الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

8/90

وزارة التعليم العالي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : Genie Electrique Option : automatique

PROJET DE FIN D'ETUDES

SUJET

Elaboration d'un Logiciel pour l'Analyse et la synthèse des réglages dans l'Espace d'Etat

Proposé par :
Boucherit. M.S
Chekireb. H

Etudié par :
Kendjouh. L.
Zadjine. M.

Dirigé par :

PROMOTION : Juin 1990.

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : GENIE ELECTRIQUE

PROJET DE FIN D'ETUDES

SUJET

ELABORATION D'UN LOGICIEL POUR L'ANALYSE ET LA SYNTHESE DES REGLAGES DANS L'ESPACE D'ETAT

Proposé par : Boucherit M^ed S
Chekireb Hachemi

Etudié par : Kendjouh Lemnouar
Tadjine Mohamed

Dirigé par :

PROMOTION : 1990

بسم الله الرحمن الرحيم

الإهداء

إلى الذين أدركوا حقيقة قوله تعالى: "وَقَالَ الَّذِينَ أُوتُوا الْعِلْمَ وَيْلَكُمْ ثَوَابُ اللَّهِ خَيْرٌ لِمَنْ آمَنَ وَعَمِلَ صَالِحًا"

"قُلْ هَلْ يَسْتَوِي الَّذِينَ يَعْلَمُونَ وَالَّذِينَ لَا يَعْلَمُونَ"

إلى كل من علمنا حرفا وسقانا أدبا وأروانا علما ..

إلى العيون التي ترقب الزمن لتستكشف حاضرها وماضيها ومستقبلها ...

نهدي عملنا المتواضع هذا

REMERCIMENT

Nos remerciment vont à toutes les personnes ayant contribué de prés ou de loin à la réalisation de ce travail.

Nous remercions plus particulierement :

M^r CHEKIREB HACHEMI

M^r BOUCHERIT Med SEGHIR

en tant que promoteurs, pour tous les conseils et l'aide qu'ils n'ont cessé de nous apporter tout le long de ce projet.

Que M^r SETIHI OMAR trouve ici l'expression de nos profonds sentiments, pour l'aide technique dont il nous a fait beneficier.

TABLE DES MATIERES

II-C COMMANDABILITE ET OBSERVABILITE

INTRODUCTION

A-GÉNÉRALITÉS

L'Automatique peut être définie comme la science et les téchniques de l'automatisation des systémes.
On entend par systéme ,une collection d'objets liés entre eux, que l'on isolera artificiellement du réste du monde.

La commande d'un systéme, éxige au préalable d'établir un modéle mathémathique entrées-sorties, qui approche au mieux la réalité physique qui nous intéresse.

Pour préciser cette modélisation, deux niveaux de classification apparaissent immédiatement [4]:

a: - de la nature même des équations mathémathiques, on peut distingué déja deux grandes classes (déterministe - stochastique). Nous nous intéressons uniquement à la classe déterministe, cette derniére se subdivise en sous classes et le travail proposé concerne (fig a) finalement la sous classe linéaires à coéfficients constants.

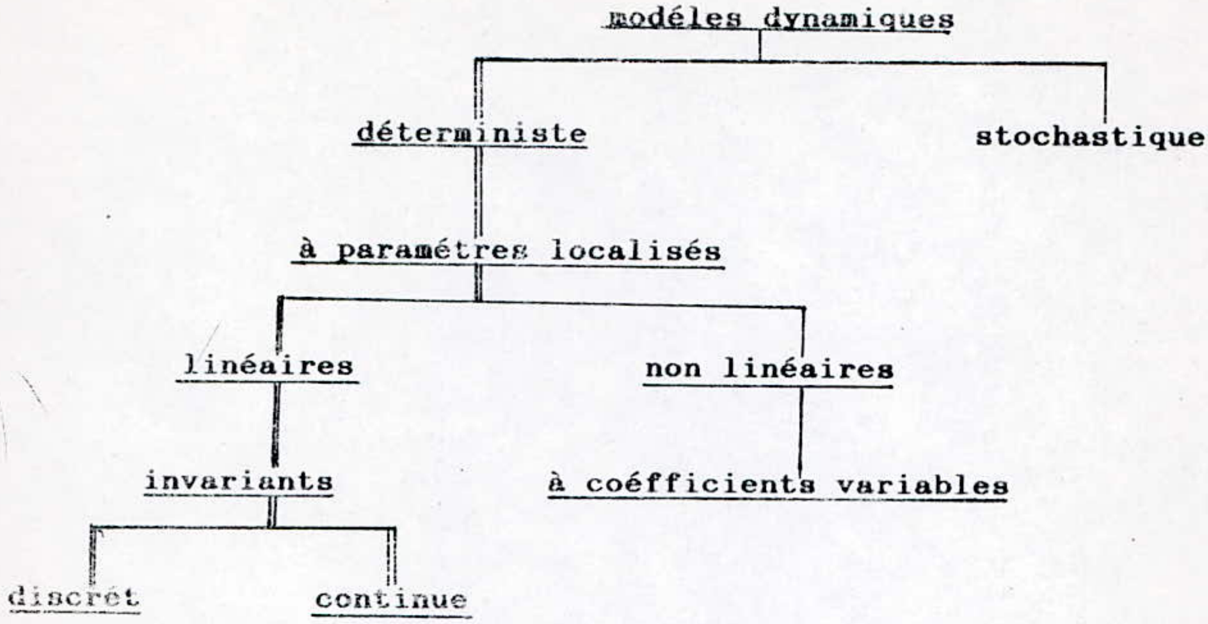


fig a: classification selon la nature des équations mathémathiques(en trait double la branche qui fera l'objet de l'étude)

b: -la nature intrinséque du modéle étant définie, le probléme de la représentation mathémathique se pose (fig b).
La richésse de la représentation retenue conditionne naturéllement la synthése futur des lois de commande.

Nous traitons uniquement le modéle dans le formalisme d'état.

