

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

INGENIORAT D'ETAT EN ELECTRONIQUE

**ETUDE DE DEUX
METHODES DE REDUCTION
DES SYSTEMES COMPLEXES**

« APPLICATION A UN REACTEUR NUCLEAIRE »

Proposé par :

M. BETTAYEB

Chargé de Recherche au CEN

Etudié par :

Said MADI

Mokhtar Hamid MAGUEMOUN

ECOLE NATIONALE POLYTECHNIQUE

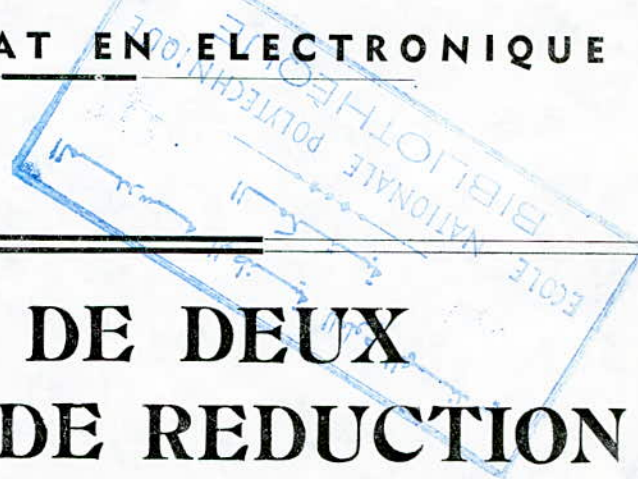
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

INGENIORAT D'ETAT EN ELECTRONIQUE

**ETUDE DE DEUX
METHODES DE REDUCTION
DES SYSTEMES COMPLEXES**

« APPLICATION A UN REACTEUR NUCLEAIRE »



Proposé par :

BETTAYEB

chargé de Recherche au CEN

Etudié par :

Said MADI

Mokhtar Hamid MAGUEMOUN

Dedicaces

A mon père, A ma mère
A mes frères et soeurs.
A mon petit neveu Ryad
A la mémoire de ma grand-mère Azizou.

Hamid.

A mon père, A ma mère
A mes frères Abdelhacim et M. Tahar
A mes soeurs et ma belle soeur Naceq
A ma petite niece Dalil et mon petit
neveu Basim
A tous mes amis

Said

REMERCIEMENTS

A l'issue de ce projet, nous tenons à remercier vivement MR M. BETTAYEB chargé de recherche au centre de développement et de la conservation de l'énergie pour avoir tout d'abord bien voulu nous accueillir dans son laboratoire et les conseils précieux qu'il a bien voulu nous prodiguer.

Nous tenons aussi à nous excuser auprès de lui pour certaines fois où la passion l'a emporté sur la raison.

Nos vifs remerciements sont accordés aussi à MRS DJEFFAL et HAMMADI pour l'aide précieuse qu'ils nous ont manifestée.

Enfin que tous ceux qui ont contribué de près ou de loin à l'élaboration de ce modeste travail trouvent ici l'expression de notre grande reconnaissance.

Hamid et Saïd

SOMMAIRE

AVANT-PROPOS

1 INTRODUCTION

1.1 GENERALITES

1.2 NECESSITE DE REDUIRE

1.3 APPLICATIONS ET MOTIVATIONS

1.4 APERÇU DES METHODES DE REDUCTION

1.4.1 Méthodes temporelles.

1.4.2 Méthodes fréquentielles

2 REDUCTION DE SYSTEMES LINEAIRES COMPLEXES

2.1 POSITION DU PROBLEME

- 2.2 REDUCTION ANALYTIQUE DE (A, B, C) EN SYSTEME CONTINU
- 2.3 REDUCTION ANALYTIQUE DE (A, B, C) EN SYSTEME DISCRET
- 2.4 REDUCTION DES PARAMETRES DE MARKOV: APPROCHE DE HANKEL
- 2.5 EQUIVALENCE DES DEUX METHODES

3

ALGORITHMES ETUDIES

- 3.1 REDUCTION DES VARIABLES D'ETATS DU SYSTEME (A, B, C)
 - 3.1.1 Algorithme de résolution des équations de Lyapunov par la méthode de Bingulac
 - 3.1.2 Algorithme de décomposition de Cholesky.
 - 3.1.3 Organigramme de réduction.
- 3.2 REDUCTION DES PARAMETRES DE MARKOV (Algorithme).
 - 3.2.1. Organigramme de réduction.
 - 3.2.2 Décomposition en composantes principales.

4

PROGRAMMATION et SIMULATION

4.1 PROGRAMMATION DETAILLEE

4.2 SIMULATIONS NUMERIQUES

5

CONCLUSIONS et INTERPRETATIONS

AVANT-PROPOS

Nous nous proposons d'étudier dans le cadre de ce mémoire, deux méthodes d'approximations de systèmes linéaires équilibrés

L'approximation d'un système linéaire dynamique représenté par les équations d'états :

$$\dot{x} = Ax + B\mu$$

$$y = Cx$$

et celle, à partir des paramètres de Markov ; (H_i) ; $i = 1, \dots, n$
avec $H_i = CA^{i-1}B$.

Nous présenterons dans un premier temps, l'étude théorique correspondante à chacune des deux méthodes, puis après, l'étude numérique et simulation des données à travers la programmation en langage FORTRAN IV.

Enfin une conclusion sera faite après l'étude numérique pour l'interprétation sur la qualité de l'approximation dans leur forme équilibrée.

chapitre 1

INTRODUCTION

1.1 GENERALITES

Le monde est complexe.

Beaucoup de systèmes pratiques dans le monde environnant qui nous sont proposés aujourd'hui sont très complexes.

L'analyse concrète et la conception de ces modèles de grandes dimensions est bien au-delà du domaine de traitement et d'exécution de contrôle du système.

Même, un système large, présenté comme une représentation de dimension très élevée, est-il incertain du fait des perturbations engendrées expérimentalement par les bruits d'entrées-sorties et toute sorte de variables parasites.

Il analyse, la simulation et la conception basée sur des modèles d'ordre élevés peuvent éventuellement conduire à des contrôles compliqués, présentant une logique très complexe, d'où un temps de calcul extrêmement grand.

La réduction du modèle est alors désirable.

L'approximation de modèles résolus dans cette étude, consiste à trouver un modèle simple qui représente le mieux possible le modèle complexe à très grand nombre de paramètres.

Le problème est donc de trouver l'approximation d'ordre le plus bas pour une performances donnée, en utilisant systématiquement les valeurs propres des Grammians de commandabilité et d'observabilité des équations de Lyapunov et après les valeurs singulières de la matrices de Hankel indépendamment.

Ce travail à l'origine exécuté et expérimenté par M. Bettayeb (University of Southern California) ouvre un nouvel espace dans les théories de contrôle de systèmes, et qui aura un impact majeur sur des problèmes quelconques dans l'identification de systèmes et dans les considérations énergétiques.

1.2 NECESSITE DE REDUIRE

La commandabilité et l'observabilité d'un modèle physique régi par un système mathématique très large nous obligent à tenir compte de sa réduction, afin de pouvoir le simuler et exploiter ses données et interprétations, car il est pratiquement impossible de simuler un système à très grandes dimensions. Ceci du fait de la capacité limitée des ordinateurs et autres unités de calculs.

Le choix d'une meilleure schématisation nous met dans l'obligation de considérer quelques critères de performances nécessaires à la commandabilité, observabilité et stabilité du système.

Parmi ces précautions à prendre nous considérons :

- la commandabilité du système
- la performance numérique.
- la simplification de la programmation
- la durée de calcul (par l'ordinateur)

et d'autres considérations qui sont dépendantes les unes des autres.

1.3 APPLICATIONS et MOTIVATIONS

Le modèle mathématique des systèmes physiques donne naissance aux modèles de grandes dimensions.

Quelques exemples d'applications sont dans les systèmes de circuits des télécommunications, des lignes de transmission, de la propagation des ondes en acoustique, des systèmes en économie appliquée, des réacteurs nucléaires, et bien d'autres domaines.

Basé sur un dimensionnement fini des représentations des espaces d'états, un certain nombre de modèles du contrôle moderne, par exemple les filtres de Kalman, les "feedback",

les modèles estimateurs d'états, et les filtres récursifs.

Naturellement, l'ordre du système dictera le calcul complexe. Dans un système d'ordre plus bas serait préférable.

Ceci motive l'étude du problème d'approximation qui provient dans au moins les situations suivantes:

Réduction de modèles:

Pratiquement, le modèle du système de control d'un ordre élevé est susceptible de devenir de plus grand ordre, ou même devenir plus compliqué à construire.

En somme un modèle d'ordre plus bas, conduira à un processus plus simple.

Donc la réduction technique d'un modèle qui approxime un système plus large par un ordre plus bas, est très important aussi bien que les théories de control moderne.

Identifications:

Cette situation se présente quand il est seulement fourni, une mise en information de l'entrée-sortie c'est à dire la séquence de la réponse à l'impulsion pour laquelle nous sommes en mesure d'identifier les paramètres qui représentent le sous-système ou alors simplement de choisir un modèle qui représentera le mieux possible le système.

Dans ce cas l'ordre devrait être déterminé pour un avantage de l'information signal/bruit.

Conceptions de filtres digitaux:

Cette application est basée sur la conception de filtres digitaux, qui a été un sujet assez célèbre dans le processus du signal où il est donné une spécification du domaine de fréquence.

En résumé, les problèmes de réduction de modèles, d'identifications, et de conceptions de filtres digitaux, et bien d'autres encore peuvent être traités mathématiquement.

1.4. APERÇU DES METHODES DE REDUCTION :

La contribution universelle à la réduction des systèmes linéaires dynamiques complexes par des modèles d'ordres inférieures est assez riche compte tenu des innombrables travaux faits depuis plusieurs années et continue à susciter l'intérêt des spécialistes du contrôle de systèmes.

Un grand nombre de conceptions de problèmes pratiques ont vu le jour sous forme de modèles simplifiés correspondants aux systèmes exacts donc complexes.

Une des premières approches des méthodes temporelles, utilisant les techniques d'espaces d'états était le modèle d'analyse de E.M. Davison [5] puis M.R. CHIDAMBARA dans laquelle, ils retiennent essentiellement le système des valeurs propres dominantes et leurs vecteurs propres correspondants.

Dans d'autres approches, Chen et Chieh [6] utilisèrent les expansions de fréquences, et Hutton et Friedland [7] utilisèrent l'approche de Routh [8] pour l'approximation des hautes fréquences (récemment modifiée par Langholz et Feinmesser [9]).

Il a été démontré que toutes ces méthodes sont des reformulations d'espaces d'états.

La théorie de réalisation minimale de Kalman [9] repose sur des objectifs géométriques (Commandabilité, sous-espaces non observables) qui sont sujets aux structures instables.

Spécifiquement, des petites perturbations arbitraires dans un modèle peuvent causer un changement dans les dimensions des sous-espaces correspondants.

Cette situation est manifestée dans les difficultés de calcul (programmation) qui provient dans les essais d'applications des algorithmes pour la détermination d'une réalisation minimale.

L'instabilité structure associée aux théories géométriques n'est pas unique pour le contrôle de systèmes, elle provient aussi bien dans les théories des équations linéaires.

Le problème posé par les textes des algorithmes pour les théories de réalisations minimales, est, qu'elles sont basées sur le contenu littéral des théories mais ne peuvent lutter contre les discontinuités structurelles.

Nous nous intéresserons dans une large mesure aux :

- méthodes temporelles
- et - méthodes fréquentielles.

1.4.1 Méthodes temporelles: .5.

À titre d'exemple, nous présenterons une méthode de réduction des systèmes linéaires dynamiques élaborée par M.R. CHIDAMBARA qui s'inspire directement de l'approche de E.J. DARISON, qui ne conserve que les valeurs propres dominantes.

Il est donc demandé de trouver une approximation d'un modèle mathématique représenté par un système multivariable linéaire continu caractérisé par les équations d'espaces d'états familiers:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Hx \end{aligned} \quad (1)$$

où A est une matrice $n \times n$; l'approximation du modèle mathématique est donnée par:

$$\begin{aligned} \dot{y} &= A^*y + B^*u \\ C &= Hx \end{aligned} \quad (2)$$

où existe une relation entre x et y avec A^* est de $\dim(l \times l)$ avec $l < n$.

On suppose que l'équation caractéristique associée à la matrice A est donnée par:

$$(\lambda - \lambda_1)^{k_1} (\lambda - \lambda_2)^{k_2} \dots (\lambda - \lambda_m)^{k_m} = 0 \quad (3)$$

où k_i est l'ordre de multiplicité des valeurs propres λ_i avec la relation:

$$\sum_{i=1}^m k_i = n \quad (4)$$

Soit maintenant la nullité de la matrice:

$$E_i = [A - \lambda_i I] \quad (5)$$

$$\text{et } v_i = (n - r_i) \quad (6)$$

où r_i est le rang de la matrice E_i , $v_i \leq k_i$

si $v_i = k_i$ ($i = 1, \dots, m$) alors pour chaque valeur propre λ_i , on peut trouver k_i vecteurs propres linéairement indépendants et donc la matrice A peut être transformée en une matrice purement diagonale.

Dans le cas où $v_i \neq k_i$, la valeur propre λ_i donne

des vecteurs propres dégénérés. Dans ce cas il est connu que la matrice non singulière P peut être trouvée telle que :

$$P^{-1}AP = J = \begin{bmatrix} J_1 & 0 & \dots & 0 \\ 0 & J_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & J_m \end{bmatrix} \quad (7)$$

où les $(k_i \times k_i)$ de J , possède à chacune de ses principales éléments de la diagonale, égaux à λ_i , et la diagonale oblique supérieure a $(k_i - v_i)$ éléments égaux à l'unité avec $(v_i - 1)$ éléments égaux à zéro.

On définit une matrice colonne X telle que :

$$X = P \cdot Z \quad (8)$$

A partir de (8), (7) et (1) on obtient :

$$\dot{Z} = JZ + P^{-1} \cdot B U \quad (9)$$

La partition de Z en deux matrices colonnes Z_1 et Z_2 respectivement d'ordre $l \times 1$ et $(n-l) \times 1$.

Il est toujours possible de mettre P de façon que les valeurs propres de A soient dans J_1, J_2, \dots, J_j et les autres dans J_{j+1}, \dots, J_m .

$$\dot{Z}_1 = P_j \cdot Z_1 + \text{les premières } l \text{ lignes de } P^{-1} \cdot B U \quad (11)$$

$$\text{si on définit } Z_1 = Y.$$

$$P_j = A^*$$

$$\text{les premières } l \text{ lignes de } P^{-1} \cdot B = B^* \quad (12)$$

Il reste uniquement de mettre X en relation de Y .
A partir de (8) on peut écrire :

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (13)$$

$$x_1 = P_{11} z_1 + P_{12} z_2 \quad (14)$$

$$x_2 = P_{21} z_1 + P_{22} z_2 \quad (15)$$

7.

