( Fic, NomExterne);

```

```

(* ouverture en creation *)
{ $I- }
rewrite(fic);
{ $I+ }
if ioresult<>0 then
    begin
        Writeln (' Error creation fichier ');
        Halt;
    end
else
    for i:=1 to n do
        begin
            for j:= 1 to u do
                Write ( Fic,T[i,j]:10:4);
                Writeln ( Fic,' ');
            end;
        end;

(* Fermeture fichier *)
{ $I- }
Close (fic);
{ $I+ }
if ioresult<>0
then
    writeln (' Erreur fermeture fichier ',NomExterne) ;
end;
*****
procedure DataOutFile ( Var T:mat ; n,u:integer);
var
    Fic:Text;
    i,j:integer;
begin
    write(' entrer le NomExterne du fichier ');
    Readln ( NomExterne );

    (* Assignment *);
    Assign ( Fic, NomExterne);

    (* Ouverture en Consultation *)
    { $I- }
    Reset ( Fic );
    { $I+ }
    if ioresult<>0 then
        begin
            Writeln (' Error ...Fin ');
            Halt;
        end
    else
        begin
            While not eof ( Fic ) do
                begin
                    (* Lecture du fichier de Donnees *)
                    for i:=1 to n do
                        begin
                            for j:=1 to u do
                                Read ( Fic,T[i,j]);
                                Readln ( Fic );
                            end;
                        end;
                    Readln ( T,n,u) ;
                end;
        end;

```

```

end;
end;

(* Fermeture fichier *)
($I-)
Close (fic);
($I+)
if ioresult<>0
then
  writeln (' Erreur fermeture fichier ',NomExterne) ;
End;
(*****)
Procedure SommeMat( a,b:mat; n,u: integer; VAR s : mat ) ;
Var
  i,j : integer;
Begin
  for i:=1 to n do begin
    for j:=1 to u do
      s[i,j]:= a[i,j]+b[i,j] ;
    end;
  end;
End;
(*****)

Procedure DifferenceMat( a,b : mat; n,u:integer; VAR d : mat);
Var
  i,j : integer;
Begin
  for i:=1 to n do begin
    for j:=1 to u do
      d[i,j]:=a[i,j]-b[i,j] ;
    end;
  end;
End;
(*****)

Procedure ProduitMat(a,b:mat;n,m,u:integer; VAR c: mat);
Var
  i,j,v : Integer;
  S:Real;
Begin
  for i:=1 to n do
    for j:=1 to u do
      begin
        S:= 0.0;
        for v:= 1 to m do
          S:= S+ a[i,v]* b[v,j] ;
        end;
        c[i,j]:=S;
      end;
    end;
  end;
End;
(*****)

Procedure CalculCoef ( var A:mat; n:integer; Var Coef:Vecteurligne);
Var
  i,j,k : Integer;
  TrAk,x: real;
  Ak,Id,Tamp,Tamp1:Mat;
Begin
  Coef[1]:=1;
  for i:=2 to n do Coef[i]:=0.0;

```

```

matamp[k,j]:= matamp[k,j]/matamp[k,k];
for i:=1 to n do
  begin
    if i<>k
    then
      begin
        for j:=2*n downto k do
          matamp[i,j]:=matamp[i,j]-matamp[i,k]*matamp[k,j];
        end;
      end;
  end;
end;
for i:=1 to n do
  begin
    for j:= n+1 to 2*n do
      b[i,j-n]:= matamp[i,j];
    end;
  end;
nd;

```

*****)

```

procedure Operation (a : Real; ck : mat; VAR k : mat; n,sign : integer);
ar
  i,j : integer;

```

```

egin
  for i:= 1 to n do
    for j:= 1 to n do
      begin
        (* affecte les nouvelles valeurs a la matrice K *)
        k[i,j] := k[i,j]+ sign*ck[i,j] * a;
      end;
    end;
  end;

```

*****)

```

procedure Transposemat(t:mat;n,u:integer; VAR mt: mat);

```

```

ar
  i,j : integer;
egin
  for i:=1 to u do
    for j:=1 to n do
      mt[i,j]:= t[j,i];
    end;
  end;

```

*****)