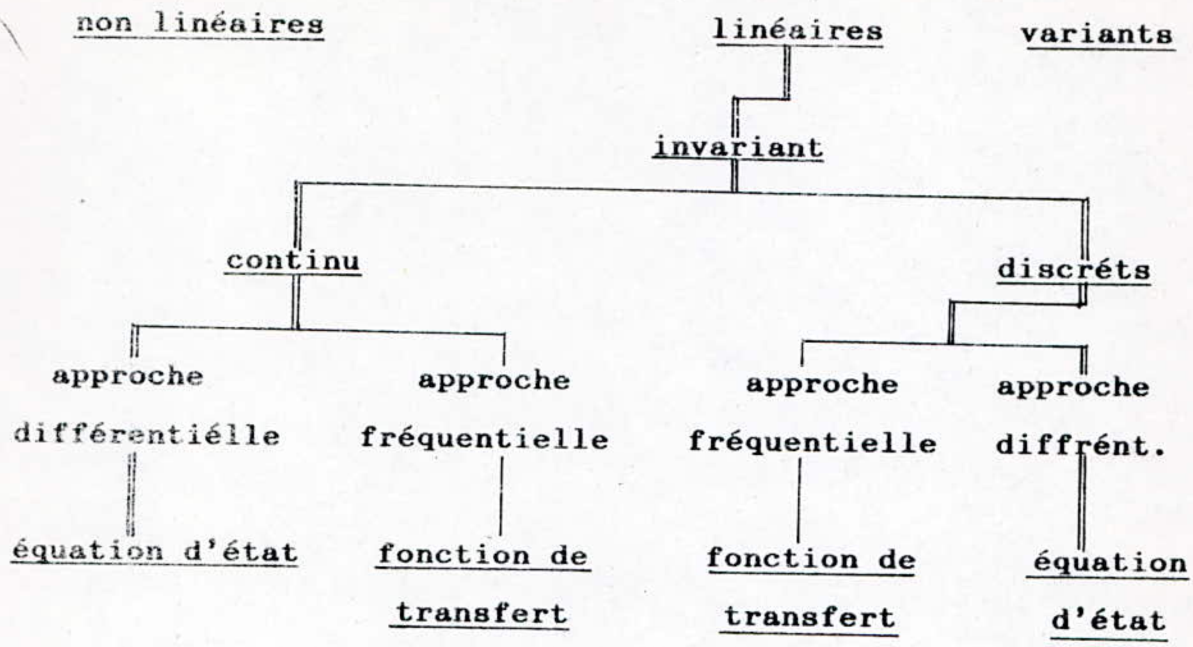


fig b:représentation des systémes linéaires invariants (en trait double les deux branches qui font l'objet de l'étude)

IL est connue que le traitement des systémes dans l'éspace d'état, que ce soit au niveau 'analyse' ou 'synthése' fait appel éssentiellement au calcul matriciel.
Les algorithmes proposés dans la littérature à ce propos, sont souvent à caractére numérique.
Lors de la synthése de la commande dans le formalisme d'état, on est amené le plus souvent à traiter une importante quantité d'information, ce qui justifie le recours à un ou plusieurs calculateurs numériques .

Le travail proprement dit, consiste à concevoir et réaliser un logiciel, qui traitera (dans un premier pas) de l'analyse et de la synthése des systémes de commande dans l'éspace d'état.

B-CONCÉPT D'ÉTAT:

Les états d'un systéme etaient définie par KALMAN [6], de la façon suivantes :

ETAT: l'état d'un systéme, est une structure mathémathique contenant un ensemble de n variables $X_o(t)$......$X_n(t)$, les valeurs initiales $X_i(t_o)$ de cet ensemble et les entrées $U_j(t)$ du systéme ,suffisent à décrire les futur réponses du systéme pour $t>t_o$.

Pour une représentation correcte, un ensemble minimum de variables d'état est nécessaire.

Les entrées $U_1(t)$....$U_r(t)$, sont détérministe, ie elles prénnent des valeurs spécifique pour chaque valeur de $t>t_o$.

généralement, la quantité t_o est prise égale à zéro.

Les variables d'état ne sont pas necéssairement, physiquement tous mésurables, elles peuvent étre des quantités purement mathémathique.

consequences:

vecteur d'état: l'ensemble des variables d'état constituent les composantes du vecteur dit d'état $X(t)$ de dimension n

$$X(t)=\begin{vmatrix} X_1(t) \\ \vdots \\ X_n(t) \end{vmatrix}$$

Lorsque toutes les entrées $U_j(t)$ d'un systéme donné sont spécifier pour t>to, le vecteur d'état résultant détermine le comportement dynamique de ce systéme.

éspace d'état: l'Espace d'état est définie comme l'éspace à n dimension dans lequel les composantes du vecteur d'état représentent les axes de coordonnées.

équations d'état: Les équations d'états d'un systéme, sont un ensemble de n équations différentielles (récurentes pour les systémes discréts) du premier ordre, ou n est le nombre d'état indépendant.

La forme générale de ces équations est:

-cas continue:

$$\left.\begin{array}{l} X = A\,X + B\,U \\ Y = C\,X + D\,U \end{array}\right\} \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \;(1)$$

-cas discrét :

$$\left.\begin{array}{l} X[k+1] = F\,X[k] + H\,U[k] \\ Y[k] = C\,X[k] + D\,U[k] \end{array}\right\} \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\;(2)$$

avec:

F (ou A)= matrice d'état (n × n)

H (ou B)= matrice d'entrée (n × r)

C = matice de sortie (m × n)

D = matrice de couplage (m × r)

U = vecteur d'entrée de dimension r

Y = vecteur de sortie de dimension m

C-ORGANISATION DU TRAVAIL:

Ce travail est divisé en quatre parties :

- Dans la premiére partie appelée conception, nous parlerons des objectifs visés par la conception du logiciel, la méthodologie de conception, ainsi que du choix d'implantation.

-La deuxieme partie qui concérnera la réalisation traitera les différents algorithmes d'analyse et de synthése à elaborer et l'organisation global du logiciel.

-La troisiéme partie sera consacré à l'exploitation du logiciel nous parlerons ainsi de la communication entre l'utilisateur la machine, en dernier lieu on présentera quelque exemples d'applications.

-En fin nous donnerons quelque conclusions générales sur le logiciel et les pérspéctives éventuel de son éxtention.

1ere

partie

I-1 OBJECTIF DU LOGICIEL:

Le logiciel qu'on se propose de réaliser doit avoir pour objectifs:

-Le traitement, l'analyse et la synthése des systémes de commande que ce soit pour le cas continue ou échantillonnés, monovariables ou multivariables utilisant l'approche vectoriel d'état.

-IL devra ainsi constitué un fond d'aide:

- à la concéption de la commande et la régulation des systémes
- pédagogique, pour les travaux pratiques de traitement des systémes de commande.