Maintenant on doit faire attention dans la négligence de certains termes. Seulement ces termes qui sont de m ordre importantes en les comparant aux autres pensent être négligés

Il serait faux de dire que:

$$x_1 = P_{11} z_1$$

$$\text{et } x_2 = P_{21} z_1. \quad (16)$$

même si cependant $z_1(t)$ contient seulement des termes avec des valeurs propres dont l'effet est négligeable.

Il est possible d'arranger les lignes de P de façon que P_{11} soit non singulière. Alors les colonnes de P sont linéairement indépendantes, les premières l colonnes de P contiendraient au moins une matrice non singulière $l \times l$.

En utilisant (16) et (11) on obtient:

$$\dot{x}_1 = \dot{y}_1 = P_{11} \int P_{11}^{-1} \cdot y + P_{11} \left\{ \text{les premières } l \text{ lignes de } P^{-1} \cdot B \right\} u \quad (17)$$

qui a la même forme que (2) alors:

$$A^* = P_{11} \int P_{11}^{-1}$$

$$B^* = P_{11} \left\{ \text{les premières } l \text{ lignes de } P^{-1} \cdot B \right\} \quad (18)$$

En utilisant l'équation (16) dans:

$$\dot{x} = Ax + Bu \quad (1-a)$$

on obtient un résultat différent. En arrangeant les lignes et colonnes de A , on écrit:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \quad (19)$$

$$\dot{x}_1 = a_{11} x_1 + a_{12} x_2 + B_1 u.$$

en substituant x_2 à partir de l'équation (16)

$$\text{on a: } \dot{x}_1 = [a_{11} + a_{12} P_{21} P_{11}^{-1}] x_1 + B_1 u \quad (20)$$

ceci est de nouveau de forme (2) alors:

$$A^* = [a_{11} + a_{12} P_{21} P_{11}^{-1}]$$

$$B^* = B_1 = \left\{ \text{les lignes de } B \text{ correspondants aux variables d'état } x_1 \right\}$$

1.4.2 Méthodes fréquentielles.

À titre d'exemple, nous présenterons ici, une méthode de réduction fréquentielle élaborée par Routh [8] qui consiste donc préserver les hautes fréquences au détriment des basses fréquences.

On considère un système linéaire ayant une fonction de transfert :

$$H(s) = \frac{b_1 s^{n-1} + \dots + b_n}{a_0 s^n + a_1 s^{n-1} + \dots + a_n} \quad (1)$$

Un système linéaire avec l sorties et m entrées peut être représenté par une matrice de fonction de transfert de la forme (1) avec les coefficients au numérateur b_i formant des matrices de dimension $l \times m$.

Expansion Alpha-Bêta

Une fonction de transfert de la forme (1) qui est asymptotiquement stable peut toujours être étendue à la forme canonique suivante :

$$\begin{aligned} H(s) &= \beta_1 F_1(s) + \beta_2 F_2(s) F_2(s) + \dots + \beta_n F_1(s) F_2(s) \dots F_n(s) \\ &= \sum_{l=1}^n \beta_l \prod_{j=1}^l F_j(s) \quad (2) \end{aligned}$$

avec les β_i ($i=1, 2, \dots, n$) constants et les F_i ($i=2, \dots, n$) sont définies par les expansions en fractions continues.

$$F_i(s) = \frac{1}{\alpha_i s + \frac{1}{\alpha_{i+1} s + \frac{1}{\ddots \ddots \ddots \alpha_{n-1} s + \frac{1}{\alpha_n s}}}} \quad (3)$$

Pour $F_1(s)$, la relation (3) est légèrement modifiée, le premier terme dans la forme de la fraction continue est $(1 + \alpha_1 s)$ à la place de $\alpha_1 s$.

La forme canonique (2) est référée à l'expansion alpha-Bêta de l'approximation de Routh.

Une représentation bloc-diagramme de cette expansion canonique est donnée dans la (fig 1)

Les n coefficients α_i apparaissant dans l'expansion alpha-bêta peuvent être calculés en utilisant l'algorithme de construction de la table de Routh (8) utilisé dans l'analyse de stabilité comme montré dans la table (I).

Les 2 (deux) premières lignes de la table sont formées à partir des entrées supposées :

$$a_j^0 = a_{j-1}^1 = 0 \quad \text{pour } j > n.$$

Les entrées restantes sont formées par :

$$a_0^{i+1} = a_2^{i-1} - \alpha_i a_2^i$$

$$a_2^{i+1} = a_4^{i-1} - \alpha_i a_4^i$$

$$\vdots$$

$$a_{n-i-2}^{i+1} = a_{n-i}^{i-1} - \alpha_i a_{n-i}^i \quad \text{avec } i = 1, \dots, n-1 \quad (4)$$

Pour $n-i$, impair, la dernière équation (4) est remplacée par (5)

$$a_{n-i-1}^{i+1} = a_{n-i+1}^{i-1}$$

Les α_i entrées marginales sont données par :

$$\alpha_i = \frac{a_0^{i-1}}{a_0^i} \quad \text{avec } i = 1, 2, \dots, n$$

L'utilisation de la table de Routh pour le développement de la forme canonique utilisée dans les approximations à développer résultantes sont appelées "Approximations de Routh"

Les coefficients β_i apparaissant dans la forme canonique peuvent aussi être obtenus en utilisant l'algorithme tabulaire comme montré dans la table (II)

Les 2 premiers coefficients de la table β sont obtenus à partir des coefficients du numérateur de $H(s)$. Les entrées restantes sont calculées à partir des entrées de la table de Routh (table I), et les entrées premières lignes de la table bêta, utilisant de la récursion suivante :

$$\beta_i = \frac{b_i^i}{a_0^i} \quad ; \quad i = 2, 3, \dots, n \quad (5)$$

$$\text{et } b_{j-2}^{i+1} = b_j^i - \beta_i a_j^i \quad \text{avec } j = \begin{cases} 2, 4, \dots, n & \text{pour } n-i \text{ pair} \\ 2, 4, \dots, n-i-1 & \text{pour } n-i \text{ impair} \end{cases}$$

et $i = 1, 2, \dots, n-2$

L'expansion canonique (2) définit une paire d'algorithmes récursifs finis :

1^{er} ALGORITHME : \int partir de (4) et (6) pour le calcul des n coefficients α_i à partir du dénominateur de la fonction de transfert.

2^e ALGORITHME : \int partir de (7) et (8) pour le calcul des n coefficients β_i à partir du dénominateur de la fonction de transfert.

Les paramètres et les polynômes du dénominateur de la fonction de transfert, peuvent être calculés à partir de l'expansion alpha-beta et sera résolue en introduisant la convergence de Routh.

Convergence de Routh :

La k ^{ème} convergence de Routh $R_k(s)$ pour la fonction de transfert $H(s)$ est obtenue en tronquant l'expansion $(\alpha-\beta)$ et en arrangeant les résultats comme une fonction rationnelle de s .

La troncation élimine les termes $\alpha_{k+1}, \dots, \alpha_n, \beta_{k+1}, \dots, \beta_n$ et donc l'expansion $(\alpha-\beta)$ dépendra seulement des premiers α, β d'ordre k .

On considère les fonctions $G_{i,k}$ de k définies par les expansions fractionnaires continues suivantes :

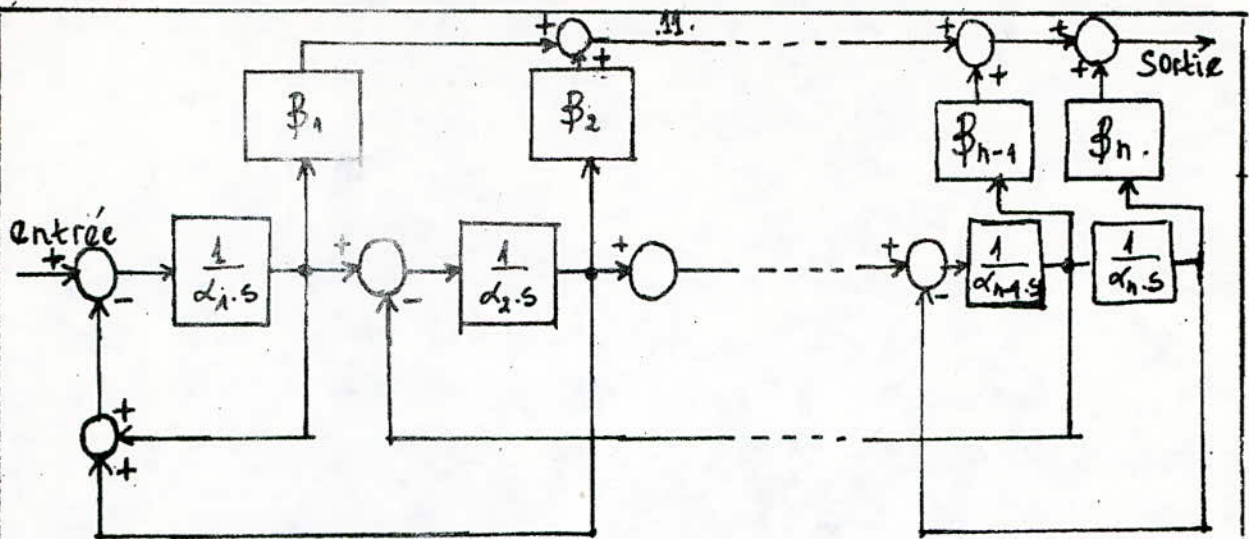
$$G_{i,k} = \frac{1}{\alpha_i s + \frac{1}{\alpha_{i+1} s + \frac{1}{\alpha_{i+2} s + \dots + \frac{1}{\alpha_{i+k} s + \frac{1}{\alpha_k s}}}}}$$

pour $i=1$, la relation ci-dessus est modifiée légèrement : le premier terme dans l'expansion continue est : $1 + \alpha_1 s$ au lieu de $\alpha_1 s$.

Une comparaison de (3) montre que $G_{i,k}$ est le résultat de la troncation de $(n-k)$ derniers termes dans l'expansion F_i .

En utilisant ces nouvelles fonctions $G_{i,k}$, la k ^{ème} convergence de Routh est égale à :

$$R(s) = \beta_1 G_{2,k}(s) + \beta_2 G_{1,k}(s) + \dots + \beta_k G_{1,k}(s) G_{2,k}(s) \dots G_{k,k}(s) \\ = \sum_{l=1}^k \beta_l \prod_{i=1}^l G_{i,k}(s) \quad (9)$$



Bloc-diagramme de la forme canonique (α - β)

TABLE (I) - TABLE ALPHA de ROUTH -

	$a_0^0 = a_0$ $a_1^0 = a_1$	$a_0^2 = a_2$ $a_2^2 = a_3$	$a_0^4 = a_4$ $a_2^4 = a_5$	$a_0^6 = a_6 \dots$ \dots
$\alpha_1 = a_0^0/a_1^0$	$a_0^1 = a_2^0 - \alpha_1 a_2^1$	$a_2^1 = a_4^0 - \alpha_1 a_4^1$	$a_4^1 = a_6^0 - \alpha_1 a_6^1$	\dots
$\alpha_2 = a_0^1/a_1^1$	$a_0^2 = a_2^1 - \alpha_2 a_2^2$	$a_2^2 = a_4^1 - \alpha_2 a_4^2$	\dots	\dots
$\alpha_3 = a_0^2/a_1^2$	$a_0^3 = a_2^2 - \alpha_3 a_2^3$	$a_2^3 = a_4^2 - \alpha_3 a_4^3$	\dots	\dots
$\alpha_4 = a_0^3/a_1^3$	$a_0^4 = a_2^3 - \alpha_4 a_2^4$	\dots	\dots	\dots
$\alpha_5 = a_0^4/a_1^4$	$a_0^5 = a_2^4 - \alpha_5 a_2^5$	\dots	\dots	\dots
$\alpha_6 = a_0^5/a_1^5$	\dots	\dots	\dots	\dots

TABLE (II) - TABLE BETA DE ROUTH

	$b_0^1 = b_1$ $b_2^1 = b_2$ $b_0^2 = b_2$	$b_2^3 = b_3$ $b_2^2 = b_4$	$b_4^5 = b_5 \dots$ $b_4^2 = b_6 \dots$
$\beta_1 = b_0^1/a_1^1$	$b_0^3 = b_2^1 - \beta_1 a_2^1$	$b_2^3 = b_4^1 - \beta_1 a_4^1$	\dots
$\beta_2 = b_0^2/a_1^2$	$b_0^4 = b_2^2 - \beta_2 a_2^2$	$b_2^4 = b_4^2 - \beta_2 a_4^2$	\dots
$\beta_3 = b_0^3/a_1^3$	$b_0^5 = b_2^3 - \beta_3 a_2^3$	\dots	\dots
$\beta_4 = b_0^4/a_1^4$	$b_0^6 = b_2^4 - \beta_4 a_2^4$	\dots	\dots
$\beta_5 = b_0^5/a_1^5$	\dots	\dots	\dots
$\beta_6 = b_0^6/a_1^6$	\dots	\dots	\dots

Soient $A_k(s)$ et $B_k(s)$ dénotent respectivement le dénominateur et numérateur de la k -ième convergence de Routh.

c.a.d:

$$A_1(s) = \alpha_1 s + 1.$$

$$B_1(s) = \beta_1$$

$$A_2(s) = \alpha_1 \alpha_2 s^2 + \alpha_2 s + 1.$$

$$B_2(s) = \alpha_2 \beta_1 s + \beta_2.$$

$$A_3(s) = \alpha_1 \alpha_2 \alpha_3 s^3 + \alpha_2 \alpha_3 s^2 + (\alpha_1 + \alpha_3) s + 1.$$

$$B_3(s) = \alpha_2 \alpha_3 \beta_1 s^2 + \alpha_3 \beta_2 s + (\beta_1 + \beta_2)$$

d'une manière générale, comme montré dans (9)

$$A_k(s) = \alpha_k \cdot s \cdot A_{k-1}(s) + A_{k-2}(s) \quad (10)$$

$$\text{et } B_k(s) = \alpha_k \cdot s \cdot \beta_{k-1}(s) + \beta_{k-2}(s) + \beta_k \quad (11)$$

avec $k = 1, 2, \dots$

$$\text{et } A_{-1}(s) = 0 ; \beta_{-1}(s) = 0$$

$$A_0(s) = 1 ; \beta_0(s) = 0$$

La convergence $R_k(s)$ de Routh est une approximation de $H(s)$ qui tend à préserver les hautes fréquences au détriment des basses fréquences.

donc pour des applications de control, il est préférable d'obtenir une approximation de basses fréquences.

chapitre 2

REDUCTION DE SYSTEMES LINEAIRES

21 : POSITION DU PROBLEME :

Réduction analytique de l'espace d'état représenté par le modèle :

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}$$

Plusieurs méthodes de réduction de systèmes linéaires dynamiques complexes ont été proposées durant les quinze dernières années. Elles sont le plus souvent basées sur des analyses de modèles ou des concepts de domaines de fréquences.