```

function Puissance ( power : integer ) : integer ;

```

```

r
  h,i : integer;
gin
  h := 1;
  for i:= 1 to power
    do h := h * (-1);
  end;

  puissance := h;

```

*****)

```

cedure CalculMatriceK ( Cas: boolean; T: mat; VAR K: mat; n:integer );

```

COMMENTAIRE :

```

*)
var
  i,j,l: integer;
  MatriceIterativeC, Temporaire1, Temporaire2, Temporaire3, P : mat;
  sign:integer;
begin
  Sign:= puissance (n);
  for i:= 1 to n do
    for j:= 1 to n do
      begin
        MatriceIterativeC[i,j] :=0.0;
        K[i,j] := 0.0;
      end;
  P:= Id;
  If Cas = true (* Cas ou X = P *)
  then
    TransposeMat ( A, n, n, Temporaire3)
  Else
    Begin (* Cas ou X = P *)
      TransposeMat ( A, n, n, A );
      TransposeMat ( A, n, n, Temporaire3 );
      TransposeMat ( Temporaire3, n, n, A );
    End;
  for i:= 1 to n do
    begin
      sign:=sign*(-1);
      ProduitMat( P, T, n, n, n, Temporaire1 );
      ProduitMat(MatriceIterativeC,Temporaire3, n,n,n,Temporaire2);
      DifferenceMat(Temporaire1,Temporaire2,n,n,MatriceIterativeC);
      (* genere la valeur du vecteur K *)
      Operation (coef[n-i+1], MatriceIterativeC, K, n,sign);
      ProduitMat ( P; A, n, n; n, P );
    end;
  end;

```

```

End;
(*****

```

```

Procedure CalculMatriceRouth ( VAR Routh : Mat );
var
  i,j : integer;
Begin
  for j:=1 to (n-1) do
    begin
      Routh[1,j] := coef [(2*j-1)];
    end;
  Routh[1,n]:= 0.0;
  for j := 1 to (n-2) do
    Routh[2,j] := coef [(2*j)];
  Writeln;
  for j:= (n-1) to (n) do
    Routh[2,j] := 0.0;
  Writeln;
  (* Calcul des autres Routh [i,j] *)
  For i:= 3 to (n+1) do
  Begin
    For J:= 1 to (n-2) do
      Begin
        if Routh[i-1,1] = 0
          then
            begin
              writeln (' Routh[i-1, 1] =0 .... Erreur ...');
            end;

```

```

for i:=1 to n do
  for j:= 1 to n do
    Ak[i,j]:=0;
Identite(id,n,n);
for k:=1 to n do
  begin
    x:=coef[k];
    Scalaire ( x, Id, n,n, Tamp );
    SommeMat ( Ak, Tamp,n, n,Tamp1 );
    ProduitMat ( Tamp1,a, n,n,n,Ak );
    begin
      trak:=0;
      for i:=1 to n do
        trAk:=trAk+Ak[i,i];
      end;
      Coef[k+1]:=-1*trAk/k;
    end;
  end;
End;
*****

```

```

Procedure InverseMat( a:mat; n:Integer; VAR b: mat );
Var
  i,j,k : integer;
  tamp : real;
  matamp: array[1..maxelements, 1..2*maxelements] of real;
Begin
  for i:=1 to n do
    begin
      for j:=1 to n do
        matamp[i,j]:=a[i,j];
      end;
      for i:=1 to n do
        begin
          for j:= (n+1) to 2*n do
            begin
              if i= (j-n)
                then matamp[i,j]:=1
                else matamp[i,j]:=0;
            end;
          end;
          for k:=1 to n do
            begin
              if matamp[k,k]=0
                then
                  begin
                    i:=0;
                    repeat
                      i:=i+1;
                      if matamp[k+1,k]<>0
                        then
                          for j:=k to 2*n do
                            begin
                              tamp:=matamp[k,j];
                              matamp[k,j]:=matamp[k+i,j];
                              matamp[k+i,j]:=tamp;
                            end;
                          until ( matamp[k+i,k]<>0 );
                          if matamp[k,k]=0
                            then Exit;
                        end;
                    end;
                  end;
            end;
          for j:=2*n downto k do

```