-IL doit étre conversationnel, afin de pérmettre aux utilisateurs initiés ou non à l'automatique, de résoudre leur problémes de commande.

-IL doit offrir un grand choix de méthodes de traitement tout en réstant facile à utiliser.

-IL doit étre un produit éxploitable par de nombreux utilisateurs ne possédant pas de gros calculateurs.

I-2 MÉTHODOLOGIE DE CONCÉEPTION :

La conception de logiciel constitue un processus complexe et divérsifié, il est nécéssaire de l'aborder dans les meilleurs conditions, au niveau des moyens mais également au niveau de la méthodologie.

Le nombre de choix des méthodes que doit proposer ce logiciel impose :

-L'implantation de plusieurs algorithmes de traitement ,avec des niveaux de complexité différents allant du complexe au plus simple.

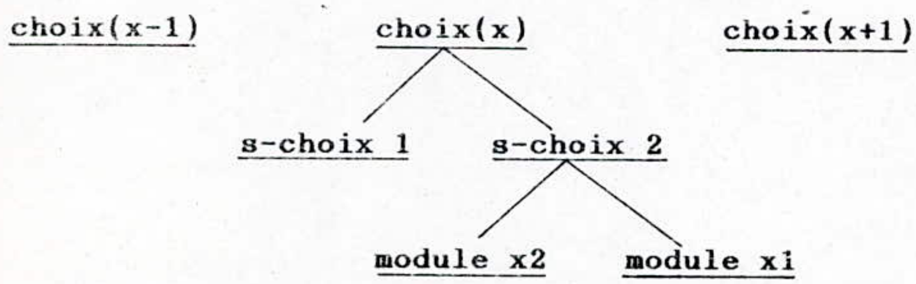
-L'implantation de programmes assurant la gestion des dépandances éxistantes entre ces algorithmes.

-La conception d'une intérface faisant le lien entre l'utilisateur et les différents traitement offérts par ce logiciel.

Le probléme étant ainsi posé, la solution que nous avons choisi pour le résoudre est la suivante:

Analyser le probléme à différents niveaux d'abstraction concrétisés par des étapes.D'une étape à l'autre on précise les élements de la solution, ainsi au fur et à mésure des étapes, le niveau d'abstraction diminue, mais les détails de réalisation augmentent.

Cette solution peut étre représentée par le schema suivant



Chaque degré de profondeur correspond à un degré de raffinement suplémentaire, plus la déscription est détaillée plus les opérations à éffectuer sont réparties dans des modules nombreux et distinct.
En fait, dans cette méthodologie,l'aspét le plus important est l'organisation hiérarchique d'un programme, cette organisation est obtenue par l'emploi de procédures.
Les procédures de plus haut niveau ont pour vocation de dégager les taches les plus éssentielles, ces procédures peuvent appeler des procédures de plus bas niveau dans la hierarchie,et ainsi de suite jusqu'à atteindre des procédures que l'on qualifie de primitives, les procédures de bas niveau fixent les détails de mise en oeuvre.
Cette méthodologie de conception s'appelle 'raffinement graduel' (ou approche descendantes) [7]....[12]
par conséquent l'implantation du logiciel doit satisfaire les régles de structuration et modularisation dérivant de cette méthodologie.

I-3 CHOIX DIMPLANTATION:

a-choix du systéme:

Nous avons choisi délibérement le systéme IBM PC/AT equipé d'une carte graphique EGA-VGA, en raison de:

-La parfaite documentation éxistantes pour ce systéme tand au niveau du matériel que du systéme d'éxploitation.

-L'implantation trés importantes de ce produit sur le marché.

par conséquent:

le logiciel proposé fonctionne dans les conditions suivantes:

-Micro-ordinateur IBM PC/XT/AT ou compatible 100% .

-Systéme d'éxploitation MS-DOS.

-Carte graphique IBM EGA-VGA.

b-choix du langage:

IL convient de relever que la méthodologie de la programmation est indépendante du langage, mais que celui ci peut en favoriser l'emploi.

La méthodologie étant défini (approche déscendante) elle trouve un support approprié avec le langage Turbo-Pascal et ceci pour sept (7) raison:

-Le Turbo-Pascal est connu pour sa haute structuration.

-Le Turbo-Pascal permét une vitésse de compilation et d'execution hors du commun (plus de 32000 lignes/minute pour les versions 5).

-Le Turbo-Pascal propose le concept d'unité (unit) qui permét d'augmenter les possibilités d'une bonne structuration des programmes et donne ainsi la possibilité de décomposer le travail en equipe, sans oublier l'établissement de bibléothéques

-Le Turbo-Pascal donne la possibilité,de génerer des programmes du type .EXE, théoriquement infini.

-L'éditeur de lien (LINKER) trés performant, pérmet une assez importante optimisation pour le code et la mémoire.

-Le gestionnaire de projet sophéstiqué (MAKE) pérmet de gérer les dépendances éventuels du programme vis-a-vis des unités qu'il utilise, selon un processus optimal.

-Le Turbo-Pascal pérmet la programmation systéme (manipulation d'objets habituellement traités en langage d'assemblage).

****

2eme

**

partie

****

II-A DISCRÉTISATION D'UNE ÉQUATION D'ÉTAT

1-INTRODUCTION:

Avec l'apparition des ordinateurs on fait de plus en plus appel à des méthodes discrétes de réglage des processus industriels.

Les ordinateurs remplaçent ainsi les classiques régulateurs fonctionnant de maniere continue.

La discrétisation consiste à passer d'une équation d'état continue à une équation d'état discréte.

2-ECHANTILLONNAGE:

on montre que [1],[2] le systéme discrét correspondant au systéme continue (1) est donner par:

$X(k+1)=F\ X(k) + H\ U(k)$

$Y(k) = C\ X(k) + D\ U(k)$

avec $F=\exp(A)$

$$H=\int_0^T \exp(A\ (t-\theta))\ B\ d\theta$$

T=periode d'échantillonnage.