Récemment, une méthode s'appuyant sur des calculs de Commandabilité et d'observabilité, a été suggérée par Moore [1] et développée par Bellayeb [2].

Cette procédure de réduction des espaces d'états équilibrés sera exposée ici. L'ordre du modèle réduit est examiné du point de vue de la stabilité, contrôlabilité (commandabilité), et observabilité.

Deux cas de systèmes seront considérés :

- système continu.
- système discret.

22 : REDUCTION de (A,B,C) en SYSTEME CONTINU :

Un système linéaire à temps invariant (continu) peut être représenté par le modèle suivant :

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\quad \text{avec } x(t_0) = x_0$$

Où la variable d'état au temps t , $x(t)$ est un vecteur d'ordre n ; l'entrée au temps t , $u(t)$ est un vecteur d'ordre p , et la sortie au temps t , $y(t)$ est un vecteur d'ordre m .

A, B, C sont des matrices de dimensions compatibles aux vecteurs auxquels ils sont affectés.

La paire (A, B) assure la commandabilité, tandis que (C, A) assure l'observabilité.

Commandabilité et observabilité peuvent être caractérisés par les termes des matrices suivantes :

$$W_c(t) = \int_0^t e^{\tau A} B B^T e^{\tau A^T} d\tau \quad (1.1)$$

$$\text{et } W_o(t) = \int_0^t e^{\tau A^T} C^T C e^{\tau A} d\tau \quad (1.2)$$

(Grammians de commandabilité et d'observabilité),
et qui sont (pour $t \rightarrow \infty$), solutions des équations algébriques de Lyapunov:

$$A W_c + W_c A^T = -B B^T \quad (1.3)$$

$$A^T W_o + W_o A = -C^T C \quad (1.4)$$

apparaissent fréquemment dans cette partie de l'étude.

La matrice A est assurée stable sur $t \in [0, \infty[$.
Dans tout ce qui suit, nous considérons le système commandable, observable, et asymptotiquement stable et équilibré (balanced) sur l'intervalle $[0, \infty[$.

Soit le système décrit par les équations d'états de la forme (1.1) et (1.2), les grammians, qui sont tous deux non singuliers pour $t > 0$, et égaux à une matrice diagonale Σ , sont alors donnés comme une solution unique définie positive par les deux équations de Lyapunov.

$$A \Sigma + \Sigma A^T = -B B^T \quad (1.5)$$

$$\text{et } \Sigma A + A^T \Sigma = -C^T C \quad (1.6)$$

Transformation de coordonnées :

Si nous considérons une transformation des objectifs ci-dessus c.a.d des états $x(t)$, sous un changement de coordonnées :

$$x(t) = T \hat{x}(t) \quad \text{où } T \in \mathbb{R}^{n \times n} \text{ et non singulière}$$

$$\text{alors } \hat{A} = T^{-1} A T \quad ; \quad \hat{B} = T^{-1} B \quad ; \quad \hat{C} = C T \quad (1.7)$$

$$\text{d'où } \hat{W}_c = T^{-1} W_c T^{-T} \quad \text{et } \hat{W}_o = T^T W_o T$$

$$\text{ce qui entraîne : } \hat{W}_c \hat{W}_o = T^{-1} W_c W_o T$$

ceci possède des propriétés intrinsèques de réductions de systèmes.

On cherche la transformation T qui diagonalise simultanément les matrices W_c et W_o .

W_c et W_o étant symétriques définies positives, il existe donc une matrice V_c orthogonale telle que :

$$V_c^T W_c V_c = \Sigma_c^2 \quad (1.8)$$

où Σ_c est une matrice diagonale de nombres positifs.

De façon similaire il existe une matrice orthogonale V_0 et une matrice diagonale positive, Σ_0 telle que :

$$V_0^T W_0 V_0 = \Sigma_0^2 \quad (1.9)$$

Il existe par ailleurs une matrice orthogonale U et une matrice diagonale positive Σ telle que :

$$U^T [(V_c \Sigma_c)^T W_0 (V_c \Sigma_c)] U = \Sigma^2 \quad (1.10)$$

Maintenant nous considérons la transformation d'équilibre qui nous intéresse :

$$T = V_c \Sigma_c U \Sigma^{-1/2} \quad (1.11)$$

On peut remplacer $V_c \Sigma_c$ par L_c où L_c est un facteur de Cholesky $[]$ de W_c .

$$\text{d'où } W_c = L_c L_c^T \quad (1.12)$$

Spécifiquement T peut être construit à partir de :

(1) la factorisation de Cholesky (1.12) $W_c = L_c L_c^T$

(2) les valeurs propres et vecteurs propres :

$$U^T (L_c^T W_0 L_c) U = \Sigma^2$$

(3) $T = L_c U \Sigma^{-1/2}$.

avec T maintenant et en utilisant la forme équilibrée, on peut trouver : $(\hat{A}, \hat{B}, \hat{C})$.

On montre maintenant grâce à la transformée T que :

$\hat{W}_c = \Sigma$ et $\hat{W}_0 = \Sigma$ donc les grammians sont égaux et diagonaux (dans la forme équilibrée).

Si nous considérons $(A, B, C)_n$ équilibrée et la matrice diagonale Σ telle que :

$$\Sigma = \begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_n \end{bmatrix} \quad \text{et la partition } \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}$$

alors les partitions correspondantes à (A, B, C) sont :

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}; \quad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}; \quad C = (C_1 \quad C_2)$$

Cette partition a plusieurs propriétés remarquables.

si $\gamma_k > \gamma_{k+1}$, alors il est naturel de penser que $(A_{11}, B_1, C_1)_k$ est la partie la plus commandable et observable. Moore []

Ceci nous motive donc à prendre $(A_{11}, B_1, C_1)_k$ comme une approximation raisonnable d'ordre k de $(A, B, C)_n$.

La stabilité de $(A_{11}, B_1, C_1)_k$ peut être aussi assurée. Cette technique d'approximation de mesure très favorablement avec les autres types d'approximations.

On signale qu'une autre manière d'aboutir à l'approximation équilibrée est obtenue par le biais de la décomposition en composantes principales de la matrice de Hankel H .

23 : REDUCTION DE (A, B, C) EN SYSTEME DISCRET :

La méthode de réduction du système en temps discret est analogue à celle utilisée en système continu. Il y'a cependant quelques différences fondamentales entre les systèmes continus et systèmes discrets équilibrés.

Le système discret représenté par :

$$X_{k+1} = AX_k + BU_k \quad (2.1)$$

$$Y_k = CX_k \quad (2.2)$$

qui est assuré asymptotiquement stable, commandable et observable, définit les grammians W_c, W_o respectivement de commandabilité et d'observabilité qui sont alors :

$$W_c = \sum_{k=0}^{\infty} A^k B B^T (A^T)^k \quad (2.3)$$

$$W_o = \sum_{k=0}^{\infty} (A^T)^k C^T C A^k \quad (2.4)$$

Comme dans le cas continu, il est démontré, qu'il est possible de trouver une transformation d'équivalence qui transforme

17.
 Les deux (2) grammians égaux à une matrice diagonale définie positive (Σ). La correspondance des représentations d'états est dite équilibrée sur l'intervalle $[0, \infty[$, dans lequel le système est asymptotiquement stable, commandable et observable.

Les grammians, qui sont tous deux égaux à Σ , sont donnés par l'unique solution définie positive des équations de Lyapunov.

$$A\Sigma A^T - \Sigma = -BB^T \quad (2.5)$$

$$\text{et } A^T \Sigma A - \Sigma = -C^T C \quad (2.6)$$

D'un point de vue analytique, il est possible de passer du système continu en système discret.

Étant donné que la résolution des équations de Lyapunov s'est effectuée dans notre cas en système continu (voir chapitre Programmation et simulation), on passe donc en système continu en s'appuyant sur la forme discrète des équations de Lyapunov donnée par (2.1) et (2.2)

La programmation utilisée en FORTRAN IV pour la résolution des équations de Lyapunov, se fait dans notre cas par la méthode de BINGULAC [3] en système continu:

Le passage du système discret au système continu se fait par la correspondance suivante: (voir Bellayeb [2])

$$A_c = (A_D + I)^{-1} \cdot (A_D - I)$$

$$B_c = \sqrt{2} (A_D + I)^{-1} \cdot B_D$$

$$C_c = \sqrt{2} C_D (A_D + I)^{-1}$$

N.B: Il n'est pas nécessaire de donner la démonstration de cette correspondance entre les matrices (A, B, C) continue et les matrices (A, B, C) discret.

2.4: REDUCTION DES PARAMETRES de MARKOV (H_i) de la matrice de Hankel.

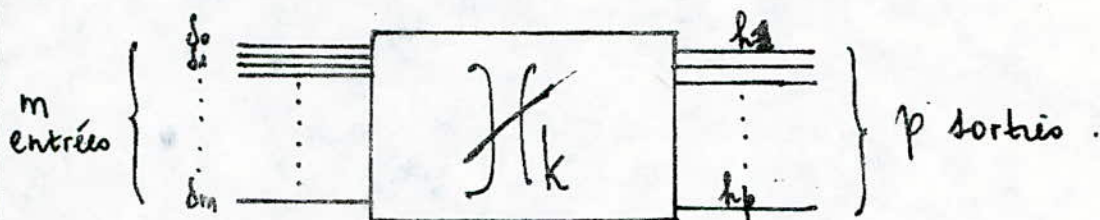
Nous serons concernés ici par les représentations $(A, B, C)_n$ d'un système linéaire dynamique discret, décrits par les équations:

$$X_{k+1} = AX_k + BU_k$$

$$Y_k = CX_k$$

Où X_k, U_k, Y_k sont respectivement, la variable d'état, les vecteurs d'entrées et sorties à l'instant k du système.

Où $X_k \in \mathbb{R}^n, U_k \in \mathbb{R}^m, Y_k \in \mathbb{R}^p$
correspondants à $(A, B, C)_n$ et la réponse à l'impulsion donnée par:



$$H_k = CA^{k-1}B \quad k = 1, 2, \dots$$

et la fonction de transfert du système:

$$F(z) = C(zI - A)^{-1} \cdot B = \sum_{k=0}^{\infty} H_k \cdot z^{-k}$$

La séquence de Hankel formée par la séquence H_i joue un rôle majeur ici et est définie par la matrice finie:

$$H_j = \begin{bmatrix} H_1 & H_2 & \dots & H_j \\ H_2 & H_3 & \dots & H_{j+1} \\ \vdots & \vdots & \ddots & \vdots \\ H_j & H_{j+1} & \dots & H_{2j-1} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{j-1} \end{bmatrix} \begin{bmatrix} B & AB & \dots & A^{j-1}B \end{bmatrix}$$

qui peut se mettre sous forme :

$$H_j = O_j G_j$$

Où O_j et G_j sont les matrices de rang j , d'observabilité et de commandabilité pour $(A, B, C)_n$.

Après avoir formé les paramètres de Markov H_i ($i = 1, 2, \dots, 2N-1$), on considère la matrice de Hankel de dimension $(N \times N_m)$, avec H_i blocs-symétriques de dimension $(p \times m)$.

$$H_N = \begin{bmatrix} H_1 & H_2 & H_3 & \dots & H_N \\ H_2 & H_3 & \dots & \dots & H_{2N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ H_N & \dots & \dots & \dots & H_{2N-1} \end{bmatrix} = O_N G_N.$$

On décompose H en composantes principales :

$$H = U \Sigma V^T = (U \Sigma^{1/2}) (\Sigma^{1/2} V^T)$$

$$\text{avec } \Sigma = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & \sigma_N \end{bmatrix} \text{ avec } \sigma_1 \geq \sigma_2 \dots \geq \sigma_N$$

$$\text{et } U = [U_1, U_2, \dots, U_N], \quad V^T = \begin{bmatrix} V_1^T \\ V_2^T \\ \vdots \\ V_N^T \end{bmatrix}$$

Initialement l'ordre exact du système est égal au rang de la matrice de Hankel H . Si (A, B, C) représente un espace d'état d'ordre N , avec $H_k = CA^{k-1}B$, alors la matrice de Hankel sera d'ordre N et on peut la factoriser comme :

$$H_N = O_N G_N.$$

où Θ_N représente la matrice d'observabilité étendue à N .
 et G_N représente la matrice de contrôlabilité étendue à N .

c.a.d:

$$\Theta_N = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{N-1} \end{bmatrix} \quad \text{et} \quad G_N = \begin{bmatrix} B & AB & A^2B & \dots & A^{N-1}B \end{bmatrix}$$

Si le système possède m entrées et p sorties, c'est à dire B est $(N \times m)$, et C est $(p \times N)$ alors nous avons:

$$\Theta A = \Theta \uparrow$$

$$\text{et } A G = \overleftarrow{G}$$

avec:

$$\Theta \uparrow = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{\Theta} \\ 0 \end{bmatrix}$$

$$\text{et } \overleftarrow{G} = \begin{bmatrix} AB & A^2B & \dots & A^{N-1}B & 0 \\ \tilde{G} & & & & 0 \end{bmatrix}$$

Nous affirmons (et il est facile de le vérifier) que la paire $[\Theta_N, G_N]$ représentent les matrices d'observabilité et de commandabilité d'ordre N du système, si et seulement si, il existe une solution A , satisfaisant simultanément les équations:

$$\Theta A = \Theta \uparrow$$

$$\text{et } A G = \overleftarrow{G}$$

Une factorisation particulière de Θ , et G à partir de la matrice de Hankel c'est à dire :

$$H = \Theta \cdot G = [U \Sigma^{1/2}] [\Sigma^{1/2} V^T]$$

$$\text{où } \Theta A = \Theta^\dagger \quad \text{et } A G = \overleftarrow{G}$$

admet une solution qui est :

$$A = \Theta^\dagger \Theta^\dagger = \overleftarrow{G} G^+$$

Le signe (+) signifie "pseudo-inverse"

$$\Theta^\dagger = (\Theta^T \Theta)^{-1} \Theta^T$$

$$\text{et } G^+ = G^T (G G^T)^{-1}$$

donc une réalisation $[A, B, C]_N$ est donnée par l'assignation :

$$A = \Theta^\dagger \Theta^\dagger$$

$B =$ les m premières blocs-colonnes de G

$C =$ les p premiers blocs-lignes de Θ .

Si $(A, B, C)_N$ est une représentation d'un système stable minimal, alors la matrice de Hankel H peut être partitionnée

comme :

$$H = [U_1 \mid U_2] \cdot \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \cdot \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

$$\text{avec : } \Sigma_1 = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & 0 \\ & & & \ddots \\ 0 & & & & \sigma_p \end{bmatrix} \quad \text{et} \quad \Sigma_2 = \begin{bmatrix} \sigma_{p+1} & & \\ & \ddots & \\ & & 0 \\ & & & \ddots \\ & & & & \sigma_N \end{bmatrix}$$

Alors le produit de matrice $\Delta_1 = U_1 \Sigma_1 V_1^T$ minimise $\|H - \Delta\|_s$ de rang r et $\|H - \Delta_1\|_s = \sigma_{r+1}$; où ($\|A\|_s^2 = \max$ des valeurs propres de $A^T A$).