```

                HALT;
            end;
            Routh[i,j]:=Routh[(i-2),(j+1)]-(Routh[(i-1),(j+1)]*
            (Routh[(i-2),1]/Routh[(i-1),1] ));
        end;
        writeln;
    end;
end;
id;
*****
procedure Scalaire (scal:real;matrice:mat; n,u:integer; VAR result:mat );
var
    i, j : integer;
begin
    for i:= 1 to n do
        for j:=1 to u do
            result [i,j] := scal * matrice [i,j];
        end;
    end;
*****

procedure CalculMatriceInverseDeG ( VAR InvG : mat; n: integer );
var
    iter, j : integer;
    Routh, Ha, Hb, Hc, Tempo : mat;
    Raport : Real;
begin
    If n <= 2
    Then
        Hc := Id
    Else
        Begin
            CalculMatriceRouth ( Routh );
            Ha := Id;
            Hb := Id;
            Hc := Id;
            for iter:= 3 to ( N+1 ) do
                Begin
                    Ha := Hb;
                    Hb := Hc;
                    ProduitMat ( A, Hb, n, n,n, Tempo );
                    if Routh[(iter-1),1] = 0
                    then
                        begin
                            writeln ('Routh[i-1, 1] = 0 .... Erreur ...');
                            HALT;
                        end
                    else
                        begin
                            Raport:= Routh[(iter-2),1]/Routh [(iter-1),1];
                            Scalaire ( Raport, Tempo, n,n, Hc );
                            SommeMat ( Ha, Hc, n, n, Tempo );
                            Hc := Tempo;
                        end ;
                    end;
                end;
            End;
        End;
    End;
end;

(* REMARQUE : La matrice de l'iteration n+1 est Hc *)
(* Connaissant Routh, finale ie H de l'iteration N+1 on deduit
la matrice inverse de G *)

```

```

IF Routh[N+1,1]= 0
then
begin
    writeln ( ' Routh[N+1,1] =0 .... ERREUR ... ');
    HALT;
end;
Scalaire ( puissance (n) / ( 2 * Routh[N+1,1] ), Hc, n, n, InvG );
nd;
*****

procedure CalculDeX ( Cas: boolean; T: mat; VAR X : mat; n: integer);
var
    InvG , MatriceK : Mat;
begin
    CalculMatriceK ( Cas , T, MatriceK,n);
    CalculMatriceInverseDeG ( InvG,n);
    ProduitMat ( InvG, MatriceK, n, n ,n, X );
end;
*****

procedure Cholesky ( n:integer; A:mat; var L:mat );
var
    i,j,v : integer;
    H : boolean;
    cumul : REAL;
begin
    for i:=1 to n do
        for j:= 1 to n do
            L[i,j] := 0.0;

        if a[i,1] <= 0
        then
            begin
                writeln ( ' Erreur... A est negative ..');
                halt;
            end;

        for i:=1 to n do
            begin
                cumul := 0.0;
                for j:=1 to (i-1) do
                    cumul := cumul + L[i,j] * L[i,j];
                    L[i,i] := sqrt ( (a[i,i] - cumul) );
                    for j:= 2 to n do
                        begin
                            cumul := 0;
                            for v:=1 to (i-1) do
                                cumul := cumul + (L[i,v]* L[j,v]);
                                L[j,i]:= (a[i,j] - cumul) / (L[i,i]);
                            end;
                        end;
                    end;
            end;
        end;
    end;
*****

procedure LireValeursPropres( Var Propres : VecteurColonne; n : Integer );
var
    FileToEdit : String [14];
    FileFromEigen3 : TEXT ;
    STR : String [17];
    VAL,s : REAL;

```