3-ALGORITHME:

-pour la matrice F:

$$F = F + T^k A^k / K!$$

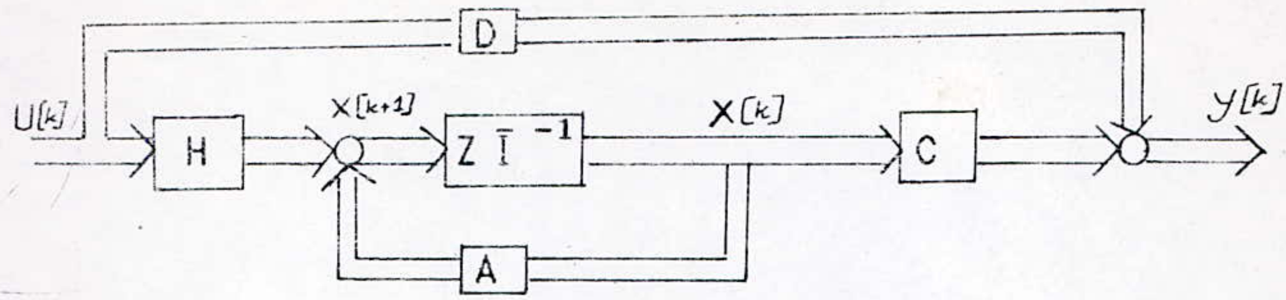
-pour la matrice H:

-si le systéme est avec élement de maintient:

$$H = H + T^{k+1} A^k / (k+1)! \quad , H = H\ B$$

-si le systéme est sans élement de maintient :

$$H = F\ B$$



STRUCTURE D'UN SYSTEME ECHANTILLONNEE DANS L'ESPACE D'ETAT.

II-B:REPRESENTATIONS:

1-INTRODUCTION:

Dans ce chapitre nous allons abordé les relations éxistantes entre la représentation par matrice de fonction de transfert (M-F-T) et la représentation d'état (R-E).

2-PASSAGE M-F-T ---> R-E :

On s'est limiter au cas des pôles reéls simple (bien qu'il apparait à premiere vue que se n'est qu'un cas particulier mais en réalité c'est un cas trés intéréssant) .

L'algorithme se base sur la méthode de GILBERT [2] qui consiste à calculer les différents matrices A,B,C,D à partir de la matrice de fonction de transfert en décomposant celle ci sous la forme:

$$G(P) = \sum_{i=1}^{q} M_i \frac{1}{P - P_i} + D$$

q:nombre de pôles de G(P)

$$D: \lim_{|P| \to \infty} (G(P))$$

Les matrices B et C sont déterminées à partir des matrices M_i tandis que la matrice A est donneé par :

$$A = \mathrm{diag}(\underbrace{P_1, \ldots P_1}_{n_1 \text{ fois}}, \underbrace{P_2 \ldots P_2}_{n_2 \text{ fois}}, \ldots, \underbrace{P_q \ldots P_q}_{n_q \text{ fois}})$$

n_i : est le rang de la matrice M_i $i=1,n$.

3-PASSAGE R-E ---> M-F-T:

C'est par la résolution des équations d'état par la transformation en Z (P dans le cas continue) qu'on obtient la fonction de transfert matricielle donnant la relation entre le vecteur de sortie Y et le vecteur d'entrée U.
En effet un simple calcul donne:

-dans le cas discrét :

$$G(Z) = C\,(ZI - F)^{-1} H + D$$

- dans le cas continue:

$$G(P) = C\,(PI - A)^{-1} H + D$$

La méthode de calcul de la M-F-T est basé sur l'algorithme de LEVERIER [1] qui la donne sous la forme d'un polynome matricielle:

$$G(Z) = \sum_{i=0}^{n} Q_i\, Z^i \frac{1}{P(Z)}$$

P(Z)= déterminant de F

Q_i = matrices de coéfficients calculer à partir des matrices B,C et les coéfficients de P(Z).

II-C: COMMANDABILITE ET OBSERVABILITE:

1- INTRODUCTION

Les variables d'états sont des variables interne, de la il se pose le probléme de savoir si on peut imposer leur évolution ou si on peut retrouver (ou reconstituer) ces variables.

On touche ici des notions trés importantes qui sont la commandabilité et l'observabilité.

2-COMMANDABILITE :

-GLOBALE: Une condition necéssaire et suffisante pour qu'un systéme (A,B) soit commandable globalement est que le rang de la matrice V définie par:

$$V = [\ B,\ A\ B\ ,\ A^2\ B\ ,\ \ldots\ldots,\ A^{n-1}\ B\ ]$$

soit égal à n (n étant l'ordre du systéme).

donc l'algorithme consiste à:

- . calculer la matrice V
- . déterminer son rang par test de dépendances linéaires (voir annexe 2)

-PARTIELLE: On entend par commandabilité partielle, la commandabilité par rapport à chacunes des entrées.

Dans ce cas on montre aussi [1]..[3] qu'une condition nécéssaire et suffisante pour que le systéme (A,B) soit commandable par rapport à 1 entrée U_i est que:

$$\text{rang}\ (\ b_i\ , A\ b_i\ ,\ \ldots\ldots,\ A^{n-1}\ B\ ) = n$$

n étant l'ordre du systéme

b_i : colonne i de la matrice B correspondant à l'entrée U_i

3-OBSERVABILITE

-GLOBALE: De la même maniere que pour la commandabilité une condition necéssaire et suffisante pour qu'un systéme (A,C) soit observable globalement est que le rang de la matrice U definie par:

$$U = \begin{vmatrix} C \\ C\,A \\ C\,A^2 \\ \vdots \\ C\,A^{n-1} \end{vmatrix}$$

soit égal à n. On applique donc le méme algorithme donné pour la commandabilité.

-PARTIELLE: Dans ce cas en entend par obsrevabilité partielle, l'observabilité par rapport à chacunes des sorties. par dualité on peut énnoncer la condition necéssaire et suffisante à l'observabilité par rapport à la sortie Y_i :

$$U = \begin{vmatrix} C_i \\ C_i\,A \\ C_i\,A^2 \\ \vdots \\ C_i\,A^{n-1} \end{vmatrix}$$

$$\mathrm{rang}(U) = n$$

4-COMMANDABILITE EN SORTIE:

La notion de commandabilité traiter précédement est réstrictive puisque elle se rapporte au vecteur d'état seuleument.

Une autre notion de commandabilité consiste à éxiger que les sorties soient commandables par rapport aux entrées [1] .

La condition necéssaire et suffisante pour qu'un systéme soit commandable en sortie est que :

$$\text{rang}(V) = \text{rang} (D , C H , C F H , ..., C F^{n-1} H) = m$$

m:nombre de sorties.

L'algorithme consiste donc à:

-calculer la matrice V

-déterminer son rang par la procédure de test de dépendances linéaires.