Malheureusement Δ_1 n'est pas une matrice de Hankel en général.

Cependant, en utilisant un algorithme similaire à celui présenté ci-dessus, l'approximation suivante est obtenue :

$$A_r = \Theta_r^+ \Theta_r^\uparrow = \Sigma_r^{-1/2} U_r^T (U_r \Sigma_r^{1/2})^T$$

$$B_r = m \text{ premières bloc-colonnes de } G_r = \Sigma_r^{1/2} V_r^T$$

$$C_r = p \text{ premiers bloc-lignes de } \Theta_r = U_r \Sigma_r^{1/2}$$

avec :

$$\Theta_r = [U_1, \dots, U_r] \cdot \begin{bmatrix} \sqrt{\sigma_1} & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & \sqrt{\sigma_r} \end{bmatrix}$$

$$G_r = \begin{bmatrix} \sqrt{\sigma_1} & & & \\ & \ddots & & \\ & & \ddots & \\ 0 & & & \sqrt{\sigma_r} \end{bmatrix} \cdot \begin{bmatrix} V_1^T \\ \vdots \\ V_r^T \end{bmatrix}$$

Cette approximation proposée par Kung [] est une donnée relative de "Leiger McEwen" [] et montrée équivalente à une approximation équilibrée par Bellayeb []
c'est à dire: $A_r = A_{11}$, $B_r = B_1$, $C_r = C_1$.

2.5 EQUIVALENCE DES DEUX MODELES REDUITS ^{.23.}

Deux modèles réduits ont été indépendamment développés dans 2.3 et 2.4.

Dans le cas de systèmes équilibrés (balanced systems), la démonstration de l'équivalence a été faite par M. Bettayeb en s'inspirant des résultats de l'approximation de systèmes réduits établis par Kung [10] dans le chap. 2.4 et par Moore [11] dans le chap. 2.3

Si $(A, B, C)_n$ représente un système équilibré partitionné comme :

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} ; \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} ; \quad C = [C_1 ; C_2]$$

où A_{11} est $r \times r$

B_1 est $r \times m$

C_1 est $p \times r$

alors : $A_{11} = \hat{A}_1$

$B_1 = \hat{B}_1$

$C_1 = \hat{C}_1$

où $(\hat{A}_1, \hat{B}_1, \hat{C}_1)_r$ est l'approximation des systèmes établis à l'ordre r .

Puisque $(A, B, C)_n$ est équilibré :

$$\Theta = [\Theta_1, \Theta_2] = [U_1 \Sigma_1^{1/2} ; U_2 \Sigma_2^{1/2}]$$

mais $\Theta^* = \Theta A$ alors :

$$\Theta_1^* A_{11} + \Theta_2^* A_{21} = \Theta_1^* \quad (1)$$

en multipliant les 2 côtés de cette expression (1) par Θ_1^+ et en utilisant le fait que $\Theta_1^+ \Theta_2 = \Theta$; par orthogonalité :

$$A_{11} = \Theta_1^+ \Theta_1^* = \hat{A}_1$$

et $B_1 = \hat{B}_1$; $C_1 = \hat{C}_1$ s'ensuivent par le même procédé

L'approche de la matrice de Hankel, fait qu'elle donne la raison pour laquelle l'approximation équilibrée est bonne mais pas optimale dans un certain sens.

Puisque $\Delta_1 = U_1 \Sigma_1 V_1^T$ est une approximation optimale de la matrice H , elle n'est donc pas de Hankel.

Remarquablement, cependant Adamjan, Arov, et Krein [] ont montré que pour des ~~matrices~~ matrices infinies de Hankel, données par un scalaire $f(z)$, il y'a plus d'une approximation optimale de rang r , mais en fait qu'une seule matrice de Hankel existe.

chapitre 3

ALGORITHMES ETUDIÉS

3.1 REDUCTION DES VARIABLES D'ETATS DU SYSTEME

(A, B, C)_n.

Algorithme:

- 1°) Résoudre les équations de Lyapunov :

$$AW_c + W_c A^T = -BB^T \quad (1)$$

$$A^T W_0 + W_0 A = -C^T C$$

correspondantes au système continu :

et donner les Grammians W_c, W_0 , comme solutions uniques de (1).

- 2°) Décomposition de Cholesky de W_c .

$$W_c = L_c \cdot L_c^T \quad \text{avec } L_c = U_c \Sigma_c^{1/2}; \quad L_c^T = \Sigma_c^{1/2} \cdot U_c^T$$

- 3°) Former la matrice :

$$\bar{W}_0 = L_c^T W_0 L_c$$

- 4°) Décomposition en valeurs propres et vecteurs propres de \bar{W}_0 :

$$\bar{W}_0 = \bar{U}_0 \Sigma^2 \bar{U}_0^T \quad \text{avec } \bar{U}_0 \bar{U}_0^T = I$$

- 5°) la transformation équilibrée est alors :

$$T = \Sigma^{1/2} \bar{U}_0^T \Sigma_c^{-1/2} U_c^T$$

- 6°) Calculer la représentation équilibrée $(\hat{A}, \hat{B}, \hat{C})$

$$\text{comme } \hat{A} = T A T^{-1}$$

$$\hat{B} = T B$$

$$\hat{C} = C T^{-1}$$

- 7°) Donner le modèle réduit qui correspond au 1^{er} bloc du système exact :

$(\hat{A}_1, \hat{B}_1, \hat{C}_1)$ de $(\hat{A}, \hat{B}, \hat{C})$ d'ordre r .

NB: $W_c = (U_c \Sigma_c^{1/2}) (\Sigma_c^{1/2} U_c^T)$; $\bar{W}_0 = (\bar{U}_0 \Sigma) (\bar{U}_0 \Sigma)^T$.

3.1.1 ALGORITHME DE RESOLUTION DE L'EQUATION DE LYAPUNOV DU SYSTEME $(A, B, C)_N$ PAR LA METHODE DE BINGULAC [3] :

Soit l'équation de Lyapunov :

$$PA + A^T P = -Q \quad (1) \quad ; \quad Q \triangleq B B^T$$

La matrice P inconnue est calculée par la méthode de Bingulac suivante qui convertit (1) en M ($M = \frac{N(N+1)}{2}$) équations linéaires algébriques :

$$[\alpha] P_v = -0.5 Q_v \quad (2)$$

$$[\alpha] = \{\alpha_{nm}\} \quad \begin{array}{l} n = 1 \div M \\ m = 1 \div M \end{array}$$

$$P_v = \{P_m\} \quad \text{triangulaire supérieure de l'inconnue } P_{NN}$$

$$P = [P_{ij}] \quad \begin{array}{l} i = 1 \div N \\ j = 1 \div N \end{array}$$

$$P_v = \{P_{11}, P_{12}, \dots, P_{1N}, P_{22}, P_{23}, \dots, P_{2N}, P_{33}, \dots, P_{(N-1)N}, P_{NN}\} \quad (3)$$

$$Q_v = \{q_m\} \quad m = 1, M$$

Q_{NN} matrice donnée

$$Q_v = \{q_{11}, 2q_{12}, \dots, 2q_{1N}, q_{22}, \dots, 2q_{2N}, q_{33}, \dots, q_{(N-1)N}, 2q_{(N-1)N}, q_{NN}\} \quad (4)$$

$[\alpha]_{MM}$ dépend des a_{ij} de A .

le problème consiste à construire $[\alpha]_{MM}$

$$\alpha_{nm} = \alpha_{nm}(a_{ij}) \quad ; \quad \begin{array}{l} i, j = 1 \div N \\ n, m = 1 \div M \end{array}$$

Description de la méthode :

d'après (1) on écrit : $U P_v = -R_v$

avec $U = 2[\alpha]$

P_v donné par (3) ; $P_v \in \mathbb{R}^M$

$R_v = \{\Gamma_m\}$; $R_v \in \mathbb{R}^M$; Γ_m est la partie supérieure de Q comme P_v .

c'est à dire : $R_v = \{q_{11}, q_{12}, \dots, q_{1N}, q_{22}, q_{23}, \dots, q_{2N}, q_{33}, \dots, q_{(N-1)(N-1)}, q_{(N-1)N}, q_{NN}\}$

$U_{MM} = \{U_{mm}^{(6)}\}$ dépend de a_{ij}

Il a été démontré (5) que U est obtenue par l'algorithme suivant en 3 étapes.

Algorithme:

1^{ère} étape: Construire $L = L^T$ donnée par:

$$L = \begin{bmatrix} 1 & 2 & 3 & \dots & \dots & \dots & N \\ 2 & N+1 & N+2 & & & & 2N-1 \\ 3 & N+2 & N+3 & 2N+1 & \dots & & 3N-3 \\ 4 & N+3 & N+4 & 3N-2 & \dots & & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & M-2, M-1 & \vdots \\ N & 2N-1 & 3N-3 & \dots & \dots & M-1, M & \vdots \end{bmatrix}$$

c'est à dire: pour $N=3$

$$L = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$

pour $N=4$:

$$L = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 6 & 7 \\ 3 & 6 & 8 & 9 \\ 4 & 7 & 9 & 10 \end{bmatrix}$$

TABLEAU I:
CAS $N=3$

i	j	k	n	m	
1	1	1	1	1	
		2	2	2 ← X	
		3	3	3 ← Y	
	2	1	1	1	2
		2	2	2	4
		3	3	3	5
2	3	1	1	3	
		2	2	5	
		3	3	6	
	1	1	1	2	4
		2	2	4	2 ← X
		3	3	5	4 ← Z
3	3	1	1	3	
		2	2	5	
		3	3	6	
	1	1	1	3	4
		2	2	6	2
		3	3	9	3
2	1	1	3	2	
	2	2	6	4	
	3	3	9	5	
3	3	1	3	3 ← Y	
		2	6	5 ← Z	
		3	9	6	

TABLEAU II

CAS N=4

i	j	k	n	m
1	1	1 2 3 4	1 2 3 4	1 2 3 4
	2	1 2 3 4	1 2 3 4	2 5 6 7
	3	1 2 3 4	1 2 3 4	3 6 8 9
	4	1 2 3 4	1 2 3 4	4 7 9 10
2	1	1 2 3 4	2 5 6 7	1 2 3 4
	2	1 2 3 4	2 5 6 7	2 5 6 7
	3	1 2 3 4	2 5 6 7	3 6 8 9
	4	1 2 3 4	2 5 6 7	4 7 8 10
3	1	1 2 3 4	3 6 8 9	1 2 3 4
	2	1 2 3 4	3 6 8 9	2 5 6 7
	3	1 2 3 4	3 6 8 9	3 6 8 9
	4	1 2 3 4	3 6 8 9	4 7 8 10
4	1	1 2 3 4	4 7 10 4	1 2 3 4

2^e étape :

Construire la matrice $V = \{V_{nm}\}$ de dim $(M \times M)$
en appliquant :

$$V_{nm} = A_{ji} \quad (8)$$

où les indices n et m ($n, m = 1, 2, \dots, M$) sont donnés
par les éléments de la matrice auxiliaire L .

$$n = L_{ik} ; m = L_{jk} ; i, j, k = 1, 2, \dots, N \quad (9).$$

3^e étape :

La matrice U nécessaire peut être obtenue en multipliant
par 2 tous les éléments de la matrice V , dont les indices
correspondent aux éléments de la diagonale de la matrice L .

Ces indices sont :

$$1, N+1, 2N, 3N-2, \dots, M-2, M.$$

Le tableau (I) donne les valeurs de $n = n(i, j, k)$ et
 $m = m(i, j, k)$ pour les cas $N=3$ et $M=6$ alors que le tableau (II)
donne une partie de ces valeurs pour $N=4$ et $M=10$.

Notons que dans le tableau (II), deux lignes dénotés par
 $X(i=1, j=1, k=2)$ et $(i=1, j=2, k=1)$, mènent à la même
paire de $n=2$ et $m=2$.

Dans ce cas, en appliquant la relation (8), l'élément V_{22}
devrait être calculé par :

$$V_{22} = A_{11} + A_{12}$$

La même chose s'applique aux cas dénotés dans le tableau (II)
par Y et Z .

N.B : Nous donnerons ultérieurement, le programme écrit en
FORTRAN IV de l'exécution de cet algorithme.

3.1.2 : ALGORITHME DE DECOMPOSITION DE SCHOLESKY [4]

Soit A symétrique définie positive, cherchons R triangulaire inférieure telle que :

$$A = R R^T$$

Par identification et cela dans l'ordre suivant :

posons : $(r_{ij}) = R$ pour $r_{ij} = 0$ si $i < j$

On doit avoir donc :

$$a_{ij} = \sum_{k=1}^{k=i} r_{ik} \cdot r_{kj}^T = \sum_{k=1}^{k=j} r_{ik} \cdot r_{jk} \quad \text{où } i \leq j$$

d'où
$$a_{ij} = \sum_{k=1}^{k=i} r_{ik} \cdot r_{jk}$$

Si on identifie le terme a_{11} , cela donne :

$$a_{11} = r_{11} \cdot r_{11} = r_{11}^2 \quad \text{d'où } r_{11} = \pm \sqrt{a_{11}} \neq 0.$$

On choisit la détermination (+), puis on identifie les termes de la première colonne :

$$a_{ij} = r_{11} \cdot r_{1j} \quad \text{d'où } r_{1j} = \frac{a_{1j}}{r_{11}} \quad (j = 2, \dots, n)$$

Par conséquent la première colonne de R .