```

I           : Integer;
Ok          : Boolean;
Reponse     : Char;
egin
  (* lecture du fichier de donnees resultats de EIGEN3 *)
  Write ('Entrez le nom du fichier contenant les valeurs propres : ');
  Readln ( FileToEdit );
  (* ASSIGNATION *)
  Assign (FileFromEigen3, FileToEdit);
  (* Ouverture en consultation *)
  Ok:=false;
  Repeat
  {#I-}
  RESET ( FileFromEigen3 );
  {#I+}
  If ioresult <> 0
  then
    begin
      writeln (' ERREUR OUVERTURE FICHER ');
      HALT;
    end
  else
    ok:=true;
  until ok;

  (* Ecouler la lignes inutile du le fichier FileFromEigen3 *)
  READLN ( FileFromEigen3, STR );
  I := 1;
  repeat
    (* lecture du coef *)
    {#I-}
    READLN( FileFromEigen3 , VAL );
    {#I+}
    if ioresult <> 0
    then begin
      writeln('Erreur...Format fichier non compatible... ');
      halt;
      end;
    (* affichage du coef *)
    WRITELN ( VAL );

    (* memorisation de la partie reelie avec la valeur VAL *)
    COEF [I] := VAL;
    I:= I+1;
  until ( I = N + 1);

  (* Fermeture Fichier *)
  {#I-}
  Close ( FileFromEigen3 );
  {#I+}
  If ioresult<>0
  .Then Writeln(' Erreur Fermeture Fichier ', FileToEdit )
End;
(*****:*****)

Procedure LireVecteursPropres ( Var MATRICE : Mat; n : Integer );
VAR (* la matrice MATRICE contient les vecteurs propres
qui seront extrait d'un quelconque fichier ADU11
provenant de EIGEN3 *)

```

```

FileToEdit      : String [14];
FileFromEigen3  : TEXT ;
START          : STRING[17];
STR            : STRING[12] ;
VAL            : REAL;
I,J            : Integer;
Reponse        : Char;
Ok             : Boolean;
begin
  (* lecture du fichier de donnees resultats de EIGEN3 *)
  Write ('Entrez le nom du fichier contenant les vecteurs propres : ');
  Readln ( FileToEdit );

  (* ASSIGNATION *)
  Assign (FileFromEigen3, FileToEdit);

  (* Ouverture en consultation *)
  Ok:=false;
  repeat
    {#I-}
  RESET ( FileFromEigen3 );
    {#I+}
  If ioresult <> 0
  then
    writeln (' ERREUR OUVERTURE FICHER ');
  else
    Ok:=true;
until Ok;
  (* Ecouler la lignes inutile du le fichier FileFromEigen3 *)
  I := 1;
  Repeat
    READLN ( FileFromEigen3, STR );
    I:= I+1;
  Until (I = N+4);
  I := 1;
  repeat
    (* lecture du vecteur *)
    J := 0;
    REPEAT
      {#I-}
      READ( FileFromEigen3 , VAL );
      {#I+}
      if ioresult <> 0
      then begin
        writeln('Erreur...Format fichier non compatible...');
        halt;
        end;
      MATRICE [i,J+1] := VAL;
      J := J+1;
    UNTIL (J = N-1);

    {#I-}
    READLN ( FileFromEigen3, VAL );
    {#I+}
    if ioresult <> 0
    then begin
      writeln ('Erreur...Format fichier non compatible...');
      halt;
      end;
    MATRICE[1,N] := VAL;

```

```
(* affichage du ccef *)
```

```
(* memorisation de la partie reelle avec la valeur VAL *)
```

```
I:= I+1;
```

```
until ( I = N + 1 );
```

```
EcrireMat ( MATRICE ,n, n );
```

```
Readln ( I );
```

```
(* Fermeture Fichier *)
```

```
{#I-}
```

```
Close ( FileFromEigen3 );
```

```
{#I+}
```

```
If ioresult<>0
```

```
Then Writeln(' Erreur Fermeture Fichier ', FileToEdit )
```

```
end;
```

```
*****
```

```
procedure Maximum ( Propres : VecteurColonne; n : integer; Start : integer;  
VAR Max : real; VAR Index : integer );
```