II-D FORMES CANONIQUES :

1-INTRODUCTION:

IL est parfois interéssant de méttre un systéme sous une forme canonique de tel sorte que ces proprietés soient misent en évidence.

On peut méttre un systéme sous trois formes canoniques differentes: forme canonique commandable, forme canonique observable et la forme canonique de Jordan.

Dans ce travail on ne traitera que les deux premiéres formes.

2-FORMES CANONIQUES COMMANDABLE:

a-Systéme Monovariable:

On essaie de transformer l'équation d'état (1) à l'équation d'état:

$$\bar{X}(k+1) = \bar{F}\,\bar{X}(k) + \bar{H}\,U(k)$$
$$\bar{Y}(k) = \bar{C}\,\bar{X}(k)$$

ou $\bar{F}, \bar{H}$ sont données par:

$$\bar{F} = \begin{vmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 \dots & 0 \\ 0 & 0 & 0 & 1 \dots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ -a_0 & \dots & \dots & \dots & -a_{n-1} \end{vmatrix} \quad , \quad \bar{H} = \begin{vmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{vmatrix}$$

les reéls a_i représentent les coéfficients du polynome caractéréstique.

L'algorithme consiste à détérminer la matrice de transformation T par :

$$T_n = H$$

$$T_{n-i} = F\,T_{n-i+1} + a_{n-i}\,T_n \quad \dots\dots\dots \; i = 1, n-1$$

puis calculer:

$$\bar{F} = T^{-1}\,F\,T$$
$$\bar{H} = T^{-1}\,H \quad \text{et} \quad \bar{C} = C\,T$$

b-Systéme Multivariable:

Pour ce cas on parle de décomposition en P ($\leq$ nombre d'entrées) sous systémes ou de décomposition en M (égal au nombre d'entrées) sous systémes.

-Décomposition en P sous systéme:

Dans ce cas on cherche la matrice de transformation T perméttant de méttre les matrices F,H sous la forme:

$$\bar{F} = \left|\begin{array}{cccc:cccc} 0 & 0 & \ldots 0 & * & & & & * \\ & & & * & & \underline{0} & & * \\ & \underline{I} & & * & & & & * \\ & & & * & & & & * \\ \hline & & & & 0 & 0 & \ldots 0 & * \\ & \underline{0} & & & & \underline{I} & & * \\ & & & & & & & * \\ & & & & & & & * \end{array}\right| , \quad \bar{H} = \left|\begin{array}{cc:ccc} 1 & 0 & * & \ldots & * \\ 0 & 0 & . & & . \\ 0 & 0 & . & & . \\ 0 & 0 & * & \ldots & * \\ \hline 0 & 1 & * & \ldots & * \\ 0 & 0 & . & & . \\ 0 & 0 & . & & . \\ 0 & 0 & * & \ldots & * \end{array}\right|$$

décomposition en P sous systéme (P=2)

L'algorithme de détermination de T s'appuie sur le plan de sélection qui est présentée dans [3] qui donne la matrice T sous la forme:

$$T = [\, b_1 , Ab_1 , \ldots . A^{\mu 1-1} b_1 , b_2 , \ldots . A^{\mu 2-1} b_2 , \ldots , b_q , \ldots A^{\mu 3-1} b_q]$$

q étant le nombre d'entrées necéssaire pour avoir n colonne indépendantes

- Décomposition en M sous systéme:

On cherche la matrice T qui pérmet de méttre le systéme sous la forme:

$$\bar{F} = \left[\begin{array}{cccc|cccc} 0 & 0 & \ldots 0 & * & & & & * \\ & & & * & & & & * \\ & \underline{I} & & * & & \underline{0} & & * \\ & & & * & & & & * \\ \hline & & & * & 0 & . & ..0 & * \\ & & & * & & \underline{I} & & * \\ & \underline{0} & & * & & & & * \\ & & & * & & & & * \end{array}\right] , \quad \bar{H} = \left[\begin{array}{cc} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \hline 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right]$$

Décomposition en M sous systémes (M = 2)

L'algorithme de détermination de T s'appuie sur le deuxieme plan de sélection présenté dans [2]. IL donne la matrice T sous la forme:

$$T = [\, b_1, Ab_1, \ldots. A^{v1-1} b_1, b_2, \ldots. A^{v2-1} b_2, \ldots, b_m, \ldots A^{vm-1} b_m]$$

m : étant le nombre d'entrées du systéme.

v_i : sont appelées Invariants de kronecker.

3-FORMES CANONIQUES OBSERVABLE:

a-Systéme Monovariable:

On cherche la matrice T qui met le systéme sous la forme

$$F = \left[\begin{array}{cccc} 0 & 0 & \ldots\ldots 0 & -a_0 \\ & & & -a_1 \\ & \underline{I} & & . \\ & & & . \\ & & & . \\ & & & -a_{n-1} \end{array}\right] , \qquad C = [0 \ldots. 0\ 1]$$

les reéls a_i étant les coéfficients de l'équation caractéréstique.

On détermine la matrice de transformation T par l'algorithme suivant:

$$T_n = C$$

$$T_{n-i} = T_{n-i+1} F + a_{n-i} T_n \quad .... \quad i=1,n-1$$

dans ce cas on'a:

$$\bar{F} = T F T^{-1}, \quad \bar{H} = T H , \quad \bar{C} = C T^{-1}$$

b-Systéme Multivariable:

De la même maniére que pour la forme canonique commandable, on parle de décomposition en P sous systéme et de décomposition en r sous systéme (égal au nombre de sorties).
On peut obtenir l'une ou l'autre forme, suivant le mode de sélection des lignes dans la matrice d'obsrvabilité U .

4-REMARQUES:

IL est évident que pour passer à une forme canonique commandable ou obsérvable il faut que le systéme soit commandable ou(et) obsevable.

Donc il faut avant tout tester la commandabilité et l'observabilité du systéme.

II-E :REPONSES INDICIELLE ET ANALYSE DE LA STABILITE

1-INTRODUCTION:

La réaction d'un systéme à un saut unité est donnée par la réponse indicielle.
Dans ce paragraphe nous donnerons une méthode numérique de calcul de la réponse indicielle pour le cas échantillonné.

Nous aborderons ensuite l'analyse de la stabilité dans l'éspace d'état pour un systéme donné, notament les conditions de stabilité.