La $i^{\text{ème}}$ colonne sera déterminée par :

$$a_{ii} = \sum_{k=1}^{k=i-1} r_{ik}^2 + r_{ii}^2 \quad (\text{formule précédente où } i=j).$$

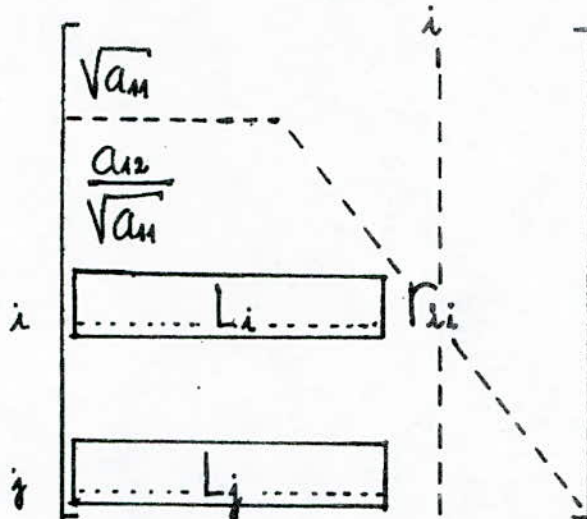
$$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{k=i-1} r_{ik}^2} \quad (1)$$

On choisit la détermination positive (+)

puis :
$$a_{ij} = \sum_{k=1}^{k=i-1} r_{ik} r_{jk} + r_{ik} r_{ji}$$

Formule précédente pour i, j avec $i \leq j$.

$$\Gamma_{ji} = \frac{1}{\Gamma_{ii}} \left[a_{ij} - \sum_{k=1}^{i-1} \Gamma_{ik} \cdot \Gamma_{jk} \right] \quad (2)$$



D'où le schéma : on remplace « a_{ii} » par la racine, carrée de $a_{ii} - \sum$ des carrés des éléments de Γ_i .

$$\Gamma_{ii}^2 = a_{ii} - L_i L_i^T$$

Ensuite on passe à la détermination de Γ_{ji} , on prend a_{ij} (ou a_{ij} symétrique) et on fait :

$$\Gamma_{ij} = [a_{ji} - (L_i L_j^T)] / \Gamma_{ii} \quad (j > 1)$$

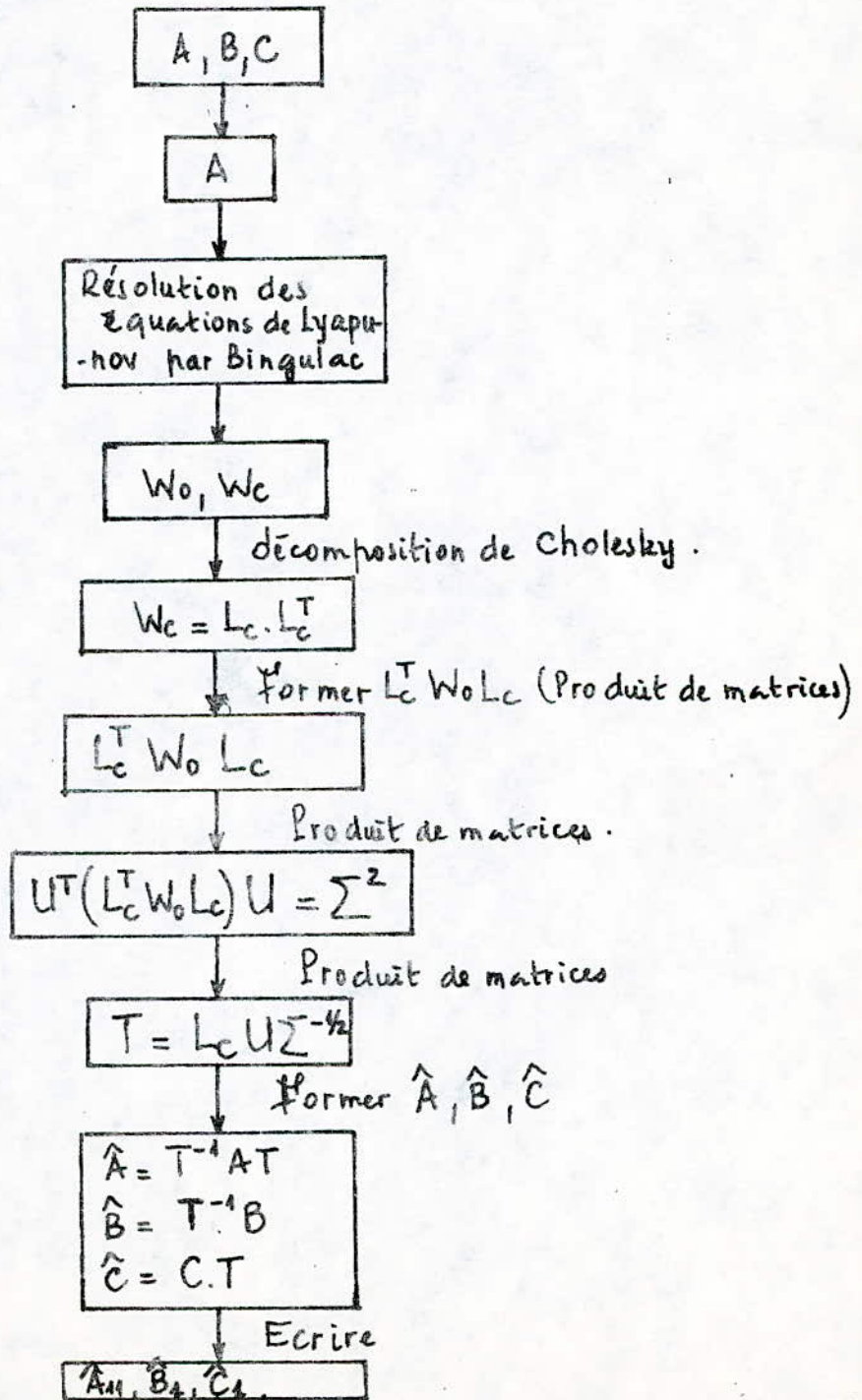
ALGORITHME :

- avec $\lambda = \Gamma$
- 1°) pour $k = 1, 2, \dots, n$
 - 1) pour $i = 1, 2, \dots, k-1$
 - 1) $a_{ki} \leftarrow \lambda_{ki} = \lambda_{ii}^{-1/2} \left(a_{ki} - \sum_{j=1}^{i-1} \lambda_{ij} \lambda_{kj} \right)$
 - 2) $a_{kk} \leftarrow \lambda_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} \lambda_{kj}^2}$

3.13: Organigramme de réduction des matrices A, B, C

correspondants aux Equations de Lyapunov (*)

$$(*) \begin{cases} AW_c + W_c A^T + BB^T = 0 \\ A^T W_0 + W_0 A + C^T C = 0 \end{cases}$$



3.2 : Algorithme de réduction des paramètres de Markov : (Approche de Hankel)

. Données $(h_1, h_2, \dots, h_{2N+1})$ de $\dim(p \times m)$

1) Former la matrice de Hankel :

$$H = \begin{bmatrix} h_1 & h_2 & \dots & \dots & h_{N+1} \\ h_2 & h_3 & \dots & \dots & h_{N+2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ h_{N+1} & \dots & \dots & \dots & h_{2N+1} \end{bmatrix}$$

$$\dim H = [(N+1)p \times (N+1)m]$$

2) Décomposition de H en composantes principales (S.V.D)

$$H = U \Sigma V^T$$

$$= \begin{bmatrix} u_1 \\ \vdots \\ u_{N+1} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & \sigma_{N+1} \end{bmatrix} \cdot [v_1^T, v_2^T, \dots, v_{N+1}^T]$$

3) Former les matrices finies d'observabilité et de contrôlabilité de H .

$$H = (U \Sigma^{1/2}) (\Sigma^{1/2} V^T) = \Theta \cdot G.$$

$$\text{avec } \Theta = U \Sigma^{1/2} \text{ de } \dim [p \times (N+1)]$$

$$G = \Sigma^{1/2} V^T \text{ de } \dim [(N+1) \times m]$$

$$[H]_{(N+1)} = [U \Sigma^{1/2}]_{(N+1)} \cdot [\Sigma^{1/2} V^T]_{(N+1)} \quad .35.$$

- 4) Trouver O^\uparrow en éliminant les p premières lignes de Θ .
 - Trouver G en éliminant les m dernières colonnes de G .

5) Trouver $O^+ = \Sigma^{-1/2} U^T$

- 6) Former les matrices de la réalisation exacte.

$$A = O^+ O^\uparrow \quad (\text{produit de matrices})$$

$$B = m \text{ premières colonnes de } G.$$

$$C = p \text{ premières lignes de } \Theta.$$

- 7) Former les produits de matrices:

$$\Theta_r = [u_1, \dots, u_r] \begin{bmatrix} \sqrt{\lambda_1} & & & \\ & \ddots & & \\ & & 0 & \\ & & & \ddots \\ & & & & \sqrt{\lambda_r} \end{bmatrix}$$

$$G_r = \begin{bmatrix} \sqrt{\lambda_1} & & & \\ & \ddots & & \\ & & 0 & \\ & & & \ddots \\ & & & & \sqrt{\lambda_r} \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix}$$

$$O_r^\uparrow = \begin{bmatrix} \sim \\ \Theta_r \\ \hline 0 \end{bmatrix}$$

$$O_r^+ = \Sigma_r^{-1/2} U_r^T$$

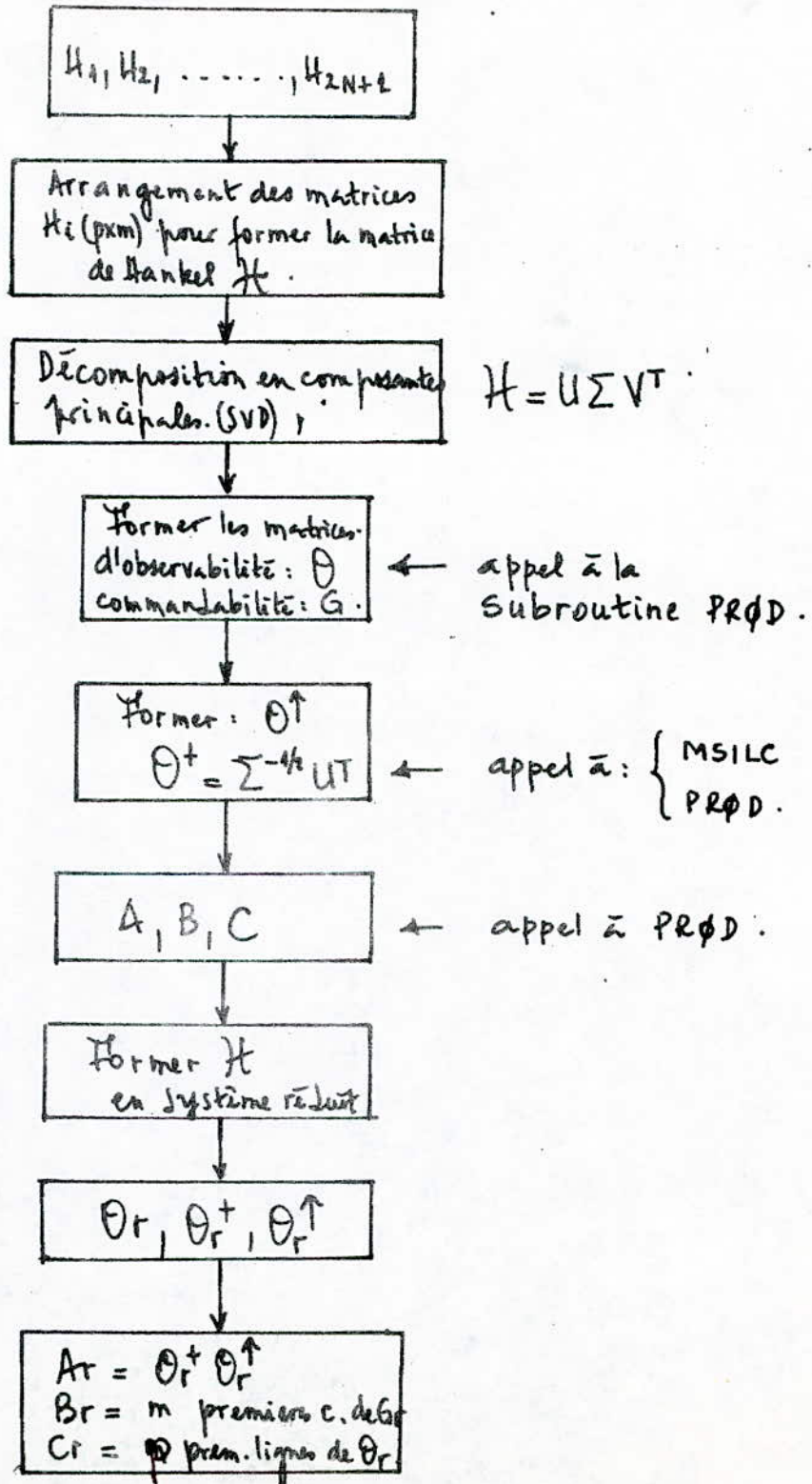
8.) Modèle réduit :

$$A_r = \Theta_r^+ \Theta_r^\uparrow$$

$B_r = m$ premières colonnes de G_r

$C_r = p$ premières lignes de Θ_r .

3.2.1 Organigramme de réduction des paramètres de Markov. (Approche de Hankel)



38.
3.22. Algorithme de décomposition en composantes principales:

Soit la matrice $H \in \mathbb{R}^{m \times n}$, on peut toujours la décomposer en ~~en~~ composantes principales comme :

$$H = U \Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^T$$

où $U = [u_1, u_2, \dots, u_r] \in \mathbb{R}^{m \times r}$

et $V^T = [v_1, v_2, \dots, v_r]^T \in \mathbb{R}^{r \times n}$

et $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ avec $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$

sont tels que :

$$H^T H V = V \Sigma^2 \quad \text{et} \quad U = H V \Sigma^{-1} ; \quad V^T V = U^T U = I_r$$

et $\text{rang } H = r$

on a donc : $H^T H v_i = \sigma_i^2 v_i ; \quad \forall i = 1, \dots, r$

$\text{rang } H^T H = \text{rang } H$

les nombres réels positifs σ_i sont appelés, les valeurs singulières de H , et ils sont les valeurs propres non nulles de $H^T H$.

Les valeurs v_i sont appelées "les valeurs singulières de H ".

(vecteurs propres orthogonaux de $H^T H$).

on a donc $H H^T U = U \Sigma^2$

où $H H^T u_i = \sigma_i^2 u_i ; \quad \forall i = 1, \dots, r$.

la procédure suggérée de construction des composantes principales, se fait en 2 étapes :

1^{re} étape : Résoudre le problème des valeurs propres

$$H^T H v_i = \sigma_i^2 v_i \quad \text{pour } \sigma_i^2 \text{ et les vecteurs propres normalisés } v_i$$

2^e étape : Calculer u_i à partir de la simple relation :

$$u_i = \frac{1}{\sigma_i} H v_i \quad (\text{ceci est un algorithme théorique})$$

chapitre 4

PROGRAMMATION et SIMULATION

4.1 PROGRAMMATION DETAILLÉE :

Afin de mieux cerner le problème de la complexité dans le cadre de la réduction de systèmes linéaires dynamiques, nous présentons ici un programme principal à chacune des deux méthodes de réduction en fonctions de leur organigramme correspondant.

Nous présenterons tout d'abord le programme principal appelé "APPROX" relatif à la réduction d'un système linéaire équilibré (A, B, C) représenté par le modèle :

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

puis le programme principal appelé "HANKEL" relatif à la réduction d'un système présenté par ses réponses impulsionnelles h_i appelées paramètres de Markov, $h_i = CA^{i-1}B \quad i=1, \dots, n$.

Ces programmes font appels parfois à quelques ~~subroutines~~ routines usuelles existantes en bibliothèque mathématique de l'ordinateur MITRA 125 utilisé (VOIR ANNEXE).

De ce fait, la programmation en FORTRAN IV a constitué pour nous un précieux outil de travail sinon la plus importante étape de notre étude.

Il va sans dire que le concours appréciable des notions élémentaires de FORTRAN IV a permis d'acquiesce à nos programmes une dimension relativement satisfaisante.