```
var
```

```
  i : integer;
```

```
begin
```

```
  for i:=start to n do
```

```
    begin
```

```
      Propres[i]:=0;
```

```
      Propres[start]:=0;
```

```
    end;
```

```
  Max:= Propres[Start];
```

```
  for i:= Start to n do
```

```
    if Propres[i] >= Max
```

```
    then
```

```
      begin
```

```
        Max := Propres[i];
```

```
        Index := i;
```

```
      end;
```

```
end;
```

```
*****
```

```
procedure Echange ( Var Reel1 : real ; VAR Reel2 : Real );
```

```
var
```

```
  Tempo : real;
```

```
begin
```

```
  Tempo := Reel1;
```

```
  Reel1 := Reel2;
```

```
  Reel2 := Tempo;
```

```
end;
```

```
*****
```

```
procedure OrdonneMat( var T:mat; n:integer );
```

```
var
```

```
  i,j:integer;
```

```
begin
```

```
  for i:=1 to (n-1) do
```

```
    for j:=(i+1) to n do
```

```
      begin
```

```
        if T[i,i] <= T[j,j]
```

```
        then Echange ( T[i,i],T[j,j] );
```

```
      end;
```

```
end;
```

```
(*****)
```

```
Procedure Diagonale( Var D:mat; n:integer );  
VAR
```

```
  s:Real;  
  Propres:vecteurColonne;  
  FileToEdit : String [14];  
  FileFromEigen3 : TEXT ;  
  STR: STRING [17];  
  VAL: REAL;  
  i,j: Integer;  
  Ok : Boolean;  
  Reponse: Char;
```

```
Begin
```

```
(* lecture du fichier de donnees resultats de EIGEN3 *)
```

```
Write ('Entrez le nom du fichier contenant les valeurs propres : ');  
Readln ( FileToEdit );
```

```
(* ASSIGNATION *)
```

```
Assign (FileFromEigen3, FileToEdit);
```

```
(* Ouverture en consultation *)
```

```
Ok:=false;
```

```
Repeat
```

```
($I-)
```

```
RESET ( FileFromEigen3 );
```

```
($I+)
```

```
If ioresult <> 0
```

```
then
```

```
  begin
```

```
    writeln (' ERREUR OUVERTURE FICHER ');
```

```
    writeln (' voulez-vous abandonner oui/non ');
```

```
    readln(reponse);
```

```
    if reponse='n'
```

```
    then
```

```
      begin
```

```
        writeln('Message de Fin');
```

```
        HALT;
```

```
      end
```

```
    else
```

```
      ok:=false;
```

```
  end
```

```
else
```

```
  ok:=true;
```

```
until ok;
```

```
(* Ecouler la lignes inutile du le fichier FileFromEigen3 *)
```

```
READLN ( FileFromEigen3, STR );
```

```
I := 1;
```

```
repeat
```

```
  (* lecture du coef *)
```

```
  ($I-)
```

```
  READLN( FileFromEigen3 , VAL );
```

```
  ($I+)
```

```
  if ioresult <>0
```

```
  then begin
```

```
    writeln('Erreur...Format fichier non compatible...');
```

```
    halt;
```

```
  end;
```

```
  (* affichage du coef *)
```

```
  WRITELN ( VAL );
```

```

(* memorisation de la partie reelle avec la valeur VAL *)
COEF [I] := VAL;
I:= I+1;

```

```

until ( I = N + 1);writeln;
for i:= 1 to n do
begin
s:=coef[i];
s:=sqrt(s);
coef[i]:=s;
end;
for i:=1 to n do
for j:=1 to n do
begin
if i<>j
then d[i,j]:=0
else
d[i,i]:=coef[i];
end;
OrdonneMat( d,n );
EcrireMat ( d,n,n);
writeln;

```

```

(* Fermeture Fichier *)
($I-)
Close ( FileFromEigen3 );
($I+)
If iorresult<>0
Then Writeln(' Erreur Fermeture Fichier ', FileToEdit );

```