2-REPONSE INDICIELLE:

Notons tout d'abord que le saut unité pour le cas multivariable, est vectoriél càd chaque entrée du systéme est éxitée simultanément par un saut unité.
La solution de l'équation récurrente (2) au instants d'échantillonnage [1] est donnée par

$$X[k] = F^{k} X[0] + \sum_{i=1}^{k} F^{i-1} H U[k-i]$$

tandis que la solution de l'équation de sortie est donnée par:

$$Y[k] = C F^{k} X[0] + \sum_{i=1}^{k} C F^{i-1} H U[k-i] + D U[k]$$

L'algorithme est donné par :

$$\begin{aligned} X[k+1] &= F X[k] + H U[k] \\ Y[k] &= C X[k] + D U[k] \end{aligned} \qquad k = 0..k1$$

3-ANALYSE DE LA STABILITE:

L'analyse de la stabilité d'un systéme donné constitue um facteur extrêment important dans la synthése des systémes,elle est l'une des problémes fondamentaux de la commande.
En effet toutes les commandes reliant les mesures Y aux entrées U doivent assurer la stabilité globale en boucle fermé.
Pour tirer des conclusions sur la stabilité d'un systéme donné on doit l'éxiter par ses entrées pendant un laps de temps, puis le laisser à son régime propre, autrement dit annuler toutes les intérventions extérne sur le systéme.

a-Stabilité des Systémes Continues:[1],[4]

La représentation d'état (1) et sa solution genérale perméttent de définir la stabilité.

Un systéme linéaire est stable si et seulement si l'éffet des conditions initiales disparait lorsque le temps augmente indéfiniment.

Une condition nécessaire et suffisante (C-N-S) de stabilité asymptotique est que les valeurs propres de la matrice A soient à partie reéls négative.

b-Stabilité des Systémes Discrét [1]:

Une C-N-S pour la stabilité d'un systéme linéaire discrét est que les valeurs propres de la matice F soient toutes en module inférieur à l'unité.

c-Algorithme:

L'algorithme se base essentiellement sur l' élaboration de l'équation caractéristique par la méthode de LEVERIER [1] et de la, déterminer ses racines à l'aide d'un algorithme numérique approprié de résolution d'équations polynomiales, ensuite les analyser à la lumiére des conditions de stabilité citées plus haut.

II-F RÉGLAGE PAR CONTRE RÉACTION D'ÉTAT:

1-INTRODUCTION :

Etant donné le modéle mathémathique déterministe à paramétres connus d'un systéme S. Le probléme fondamental de la commande consiste à rechercher les lois de commande reliant les mesures Y aux entrées U qui assurent:

- la stabilité globale en boucle fermé
- les contraintes de pérformances éxigeés

Dans ce chapitre nous allons aborder de la synthése des circuits de réglage dans l'éspace d'état.

2-STRUCTURE OPTIMALE:

Cette structure qui ne concerne que les systémes discrét décrit par (2), est obtenue par la minimisation d'un critére intégrale [1] en utilisant le principe d'optimisation de BELLMAN. L'annulation de l'écart de régalge en régime établie éxige l'introduction d'un régulateur intégrateur en cascade avec le systéme (fig 1).
On montre alors[1] que la commande optimale U_o est donnée par:

$$U_o[k] = -K\ X[k] + K_w\ W[k] - K_v\ V[k]$$

$$\text{avec} \quad K = [\ K_S \quad -K_R\ ] \qquad X[k] = \begin{vmatrix} X_S[k] \\ \\ X_R[k] \end{vmatrix}$$

K_w : matrice d'intérvention dirécte de la consigne W[k]

K_v : matrice d'intérvention dirécte de la perturbation V[k]

-Pour les systémes continue la structure retenue est représentée sur la figure -fig 2- c'est une structure classique assurant l'assignement des pôles en boucle fermée.