D'autres ~~subroutines~~ routines de travail très usuelles ont été utilisées [34].


```

DIMENSION AA(2,2),L(2,2),VV(3,3),U(3,3),WO(2,2),WC(2,2)
DIMENSION TR(2),P(2,2),CCF(3),VPR(2),VPT(2),A(2,2),VALFUR(2)
DIMENSION CC(2,2),CO(2,2),CO(1,2),PR(2,1),RLCN(2,2),RLCT(2,2)
DIMENSION TJ(2)
DIMENSION XTT(2,2)
DIMENSION IC(2),RT(2,2),RES(2,2),F(2,2),TT(2,2),XR(2,2)
DIMENSION XT(2,2)
DIMENSION XRI(2,2),AR(2,2),AB(2,2),PR(2,1),CR(1,2)
DIMENSION TECT(2,2),FR(2,2)
DIMENSION F2(2,2),F4(2,2)
DIMENSION CV(3),RLC(3),ILT(2)
EPS=1.E-06
READ(105,100)N
100 FORMAT(12)
READ(105,101)((AA(I,J),I=1,2),J=1,2)
101 FORMAT(2F5.2)
CALL LYAFCV(L,AA,U,WC,WO,N,P)
N=2
GV(1)=WC(1,1)
GV(2)=WC(1,2)
GV(3)=WC(2,2)
CALL CHOLCS(GV,N,RLC,EPS,DET,ILT,1)
WRITE(108,126)(RLC(I),I=1,3)
126 FORMAT(15X,1P.4)
RLCN(1,1)=RLC(1)
RLCN(1,2)=RLC(2)
RLCN(2,1)=0
RLCN(2,2)=RLC(3)
WRITE(108,102)
102 FORMAT(5X,'LA MATRICE RLCI EST:',/,5X,33(1H*))
WRITE(108,114)((RLCN(I,J),I=1,2),J=1,2)
114 FORMAT(15X,F6.2,5X,F6.2//)
DO 20 I=1,2
DO 20 J=1,2
RLCT(J,I)=RLCN(I,J)
20 CONTINUE
WRITE(108,114)((RLCT(I,J),I=1,2),J=1,2)
CALL PROD(RLCT,WO,R,2,2,2)
WRITE(108,103)
103 FORMAT(5X,'LA MATRICE R EST:',/,5X,33(1H*))
WRITE(108,114)((R(I,J),I=1,2),J=1,2)
CALL PROD(R,RLCN,RT,2,2,2)
WRITE(108,104)
104 FORMAT(5X,'LA MATRICE RT EST:',/,5X,33(1H*))
WRITE(108,114)((RT(I,J),I=1,2),J=1,2)
DO 40 I=1,2
DO 40 J=1,2
A(I,J)=RT(I,J)
40 CONTINUE
CALL DANJIE(A,2,TR,TT,CCF,VPR,VPT,C,EPS)
WRITE(108,105)
105 FORMAT(5X,'LA MATRICE VPR EST:',/,5X,33(1H*))
WRITE(108,117)(VPR(I),I=1,2)
117 FORMAT(15X,2(F5.2,2X))
VALFUR(I)=VPR(I)
DO 50 I=1,2
DO 50 J=1,2
XT(I,J)=ST(I,J)
50 CONTINUE

```



```

CALL FRCY(XT,2,IEC,LR,IC,VALEUR,RES)
WRITE(108,106)
106 FORMAT(5X,'LA MATRICE XT EST:',/,5X,33(1H*))
WRITE(108,114)((XT(I,J),I=1,2),J=1,2)
DO 30 I=1,2
DO 30 J=1,2
XTT(J,I)=XT(I,J)
30 CONTINUE
WRITE(108,114)((XTT(J,I),J=1,2),I=1,2)
CALL PROC(XTT,RT,RES,2,2,2)
WRITE(108,107)
107 FORMAT(5X,'LA MATRICE RES EST:',/,5X,33(1H*))
WRITE(108,114)((RES(I,J),J=1,2),I=1,2)
CALL PROC(RES,XT,F,2,2,2)
WRITE(108,108)
108 FORMAT(5X,'LA MATRICE F EST:',/,5X,33(1H*))
WRITE(108,114)(F(I,J))
CALL PROC(RIC,XT,TT,2,2,2)
WRITE(108,110)
110 FORMAT(5X,'LA MATRICE TT EST:',/,5X,33(1H*))
WRITE(108,114)((TT(I,J),I=1,2),J=1,2)
DO 5 I=1,2
F2(I,I)=ABS(SQRT(F(I,I)))
F4(I,I)=ABS(SQRT(F2(I,I)))
FR(I,I)=1/FA(I,I)
5 CONTINUE
CALL PROC(TT,FR,XR,2,2,2)
WRITE(108,111)
111 FORMAT(5X,'LA MATRICE XR EST:',/,5X,33(1H*))
WRITE(108,114)((XR(I,J),I=1,2),J=1,2)
CALL FRIN(XR,XRT,2,0,DET,EPS,TL,IC)
WRITE(108,112)
112 FORMAT(15X,16.2,5X,F6.2///)
WRITE(108,114)((XRT(I,J),I=1,2),J=1,2)
CALL PROC(YBT,AA,AB,2,2,2)
WRITE(108,113)
113 FORMAT(5X,'LA MATRICE AB EST:',/,5X,33(1H*))
WRITE(108,114)((AB(I,J),I=1,2),J=1,2)
CALL PROC(AI,XR,AB,2,2,2)
WRITE(108,123)
123 FORMAT(5X,'LA MATRICE AB EST:',/,5X,33(1H*))
WRITE(108,114)((AB(I,J),I=1,2),J=1,2)
CALL PROC(YBT,PB,BF,2,2,1)
WRITE(108,124)
124 FORMAT(5X,'LA MATRICE BR EST:',/,5X,33(1H*))
J=1
WRITE(108,114)(BR(I,J),I=1,2)
CALL PROC(CC,XR,CR,2,1,2)
WRITE(108,125)
125 FORMAT(5X,'LA MATRICE CR EST:',/,5X,33(1H*))
I=1
WRITE(108,114)(CR(I,J),J=1,2)
STOP
END

```

```

DIMENSION AD(2,2),PD(2,1),CD(1,2),AC(2,2),BC(2,1),CC(1,2),TI(2,2)
DIMENSION DIF(2,2),S(2,2),TNVS(2,2)
EPS=1.E-06
READ(105,101)(TI(I,J))
READ(105,101)(AD(I,J))
READ(105,101)(PD(I,J))
READ(105,101)(CD(I,J))
101 FORMAT(2F5.2)
S(I,J)=0.0
DO 20 I=1,2
DO 20 J=1,2
S(I,J)=S(I,J)+AD(I,J)+TI(I,J)
20 CONTINUE
WRITE(108,100)((S(I,J),I=1,2),J=1,2)
DIF(I,J)=0.0
DO 30 I=1,2
DO 30 J=1,2
DIF(I,J)=DIF(I,J)+(AD(I,J)-TI(I,J))
30 CONTINUE
WRITE(108,100)((DIF(I,J),I=1,2),J=1,2)
CALL MRINV(S,INVS,2,0,DFT,EPS,IL,IC)
WRITE(108,100)(TNVS(I,J),I=1,2),J=1,2)
CALL PROD(TNVS,DIF,AC,2,2,2)
WRITE(108,100)(AC(I,J),I=1,2),J=1,2)
DO 40 I=1,2
DO 40 J=1,2
SR=SQRT(2)*INVS(I,J)
40 CONTINUE
CALL PROD(SR,BC,PC,2,2,1)
J=1
WRITE(108,110)(BC(I,J),I=1,2)
I=1
DO 50 J=1,2
G=SQRT(2)*CD(I,J)
50 CONTINUE
CALL PROD(G,TNVS,CC,1,2,2)
WRITE(108,120)(CC(I,J))
100 FORMAT(2(15X,(F5.2,2X)))
110 FORMAT(15X,F5.2//)
120 FORMAT(15X,F5.2,5X,F5.2)
STOP
END

```

```

SUBROUTINE LYADCV(L,AA,I,WC,PO,N,M)
DIMENSION AA(2,2),L(2,2),VV(3,3),PB(2,1),PRT(1,2),L(3,3)
DIMENSION CT(2,1),CC(1,2),CCX(2,2),CCX(2,2),WC(2,2),WC(2,2)
DIMENSION CC(2,2),CO(2,2),CCT(2,1)
DIMENSION CV(3),CVC(3),SS(2,2)
DIMENSION V(3,3)
N=N+1
DO 16 T=1,2
16 WRITE(108,105)(AA(T,J),J=1,2)
M=N*(N+1)/2
K=1
DO 10 T=1,2
DO 10 J=T,2
IF(J.LT.1)GO TO 10
L(T,J)=I
L(J,1)=K
K=K+1
10 CONTINUE
DO 11 I=1,3
DO 11 J=1,3
11 VV(I,J)=0
DO 12 I=1,2
DO 12 J=1,2
DO 12 K=1,2
IK=L(I,K)
JK=L(J,K)
12 VV(IK,JK)=AA(J,I)+VV(IK,JK)
WRITE(108,104)
104 FORMAT(5X,'LA MATRICE VV OBTENUE EST:',/,5X,3*(1H*))
105 FORMAT(15X,F10.2,5X,F10.2,5X,F10.2/)
106 FORMAT(15X,F10.2,5X,F10.2//)
DO 40 I=1,2
DO 40 J=1,3
II=L(I,I)
VV(II,J)=2.*VV(II,J)
40 CONTINUE
DO 60 I=1,3
DO 60 J=1,3
L(I,J)=VV(I,J)
60 CONTINUE
WRITE(108,109)
109 FORMAT(5X,'LA MATRICE L EST:',/,5X,3*(1H*))
WRITE(108,107)(L(II,J),II=1,3),J=1,3)
J=1
READ(105,120)(PB(I,J),I=1,2)
120 FORMAT(2F6.1)
WRITE(108,107)(PB(I,J),I=1,2)
107 FORMAT(15X,F6.2/)
J=1
DO 8 I=1,2
PRT(J,I)=PB(I,J)
8 CONTINUE
WRITE(108,200)(PRT(J,I),I=1,2)
200 FORMAT(15X,2(F6.2,2X))
DO 4 I=1,2
DO 4 J=1,2
K1=1
CC(I,J)=PB(I,K1)*PRT(K1,J)
4 CONTINUE

```

```

DO 36 I=1,2
WRITE (106,106) (CC(I,J),J=1,2)
36 CONTINUE
106 FORMAT(15X,2(F6.2,2X))
I=1
READ (105,200) (CC(I,J),J=1,2)
210 FORMAT(2F6.1)
WRITE (108,200) (CC(I,J),J=1,2)
I=1
DO 9 J=1,2
CCT(I,I)=CC(I,J)
c CONTINUE
I=1
WRITE (108,110) (CCT(J,I),J=1,2)
110 FORMAT(15X,16.2//)
DO 5 I=1,2
DO 5 J=1,2
K2=1
CO(I,J)=CCT(I,K2)*CC(K2,J)
5 CONTINUE
WRITE (108,105) ((CO(I,J),I=1,2),J=1,2)
DO 20 I=1,2
DO 20 J=1,2
COX(I,J)=-CO(I,J)
COY(I,J)=-CO(I,J)
20 CONTINUE
LL=3
KK=1
DO 1 I=1,2
DO 1 J=1,2
IF (J.LT.I) GO TO 1
CV(KK)=COX(I,J)
KK=KK+1
1 CONTINUE
CALL WFSOL (I,CV,0,0,FRS)
WRITE (108,400) (CV(I),I=1,3)
400 FORMAT(15X,FR.4//)
I1=1
J1=1
DO 2 KK=1,N
WC(I1,J1)=CV(KK)
WC(J1,I1)=CV(KK)
IF (J1.EQ.2) GO TO 3
J1=J1+1
GO TO 2
3 I1=I1+1
J1=I1
2 CONTINUE
WRITE (108,114)
114 FORMAT(5X,' LA MATRIX OF EST: ',/,5X,33(1F*))
WRITE (108,112) ((WC(I,J),I=1,2),J=1,2)
112 FORMAT(15X,16.2)
DO 800 I=1,2
DO 800 J=1,2
SS(I,J)=AA(J,I)
800 CONTINUE
ILL=3
K3=1
DO 500 I=1,2
DO 500 J=1,2

```

FADI..

```

      IF(J.LT.1)GO TO 500
      GVO(K3)=GEX(T,J)
      K3=K3+1
500  CONTINUE
      DO 501 J=1,2
501  WRITE(108,105)(SS(T,J),J=1,2)
      DO 503 I=1,3
      DO 503 J=1,3
503  V(I,J)=0
      DO 504 II=1,2
      DO 504 JJ=1,2
      DO 504 K4=1,2
      IK1=L(II,K4)
      JK1=L(JJ,K4)
504  V(IK1,JK1)=S*(JJ,II)+V(IK1,JK1)
      DO 505 II=1,2
      DO 505 JJ=1,2
      II1=L(II,II)
      V(II1,JJ)=2.*V(II1,JJ)
505  CONTINUE
      DO 70 I=1,3
      DO 70 J=1,3
      U(1,J)=V(J,J)
70  CONTINUE
      WRITE(108,107)((U(T1,J1),T1=1,3),J1=1,3)
      CALL RESOL(L,GVO,H,0,EPS)
      WRITE(108,400)(GVO(IL),IL=1,3)
      I2=1
      J2=1
      DO 506 K2=1,N
      W0(I2,J2)=GVO(K2)
      W0(J2,I2)=GVO(K2)
      IF(J2.EQ.2)GO TO 507
      J2=J2+1
      GO TO 506
507  I2=I2+1
      J2=J2
506  CONTINUE
      WRITE(108,115)
115  FORMAT(5X,'LA MATRICE VC EST:',/,5X,33(1F*))
      WRITE(108,112)((W0(I3,J3),I3=1,2),J3=1,3)
      RETURN
      END

```

PADI..

```

SUBROUTINE PROD(A,B,C,M,N,IP)
DIMENSION A(M,N),B(N,IP),C(M,IP)
C(I,J)=0
DO 10 I=1,M
DO 10 J=1,N
DO 20 K=1,IP
20 C(I,J)=C(I,J)+A(I,K)*B(K,J)
10 CONTINUE
WRITE(108,102)
102 FORMAT(2X,'LE PRODUIT DE LA MATRICE A PAR LA MATRICE B EST:*')
WRITE(108,120)
120 FORMAT(2X,61(1F*))
WRITE(108,100)((C(I,J),I=1,M),J=1,IP)
100 FORMAT((15X,3(F15.6,2X))//)
RETURN
END