```

End;
(*****)

```

```

Procedure SigmaPuissMoinsUnDemi(A:Mat;Var B:mat; n:Integer);
VAR
i,j:integer;
s:Real;
Begin
for i:=1 to n do
for j:=1 to n do
begin
if i<>j
then
B[i,j]:=0
else
begin
s:= A[i,i];
s:= Sqrt(s);
s:= 1/s;
B[i,i]:= s;
end;
end;
end;

```

```

End;
(*****)

```

```

Procedure TestOrdreReduc ( A:Mat; n:Integer ) ;
Const
Epsilon= 1E-3;
Var
Som1,Som2,Norm1, Norm2, Compar, Tamp: Real;
Ratio :VecteurColonne;
i,i:Integer;

```

```

Fich : Text;
Begin
  k:=1;
  Assign ( Fich, 'Reducio.dat' );
  Rewrite ( Fich );
  Repeat
    Som1:=0;
    for i:=1 to k do
      Som1:= Som1 + sqr(sqr(A[i,i]));
      Norm1:=Sqrt(Som1);
    Som2:=0;
    for k:=i+1 to n do
      Som2:= Som2 + sqr(sqr(A[k,k]));
      Norm2:= Sqrt(Som2);
      Ratio[i]:= Norm2/ Norm1;

    j:=0;
    Tamp:=0;
    Repeat
      Writeln ( fich, i:6, ' ', Tamp:15:4 );
      j:=j+1;
      Tamp:= Ratio[i];
      If j=2
      Then
        Tamp:=0;

    Until j=3;
    k:=i+1;
  Until ( i = n-1 );
Close (fich);
i:=1;
Repeat
  Compar:= Ratio[i] - Ratio[i+1];
  If i=n-1
  Then Compar := Epsilon;
  i:= i+1;
Until ( Compar<= Epsilon );
k:=i-1;
End;
(*****)

Procedure PuissanceMat ( A:Mat ;power,n:integer; Var P: Mat);
Var
  i:integer;
Begin
  Identite( Id,n,n);
  P:=id;
  for i:=1 to Power do
    ProduitMat ( P, A, n, n, n, P);
End;
(*****)

Procedure CalculGramNumer ( Cas:boolean;T:mat;VAR X: mat; n:integer );
(* COMMENTAIRE:
  La matrice T:= - B.Bt ( si P est l'inconnue ),
  T:= - Ct.C ( si Q est l'inconnue ),
  MatriceIterativeC est la matrice C la kieme iteration ;
*)

Const
  Epsi=1.E-7;
Var
  i,j,nmax: integer;
  Temp1,Temp2,Temp3,Temp,F,Hola,EpsiMat,At,P,result1: mat;
Begin
  For i:=1 to n do

```

```

for j:=1 to n do
  X[i,j]:=0;
If Cas = False (* Cas ou X = P *)
Then
  begin
    TransposeMat ( A, n, n, A );
    TransposeMat ( A, n, n, Temp3 );
    TransposeMat ( Temp3, n, n, A );
  end;
  AffectationMat(T,n,n,X);
  TransposeMat ( A, n, n, Temp1);
  Identite ( Id, n, n );
  Scalaire( Epsi, Id,n ,n, EpsiMat );
  For i:=1 to n do
    for j:=1 to n do
      begin
        nmax:= 1;
        Repeat
          begin
            PuissanceMat ( A ,nmax, n , Result );
            PuissanceMat ( Temp1, nmax, n, result1 );
            ProduitMat ( Result, T, n, n, n, Temp2 );
            ProduitMat ( Temp2, Result1, n, n, n, F);
            SommeMat ( X, F,n,n, X );
            TransposeMat ( A,n,n,At );
            ProduitMat ( A, X, n,n,n, Temp );
            ProduitMat ( Temp, At, n,n,n, Temp );
            SommeMat ( Temp, T,n,n, Temp );
            DifferenceMat ( Temp, X, n,n, Temp );
            If ( Temp [i,j] <= EpsiMat [i,j] )
            Then
              Exit
            Else
              nmax:=nmax+1;
            end
          Until ( Temp[i,j]<= Epsi ) ;
        end;
      end;
    end;
  end;

```

```

End;
(***** )
Begin
End.

```