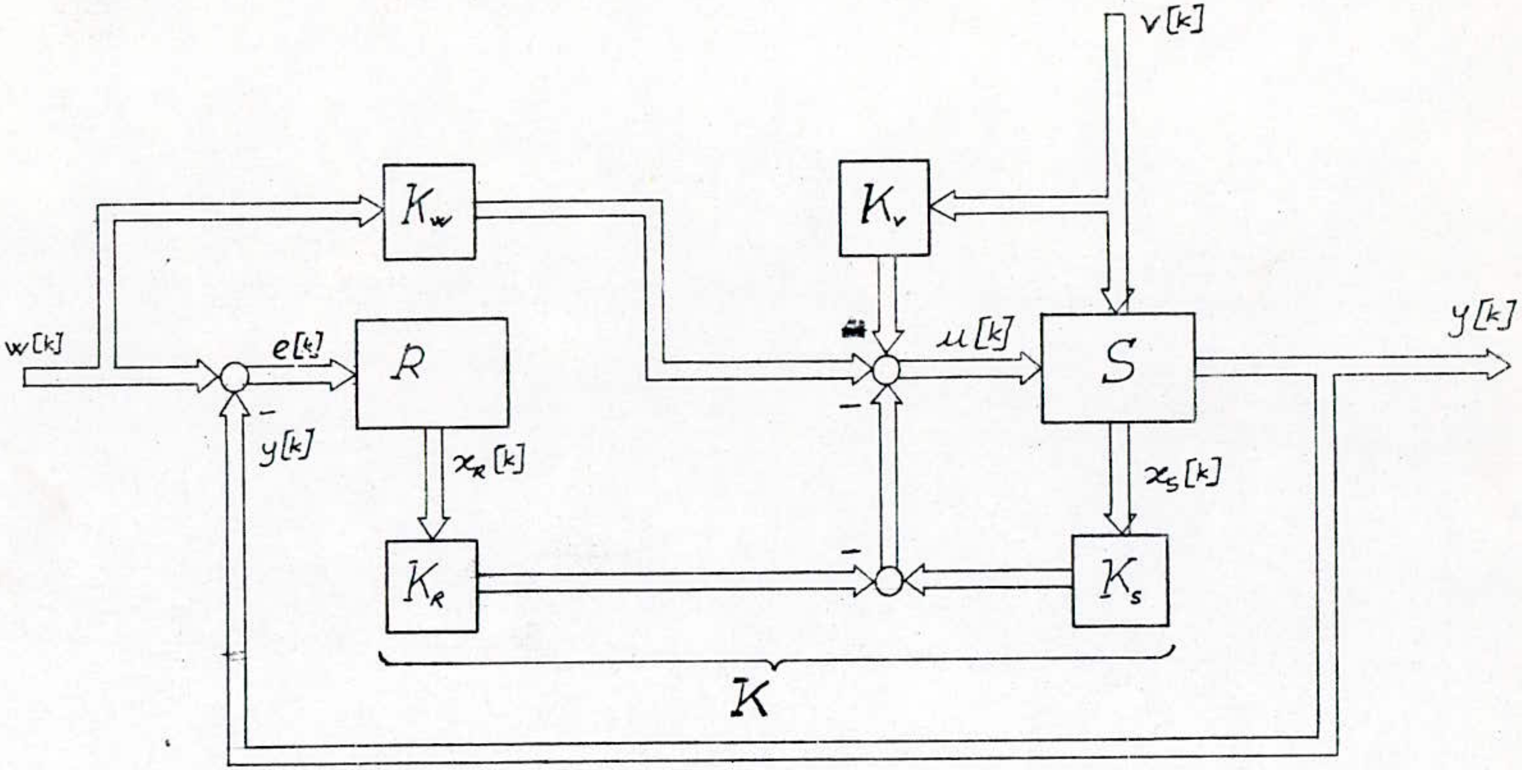


Fig. 1 Représentation schématique de la structure optimale du réglage d'état d'un système multivariable

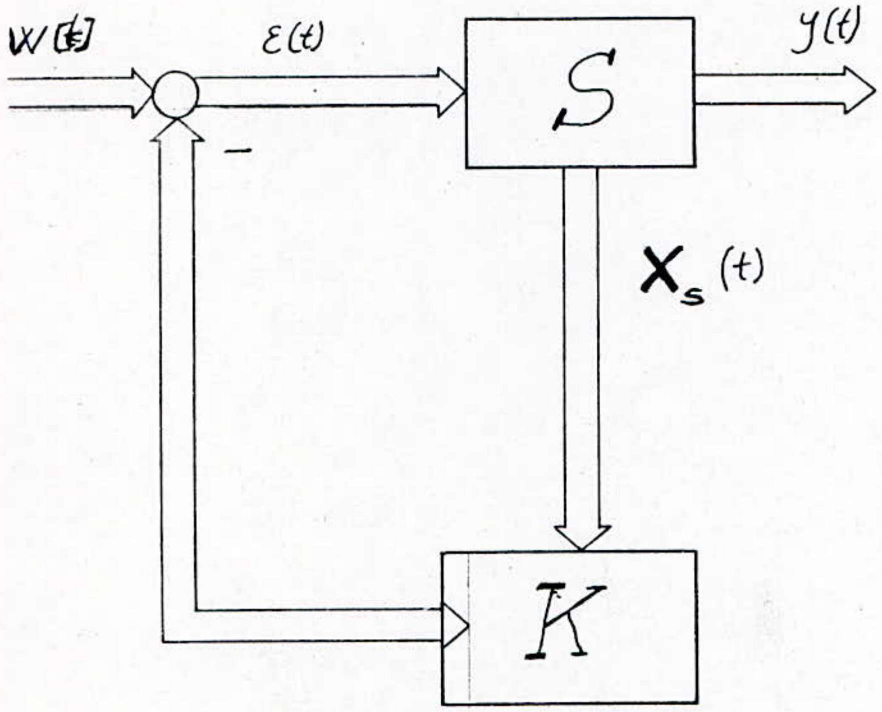


Fig. 2 Structure de réglage d'état pour les système continus

3-REACION D'ETAT POUR LES SYSTEMES MONOVARIABLES:

Dans ce cas la contre réaction d'état K^T est un vecteur, pour la détermination de ses composantes on fait appel à la forme canonique de réglage.

L'algorithme consiste à:

- calcul de la matrice de transformation T qui pérmet le passage à la forme canonique de réglage (voir paragraphe formes canoniques) ;
- calcul des coéfficients $\bar{\alpha}_i$ du polynome caractéréstique désiré ;
- calcul de $\bar{K}^T = [\, \alpha_n - \bar{\alpha}_n ,\ \alpha_{n-1} - \bar{\alpha}_{n-1} , \ldots\ldots , \alpha_1 - \bar{\alpha}_1 \,]$

  les α_i étant les coéfficients du polynome caractéréstique

- calcul du retour d'état cherché, K^T par :

$$\bar{K}^T = K^T \; T^{-1}$$

Pour la détermination de K_w et K_v, on éxige que l'influence du régulateur en régime établie soit nulle.On montre alors[1] que:

$$K_w = \frac{1}{C_s^T \; (1 - F_s + h_s K_s^T)^{-1} h_s}$$

$$K_v = K_w \; C_s \,(1 - F_s + h_s K_s^T)^{-1} h_{sv}$$

h_{sv} : vecteur liant la perturbation au systéme à régler

4-REACTION D'ETAT POUR LES SYSTEMES MULTIVARIABLE:

Dans ce paragraphe nous présenterons trois algorithmes connus pour la réaction d'état multivariable qui aboutissent à une matrice de retour à rang variable en fonction de l'algorithme utilisé.

a-Retour d'Etat de la Forme qK^T :

La caractéristique essentielle de cet algorithme est qu'il donne une matrice de réaction K dont les lignes sont linéairement dépendantes, le rang de K est donc unitaire (=1). L'algorithme consiste en fait, à convertir le probléme à m entrées en un probléme à une entrée et de la, lui appliquer la méthode précedente (monovariable) à l'aide des étapes suivantes:

- tester la commadabilié du systéme par rapport à chacune de ses entrées.
- déterminer le vecteur de transformation q de dimension m par:

  *si le systéme est commandable par une entrée U_i alors on pose : q[i] = 1

  q[j] = 0 j=1..m et j<>i

  * si non (le systéme n'est commandable par aucunes des entrées) on pose: q[i] = 1 i=1..m
- calculer le vecteur d'entrée monovariable associé à la matrice H du systéme multivariable

$$h = H q$$

- calcul de k^T par l'algorithme du monovariable
- poser $K = q\, k^T$

b-Retour d'Etat par Décomposition en P sous Systémes:

Cet algorithme trouve son appuie sur la décomposition du systéme initiale en P sous systémes.

Le résultat de cette décomposition conduit à une forme

canonique caractéristique de réglage [2].
L'algorithmes consiste à:

- initialisation de $\bar{K}^T$ à $\underline{0}$
- assignement des pôles désirés à chaque sous systéme i de la décomposition (cela est possible par l'intérmédiare de la ligne i dans la matrice $\bar{K}^T$)
- calculer la matrice de retour d'état du systéme initial par:

$$K = \bar{K}^T \; T^{-1}$$

T étant la matrice de transformation.

c-Retour d'Etat par Décomposition en M sous Systémes:

La premiére chose à constater sur les deux algorithmes précedent est que les lignes des matrices de réaction sont:

* pour q K^T, linéairement dépendantes (rang unitaire)
* pour la décomposition en P-S-S, pas forcément toutes indépendantes .