```



```

DIMENSION A(3,3),U(3,3),V(3,3),W(3),RV1(3)
DIMENSION CT(3,3),A1(3,3),B1(1,3),C1(3,1),UT(3,3)
DIMENSION SIGMA(3,3),SIGMA1(3,3),OF(3,3),OB(3,3),CF(3,3)
DIMENSION CPR(3,3)
M1=1
N1=1
N=3
M=3
NP=3
READ(105,100)((A(I,J),I=1,3),J=1,3)
100 FORMAT(F6.4)
CALL SVD (NP,M,N,A,W,TRUE,L,TRUE,V,TERR,RV1)
DO 60 I=1,3
DO 70 J=1,3
SIGMA(I,J)=0.0
70 CONTINUE
SIGMA(I,I)=SQRT(W(I))
60 CONTINUE
CALL PROC (L,SIGMA,OB,3,3,3)
WRITE(108,104)
104 FORMAT(5X,'LA MATRICE OB EST:',/,5X,33(1H*))
WRITE(108,110)((OB(I,J),I=1,3),J=1,3)
CALL PROC(SIGMA,V,CT,3,3,3)
WRITE(108,110)((CT(I,J),I=1,3),J=1,3)
DO 80 I=1,3
DO 90 J=1,3
SIGMA1(I,J)=0.0
90 CONTINUE
S=SIGMA(I,I)
IF(S.EQ.0.)GO TO 80
SIGMA1(I,J)=1./S
80 CONTINUE
DO 10 I=1,3
DO 10 J=1,3
UT(J,I)=U(I,J)
10 CONTINUE
CALL PROC (SIGMA1,UT,OP,3,3,3)
WRITE(108,106)
106 FORMAT(5X,'LA MATRICE OP EST:',/,5X,33(1H*))
WRITE(108,110)((OP(I,J),I=1,3),J=1,3)
DO 9 I=1,3
DO 9 J=1,3
CPR(I,J)=OB(I,J)
9 CONTINUE
IP=1
K=IP
L=M-10
DO 200 I=1,L
DO 200 J=1,L
K=K+1
OF(I,J)=CPR(K,J)
200 CONTINUE
K=L
DO 201 I=1,IP
DO 201 J=1,I
K=K+1
CF(K,J)=0.
201 CONTINUE
WRITE(108,107)

```


PART..

```

107 FORMAT(5X,'LA MATRICE CF EST:',/,5X,33(1H*))
WRITE(108,110)((CF(I,J),I=1,3),J=1,3)
CALL PROC(CF,CF,A1,3,3,3)
WRITE(108,111)
109 FORMAT(5X,'LA MATRICE A1 EST:',/,5X,33(1H*))
WRITE(108,110)((A1(I,J),J=1,3),I=1,3)
110 FORMAT((15X,3(F15.6,2X))//)
J=1
DO 40 I=1,3
B1(I,J)=C1(I,J)
WRITE(108,111)
111 FORMAT(5X,'LA MATRICE P1 EST:',/,5X,33(1H*))
WRITE(108,140)(P1(I,J))
40 CONTINUE
140 FORMAT(15X,(F12.6,2X)//)
I=1
DO 50 J=1,3
C1(I,J)=CPR(I,J)
WRITE(108,112)
112 FORMAT(5X,'LA MATRICE C1 EST:',/,5X,33(1H*))
WRITE(108,150)(C1(I,J))
50 CONTINUE
150 FORMAT(15X,2(F12.6,2X))
STOP
END
NO 109

```

PADT..

```

SUBROUTINE HANKEL (AA,A,N1,M1,M,N)
DIMENSION AA(1,1),A(3,3)
N=2
IN=1
IP=1
M=N+1
DO 10 JJ=1,P
DO 10 T=1,TR
DO 10 J=1,TF
READ(105,100)AA(T,J)
IH=JJ
IH1=1
5 K=I+(JJ-TF)*TR
L=J+(JJ-TF1)*IF
A(K,L)=AA(T,J)
IH=IH-1
IH1=IH1+1
IF(IH1.LE.JJ)GO TO 5
10 CONTINUE
12 JF=II
II=II+1
IF(II.GT.M)GO TO 20
I=1
13 J=1
15 READ(105,100)AA(T,J)
IV=IF
IW=N
17 K=I+IV*IF
L=J+IW*IF
A(K,L)=AA(T,J)
IV=IV+1
IW=IW-1
IF(IV.LE.N)GO TO 17
J=J+1
IF(J.LE.TF)GO TO 15
I=I+1
IF(I.LE.TR)GO TO 13
GO TO 12
20 WRITE(10P,200)((A(T,J),I=1,3),J=1,3)
100 FORMAT(F5.2)
200 FORMAT((15X,3(F5.2,2X))//)
RETURN
END

```

```

SUBROUTINE SVD (NM,M,N,A,W,MATH,U,MATV,V,IERF,RV1)
INTEGER J,K,L,M,N,II,I1,KK,K1,LL,L1,MM,NN,ITS,ITER
REAL A(3,3),W(3),U(3,3),V(3,3),RV1(3)
REAL C,F,G,H,S,X,Y,Z,SCALE,ANORM
LOGICAL MATH,MATV
MATH=.TRUE.
MATV=.TRUE.
M=3
N=3
MM=3
IERF=0
WRITE(108,110)((A(I,J),I=1,3),J=1,3)
110 FORMAT(15X,3(F15.6,2X)/)
DO 100 I=1,M
DO 100 J=1,N
U(I,J)=A(I,J)
100 CONTINUE
G=0.0
SCALE=0.0
ANORM=0.0
DO 300 I=1,I
L=I+1
RV1(I)=SCALE*C
G=0.0
S=0.0
SCALE=0.0
IF(I.GT.M)GO TO 210
DO 120 K=I,I
120 SCALE=SCALE+ABS(U(K,I))
IF(SCALE.EQ.0.0)GO TO 210
DO 130 K=I,I
U(K,I)=U(K,I)/SCALE
S=S+U(K,I)**2
130 CONTINUE
F=U(I,I)
G=-SIGN(SQRT(S),F)
H=F*G-S
U(I,I)=F-G
IF(I.EQ.N)GOTO 160
DO 150 J=L,I
S=0.0
DO 140 K=I,N
140 S=S+U(K,I)*U(K,J)
F=S/H
DO 150 K=I,I
U(K,J)=U(K,J)+F*U(K,I)
150 CONTINUE
160 DO 200 K=I,I
200 U(K,I)=SCALE*U(K,I)
210 W(I)=SCALE*C
G=0.0
S=0.0
SCALE=0.0
IF(I.GT.M.OF.I.EQ.N)GOTO 290
DO 220 K=L,I
220 SCALE=SCALE+ABS(U(I,K))
IF(SCALE.EQ.0.0)GOTO 290
DO 230 K=I,I
U(I,K)=U(I,K)/SCALE

```

```

230 S=S+U(I,K)**2
CONTINUE
F=U(I,L)
G=-SIGN(SQRT(S),F)
H=F*G-S
U(I,L)=F-G
DO240K=L,M
240 RV1(K)=U(I,K)/F
IF(I.EG.M)GOTO270
DO260 J=L,M
S=0.0
DO250 K=L,M
250 S=S+U(J,K)*L(I,K)
DO260K=L,M
U(J,K)=U(J,K)+S*RV1(K)
260 CONTINUE
270 DO280K=L,M
280 U(I,K)=SCALE*U(I,K)
290 ANORM=AMAX1(ANORM,ABS(W(I))+ABS(RV1(I)))
300 CONTINUE
IF(.NOT.MATV)GO TO 410
DO 400 II=1,N
I=N+1-II
IF(I.EG.M)GOTO390
IF(I.EG.0.0)GOTO360
DO 320 J=L,M
320 V(J,I)=(U(I,J)/U(I,L))/G
DO 350 J=L,M
S=0.0
DO 340 K=1,I
340 S=S+U(I,K)*V(K,J)
DO 350 K=1,I
V(K,J)=V(K,J)+S*V(K,I)
350 CONTINUE
260 DO 380 J=1,M
V(I,J)=0.0
V(J,I)=0.0
320 CONTINUE
390 V(I,I)=1.0
G=RV1(I)
L=I
400 CONTINUE
410 IF(.NOT.MATU)GO TO 510
M=M
IF(M.LT.N)M=M
DO 500 II=1,M
I=M+1-II
L=I+1
G=k(I)
IF(I.EG.M)GO TO 430
DO 420 J=L,M
420 U(I,I)=0.0
430 IF(G.EG.0.0)GO TO 475
IF(I.EG.M)GO TO 460
DO 450 J=L,M
S=0.0
DO 440 K=1,I
440 S=S+U(K,I)*L(K,J)
F=(S/U(I,I))/G
DO 450 K=I,M

```

```

U(K,J)=U(K,J)+F*U(K,I)
450 CONTINUE
460 DO 470 J=1,F
470 U(J,I)=U(J,I)/G
GO TO 490
475 DO 480 J=1,F
480 U(J,I)=0.0
490 U(I,I)=U(I,I)+1.0
500 CONTINUE
510 DO 700 KK=1,N
K1=N-KK
K=K1+1
ITS=0
520 DO 530 LL=1,K
L1=K-LL
L=L1+1
IF(ABS(RV1(L))+ANORM.EQ.ANCPM)GO TO 565
IF(ABS(Y(L1))+ANORM.EQ.ANCPM)GO TO 540
530 CONTINUE
540 C=0.0
S=1.0
DO 560 I=1,K
F=S*RV1(I)
RV1(I)=C*RV1(I)
IF(ABS(F)+ANCPM.EQ.ANORM)GO TO 565
G=W(I)
H=SQRT(F*F+C*C)
W(I)=F
C=C/H
S=-F/H
IF(.NOT.MATU)GO TO 560
DO 550 J=1,F
Y=U(J,L1)
Z=U(J,I)
U(J,L1)=Y*C+Z*S
L(J,I)=-Y*S+Z*C
550 CONTINUE
560 CONTINUE
565 Z=W(K)
IF(L.EQ.K)GO TO 650
IF(ITS.EQ.30)GO TO 1000
ITS=ITS+1
X=W(L)
Y=W(K1)
G=RV1(K1)
H=RV1(K)
F=((Y-Z)*(Y+Z)+(G-H)*(G+H))/(2.0*H*Y)
G=SQRT(F*F+1.0)
F=((X-Z)*(X+Z)+H*(Y/(F+SIGN(G,H))-H))/X
C=1.0
S=1.0
DO 600 I1=L,K1
I=I1+1
G=RV1(I)
Y=W(I)
H=S*G
G=C*G
Z=SQRT(F*F+H*H)
RV1(I1)=Z
C=F/Z

```

MADI..

```

S=H/7
F=X*C+C*S
G=-X*S+C*C
H=Y*S
Y=Y*C
IF(.NOT.MATV)GO TO 575
DO 570 J=1,3
X=V(J,1)
Z=V(J,1)
V(J,1)=X*C+Z*S
V(J,1)=-X*S+7*C
570 CONTINUE
575 Z=SQRT(F*F+H*H)
W(1)=Z
IF(Z.EQ.0.0)GO TO 580
C=F/7
S=H/7
580 F=C*G+S*Y
X=-S*C+C*Y
IF(.NOT.MATU)GO TO 600
DO 590 J=1,3
Y=U(J,1)
Z=U(J,1)
U(J,1)=Y*C+7*S
U(J,1)=-Y*S+7*C
590 CONTINUE
600 CONTINUE
RV1(L)=0.0
RV1(K)=F
W(K)=X
GO TO 520
650 IF(Z.EQ.0.0)GO TO 700
W(K)=-Z
IF(.NOT.MATV)GO TO 700
DO 690 J=1,3
690 V(J,K)=-V(J,K)
700 CONTINUE
WRITE(108,10)
10  FORMAT(5X,'LA MATRICE W EST:',/,5X,33(1F*))
WRITE(108,80)(W(J),J=1,3)
80  FORMAT(15X,3(F12.6,2X))
WRITE(108,20)
20  FORMAT(5X,'LA MATRICE U EST:',/,5X,33(1F*))
WRITE(108,90)((U(I,J),I=1,3),J=1,3)
90  FORMAT((15X,3(F12.6,2X))///)
WRITE(108,30)
30  FORMAT(5X,'LA MATRICE V EST:',/,5X,33(1F*))
WRITE(108,90)((V(I,J),I=1,3),J=1,3)
1000 IFRB=K
RETURN
END

```

```
SUBROUTINE PROD(A,P,C,M,N,IP)
DIMENSION A(M,N),B(N,IP),C(M,IP)
C(I,J)=0
DO 10 I=1,M
DO 10 J=1,N
DO 20 K=1,IP
20 C(I,J)=C(I,J)+A(I,K)*B(K,J)
10 CONTINUE
WRITE(108,102)
102 FORMAT(2X,'LE PRODUIT DE LA MATRICE A PAR LA MATRICE B EST:*')
WRITE(108,120)
120 FORMAT(2X,61(1F*))
WRITE(108,100)((C(I,J),I=1,M),J=1,IP)
100 FORMAT((15X,3(F15.6,2X))//)
RETURN
END
```

4.2 SIMULATION NUMERIQUE

A titre d'exemple, nous traitons un cas numérique simple représentant un système de la forme :

$$\dot{X} = AX + BU.$$

$$Y = CX$$

$$\text{avec } A = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} ; B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} ; C = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

En suivant scrupuleusement l'organigramme (3.13) relatif à la réduction d'un système, nous obtenons :

"APPROX":

- Résolution des équations de Lyapunov par Bingulac:

$$W_c = \begin{matrix} \text{(RES}\phi\text{L)} \\ \begin{bmatrix} 0.500 & 0.333 \\ 0.333 & 0.250 \end{bmatrix} \end{matrix} ; W_0 = \begin{bmatrix} 0.500 & -0.333 \\ -0.333 & 0.250 \end{bmatrix}$$

- Décomposition de Cholesky: $\text{(CH}\phi\text{LES)}$

$$W_c = L_c L_c^T ; L_c = \begin{bmatrix} 0.710 & 0.000 \\ 0.470 & 0.166 \end{bmatrix}$$

- ~~Donner~~ la matrice: $M = L_c^T W_0 L_c$. $\text{(PR}\phi\text{D)}$

$$L_c^T \cdot W_0 \cdot L_c = \begin{bmatrix} 0.087 & -0.018 \\ -0.018 & 0.006 \end{bmatrix}$$

- Valeurs propres et vecteurs propres de M. (DANILE)

$$\Sigma = \begin{bmatrix} 0.0256 & 0.0000 \\ 0.0000 & 0.0908 \end{bmatrix}$$

(-MRINV)

$$\Sigma^{-1} = \begin{bmatrix} 500 & 0.000 \\ 0.000 & 10.99 \end{bmatrix} \quad ; \quad \Sigma^{-1/2} = \begin{bmatrix} 22.360 & 0.000 \\ 0.000 & 3.315 \end{bmatrix}$$

$$U^T (L_c^T W_0 L_c) U = \Sigma^2$$

Vecteurs propres $\mu_1 = \begin{pmatrix} 0.208 \\ 0.978 \end{pmatrix}$; $\mu_2 = \begin{pmatrix} 0.978 \\ -0.908 \end{pmatrix}$

$$U = \begin{bmatrix} 0.208 & 0.978 \\ 0.978 & -0.908 \end{bmatrix}$$

- Matrice de transformation $T = L_c U \Sigma^{-1/2}$ (PRFD).

$$T = \begin{bmatrix} 3.302 & 2.301 \\ 5.684 & 1.042 \end{bmatrix} \quad ; \quad T^{-1} = \begin{bmatrix} -0.108 & 0.238 \\ 0.589 & -0.342 \end{bmatrix}$$

- Donner les matrices : (PRFD).

$$\hat{A} = T^{-1} A T = \begin{bmatrix} -3.302 & -2.301 \\ -11.369 & -2.084 \end{bmatrix}$$

$$\hat{B} = T^{-1} \cdot B = \begin{bmatrix} 0.1306 \\ 0.2470 \end{bmatrix}$$

$$\hat{C} = C T = \begin{bmatrix} 0.1306 & ; & 0.2470 \end{bmatrix}$$

d'où le système réduit :

$$(\hat{A}_1, \hat{B}_1, \hat{C}_1) = (-3.3020; 0.1306; -2.3827)$$

Interprétation :

Pour une illustration graphique, nous donnons la représentation du système : $\dot{x} = Ax + Bu$ (en trait plein)
 $y = Cx$

et la représentation graphique du système réduit :

$$\dot{x} = \hat{A}_1 x + \hat{B}_1 u \quad (\text{en pointillés})$$

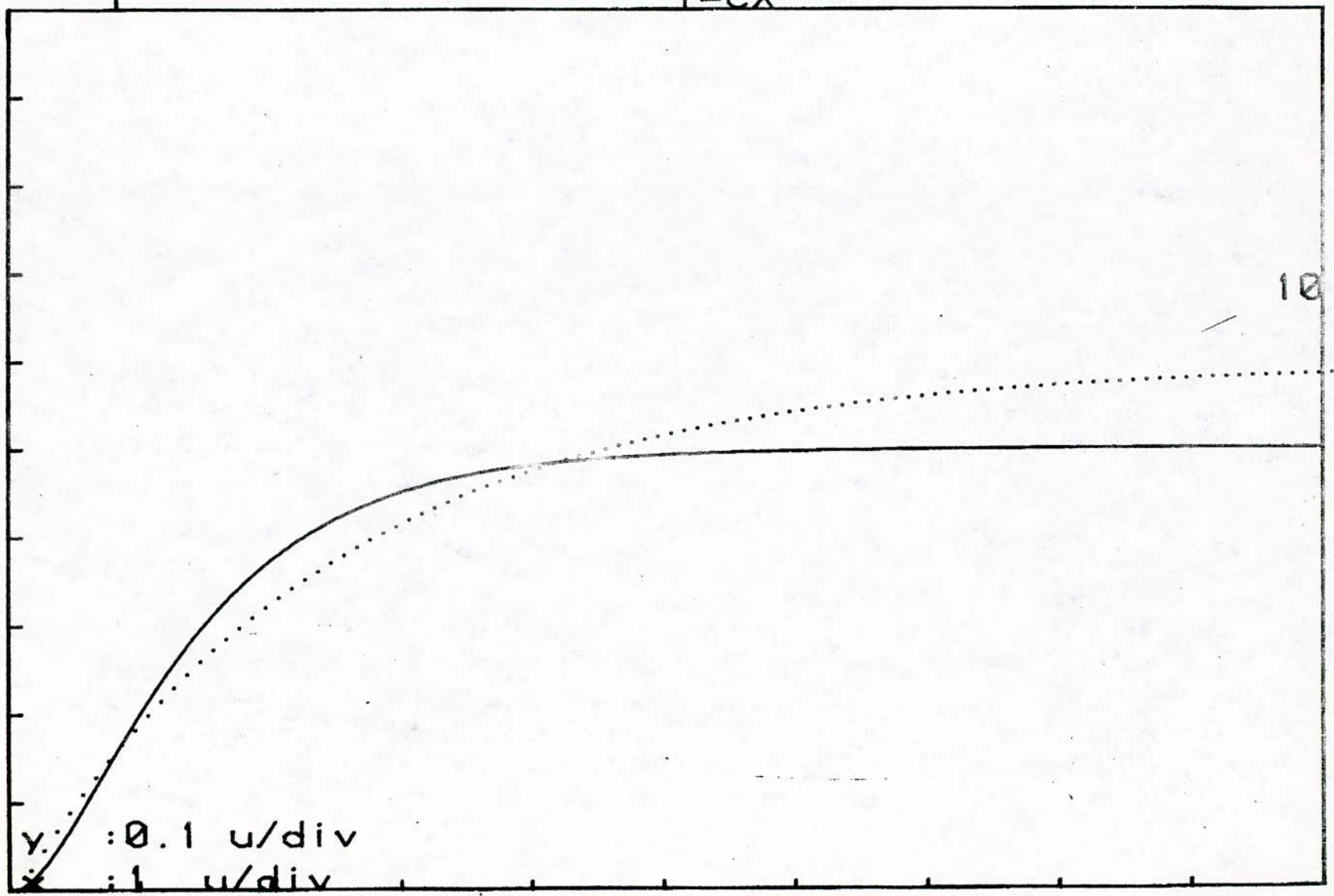
$$y = \hat{C}_1 x$$

Nous pouvons dire, eu égard, à ces deux courbes que l'approximation de (A, B, C) est bonne. Donc nous pouvons réduire l'ordre du système par la méthode équilibrée de l'ordre n (ici $n=2$) à l'ordre r (ici $r=1$)

Voir courbe (4.2.1)

4.2.1: REDUCTION DU SYSTEME $X' = AX + BU$

$Y = CX$



0

10

0

chapitre 5

CONCLUSIONS

Deux méthodes de réduction de systèmes basée sur une approche récente de décomposition en composantes principales ont été étudiées dans ce mémoire.

Cette approche consiste à ordonner des variables d'états par degré de commandabilité et d'observabilité. Les états les plus dominants correspondent à des composantes principales (T_i) de modules supérieurs. Le modèle réduit est composé de ces modes essentiels et donc retient les caractéristiques importantes du système original.

Cette nouvelle théorie a aussi montré de nombreuses applications dans les domaines de circuits électriques, commande en boucle fermée et sensibilité de systèmes.

Le modèle réduit peut être vu comme une généralisation des méthodes de modes dominants et perturbations singulières.

Il est bien ~~connu~~ connu que les valeurs propres d'une matrice possèdent des amplitudes limitées par les normes spectrales de la matrice c'est à dire ses amplitudes.

Il s'ensuit que les valeurs propres de la matrice dans le cas de Hankel sont limitées par l'unité. Ceci garantit la stabilité de la réalisation. C'est alors pourquoi la norme de la matrice est une robuste mesure aussi bien que les valeurs propres pour la stabilité dans un champ raisonnable, en tenant compte de la quantifications des erreurs.

Il apparaît donc que la réalisation intérieurement équilibrée offre un schéma numériquement favorable à la réalisation d'un système approximé.

ANNEXE

SUBROUTINE CHØLES

1. But : Décomposition triangulaire d'une matrice symétrique réelle (non nécessairement définie positive) par la méthode de CHØLESKY, puis inversion de cette décomposition.

2. Mode d'utilisation :

CALL CHØLES (A, N, B, EPS, DET, ILI, KØD).

A : Tableau monodimensionnel de taille $N * (N+1) / 2$ contenant en entrée, la partie triangulaire supérieure de la matrice donnée rangée par colonnes.
Ce tableau n'est pas détruit par l'exécution du programme si on ne veut que la décomposition triangulaire mais si on veut l'inversion, ce tableau A contiendra en sortie la partie triangulaire supérieure de l'inverse de la matrice de départ.

N : Ordre de la matrice.

B : Tableau monodimensionnel de taille $N * (N+1) / 2$ qui contient en sortie la matrice triangulaire supérieure rangée par colonnes et vérifiant :
 $A = \text{TRANS}(B) * B$ (ceci dans le cas où seule la décomposition est demandée); si on demande l'inversion, le tableau B contiendra en sortie l'inverse de la matrice citée précédemment, c'est à dire la matrice triangulaire supérieure vérifiant : $\text{INV}(A) = B * \text{TRANSP}(B)$.

EPS : Seuil au dessous duquel un pivot est considéré comme nul.

DET : Déterminant calculé de la matrice de départ.

ILI : Tableau entier de taille N indiquant la nature des lignes de la matrice B résultant de la décomposition de A :

$\text{ILI}(I) = 1$ signifie que la ligne I de B (ou la colonne I de son inverse) est réelle.

$\text{ILI}(I) = -1$ signifie que la ligne I de B (ou la colonne I de son inverse) est imaginaire pure.

.../...

$K\phi D$: Code indiquant en entrée si on veut ou non l'inversion:
 $K\phi D = 0$ la matrice est décomposée puis inversée.
 $K\phi D = 1$ la matrice est uniquement décomposée.

EN SORTIE : $K\phi D$ est un code d'erreur.

$K\phi D = 1$ un pivot est nul et la décomposition est impossible.

$K\phi D = 2$ N est inférieur à 2.

$K\phi D = 3$ le dernier élément diagonal de B est nul, dans ce cas la décomposition triangulaire est possible mais le déterminant de A étant nul, l'inversion est impossible.

Cette sous-routine peut être appelée en double précision en faisant :

CALL DCHPLES (A, N, B, EPS, DET, ILI, $K\phi D$)

SUBROUTINE RESØL

1. But : Résolution du système $AX = B$.

2. Mode d'utilisation :

CALL RESØL (A, B, N, KØD, EPS)

A : Tableau monodimensionné dans lequel on range en colonnes la matrice A. Celle-ci est détruite par l'exécution du programme.

B : Tableau de dimension N dans lequel sont rangés initialement les valeurs du second membre. Après l'exécution du programme, B contient la solution du problème.

N : Dimension du système (égal au nombre d'équations et au nombre de variables).

KØD : Code d'erreur

KØD = 0 solution normale

KØD = 1 système singulier.

EPS : Limite au dessous de laquelle un pivot est considéré comme nul. Cette subroutine peut être appelée en double précision en spécifiant :

CALL DRESØL (A, B, N, KØD, EPS)

SUBROUTINE DANILE

1. But: Calcul des valeurs propres d'une matrice réelle quelconque par la méthode de DANILEVSKI.
2. Mode d'utilisation:

CALL DANILE (A, N, TR, TI, CØF, VPR, VPI, KØD, EPS)

A : Tableau de taille $N \times N$ contenant la matrice donnée, rangée par colonnes (ce tableau est détruit).

N : Ordre de la matrice A.

TR et TI: Tableau de travail de taille N.

CØF : Tableau de taille $N+1$ contenant les coefficients du polynôme caractéristique de A (ce tableau est détruit).

VPR: Tableau de taille N contenant les parties réelles des valeurs propres de A classées par ordre croissant.

VPI: Tableau de taille N, contenant les parties imaginaires correspondantes.

KØD: code d'erreur.

KØD = 0 déroulement normal

KØD = 1 une valeur propre ne peut être obtenue après 100 itérations et deux valeurs de départ, différentes. le programme s'arrête.

KØD = 2N est inférieur à 2 ou supérieur à 40.

EPS : Seuil au dessous duquel un pivot est considéré comme nul.

Ce module utilise le sous-programme PORAB de la bibliothèque de mathématique et peut être aussi appelée en double précision en faisant.

CALL DDANIL (A, N, TR, TI, CØF, VPR, KØD, EPS)

SUBROUTINE MRINV

1. But : INVERSION D'UNE MATRICE
(calcul de l'inversion d'une matrice).

2. Mode d'utilisation :

CALL MRINV (A, B, N, KΦD, DET, EPS, IL, IC) .

A : Tableau dans lequel on range la matrice initiale en colonnes, la matrice A n'est pas détruite si A et B n'ont pas la même implantation en mémoire.
Si N est la dimension de la matrice, la dimension de ce tableau est $N \times N$.

B : Tableau contenant la matrice inverse de A. Si l'on ne désire pas garder la matrice initiale en mémoire, on peut écrire CALL MRINV (A, A, N, KΦD, DET, EPS, IL, IC) pour avoir le résultat rangé dans A.

N : Dimension de la matrice A et son inverse B.

DET : Valeur du déterminant de A.

KΦD : Code d'erreur.

KΦD = 0 Solution normale.

KΦD = 1 Matrice singulière.

EPS : Seuil au dessous duquel un pivot est considéré comme nul.

IL : Tableau de travail de dimension N à réserver dans le programme appelant.

IC : Tableau de travail de dimension N à réserver dans le programme appelant.

Cette subroutine peut être appelée en double précision en faisant :

CALL DMRINV (A, B, N, KΦD, DET, EPS, IL, IC)

BIBLIOGRAPHIE

- [1] B.C MOORE : "Principal component analysis in linear systems: controlability, observability and model reduction".
IEEE TRANS - AUTOMATIC - CONTROL - Vol- AC-26 -
pp. 17-32. Feb 1981.
- [2] M. BETTAYEB : "Approximation of linear systems: New Approaches based on singular value decomposition."
"University of Southern California".
- [3] S.P. BINGULAC : Dept. of Elec. Engrg. (pp 135 - 137. IEEE Trans. Aut.)
Institute of Nuclear Sciences "B Kidrič" and
University of Belgrade - Belgrade, Yougoslavia
"AN ALTERNATE Approach to expanding $PA + AP = -Q$
Computer Science and Applied Mathematics.
- [4] — CHOLESKY :
- [5] E.J. DAVISON : "A Method for simplifying linear dynamic systems"
"IEEE TRANS - Automatic - Contr. Vol. AC-11
pp. 93-101, Jan. 1966.
- [6] M.R CHIDAMBARA. "Further comments on "A Method for simplifying linear dynamic systems".
IEEE Trans. Automat. Contr. (Corresp) Vol. AC-12
pp 79. Dec 1967.
E.J. Davison "Reply by E.J. Davison"
IEEE TRANS - Automat. Contr (Corresp) Vol. AC-12
pp. 799 - Dec 1967.
- [7] M. Hutton and B. Friedland: "Routh Approximation for reducing order of linear, time invariant systems".
IEEE (Trans. Automat. Contr. AC - 20 (3)
pp 329-337 - 1975.
- [8] Routh - Hurwitz : "Routh Stability criterion".
number of right - and left half plane roots; Computation using Routh-Hurwitz.
T. AC 74 Jun 250-254 (2B12)

[9] G. Langholz & D. Feinmesser : " Model reduction by Routh Approximations"
Int. J. Systems Sci., 9 (5)
pp 493-496 (1978).

[10] S. Kung : " A New low-order approximation algorithm via singular value decompositions,
Submitted to IEEE Trans. Autom. Contr.
Department of electrical engineering university
of Southern California, Los Angeles, CA. 9007

[11] V.M. ADAM. JAN. D.F. Arov, and M.G. Krein : " Analytic properties
of schmit parts for a Hankel operator and
the generalised Shur-Takari - problem
math. USSR.

[12] L. PERNEBO. and L. SILVERMAN
" Model reduction via balanced state space
representations, presented at 1979 CDC in
Florida.

" Balanced systems and model reduction"
in Proc.

[13] M. Dreyfus : " FORTRAN IV

[14] George E. Forsythe - Michael Malcolm - Cleve B. Moler
Computer Methods for MATHEMATICAL
computations pp 229-235.