On montre d'autre part que la contribution d'un régulateur avec rang unitaire au systéme se traduit par un mauvais rejét des perutrbations quelles que soient les positions des pôles du systéme en boucle fermé.
Pour remédier à cela et proposer à l'utilisateur un choix suffisant de méthodes de réglage, nous avons introduit un algorithme se basant essentiéllement sur la décomposition du systéme globale en sous systémes égal au nombre d'entrées.
La caractéréstique importante de cet algorithme est qu'il donne une matrice de réaction à rang maximal càd :

rang(K) = nombre d'entrées

L'algorithme consiste à décomposer le systéme globale fermé, réaction d'état inclu, en m sous systémes

$$X[k+1] = F_G X[k]$$

$$\text{avec } F_G = F-H\ K$$

$$F_G = \left|\begin{array}{c:c:c} F_1 & \underline{0} & \underline{0} \\ \hdashline \underline{0} & F_2 & \underline{0} \\ \hdashline \vdots & \ddots & \vdots \\ \hdashline \underline{0} & \underline{0} & F_m \end{array}\right| \quad , F_i = \left|\begin{array}{cccc} 0 & & & \\ 0 & & & \\ \vdots & & \underline{I} & \\ 0 & & & \\ * & * & * & * \end{array}\right|$$

Les détails théoriques et étapes à suivre afin de méttre en oeuvre cet algorithme, étant longs et complexes nous renvoyons le lecteur à l'ouvrage de BUHLLER[1] ou il trouvera un exposé important la dessu.
Vu l'importance de cet algorithme, nous avons jugér utile de donner le programme corresspendant en annexe 3.
Les matrices K_w et K_v sont calculer de la même façon que pour le cas monovariable (pour les systémes échantillonnées).

5-DECOUPLAGE PAR RETOUR D'ETAT :

Dans ce paragraphe, on étudiera le découplage d'un systéme multivariable à l'aide de la contre réaction d'état . Chaque grandeur de consigne agit alors uniquement sur la propre grandeur à régler, au moins aux instants d'échantillonnage, la matrice de transfert sous trouve ainsi sous forme diagonale.
Dans ce qui suit on adoptera deux structures différentes, pour les systémes continues et pour les systémes discréts.

a-Systémes Continue:

La structure adopter pour ce cas est représenté sur la figure -fig 3-.

La matrice de retour K est donnée par $K = -E^{-1} F$

E et F sont déterminées par l'algorithme suivant:

* détermination de la matrice E:

$$E_i \text{ (} i \text{ eme ligne de E) } = C_i A^{d_i} B$$

d_i =plus petit entier tel que E_i <>0

* calcul de la matrice H par:

$$H = E^{-1}$$

* calcul de la matrice F définie par:

$$F = \begin{vmatrix} C_1 \; A^{d1+1} \\ C_2 \; A^{d2+1} \\ \vdots \\ C_m \; A^{dm+1} \end{vmatrix}$$

b-Systéme Discrét:

La structure adopter est représenté sur la -fig 4-

avec :K_u matrice d'entrée de découplage;

K_s matrice de découplage des états;

K_v , K_R matrices diagonaux;

L'algorithme suivi se base sur la suite de pondération, qui donne la réaction d'un systéme multivariable à des impulsions unités [1]. On obtient pour le systéme à régler avec contre réaction d'état :

$$G[k] = C_s (F_s - H_s K_s)^{k-1} H_s K_u \qquad k >= 1$$

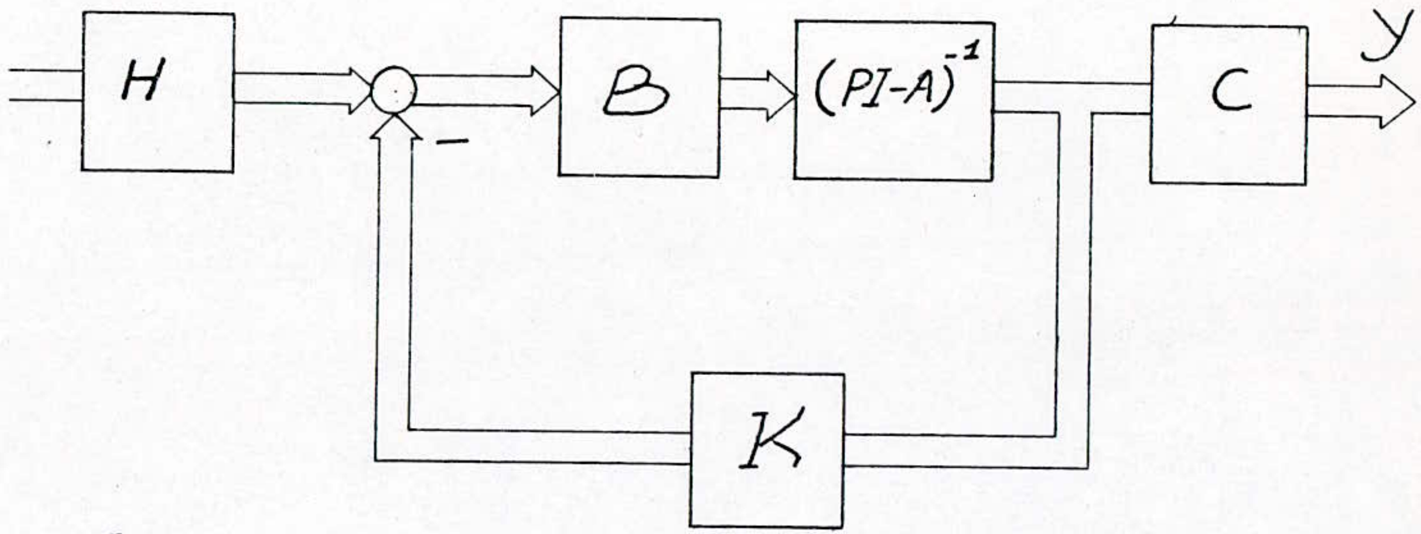


Fig. 3 Structure de réglage d'état avec découplage pour un système continu.

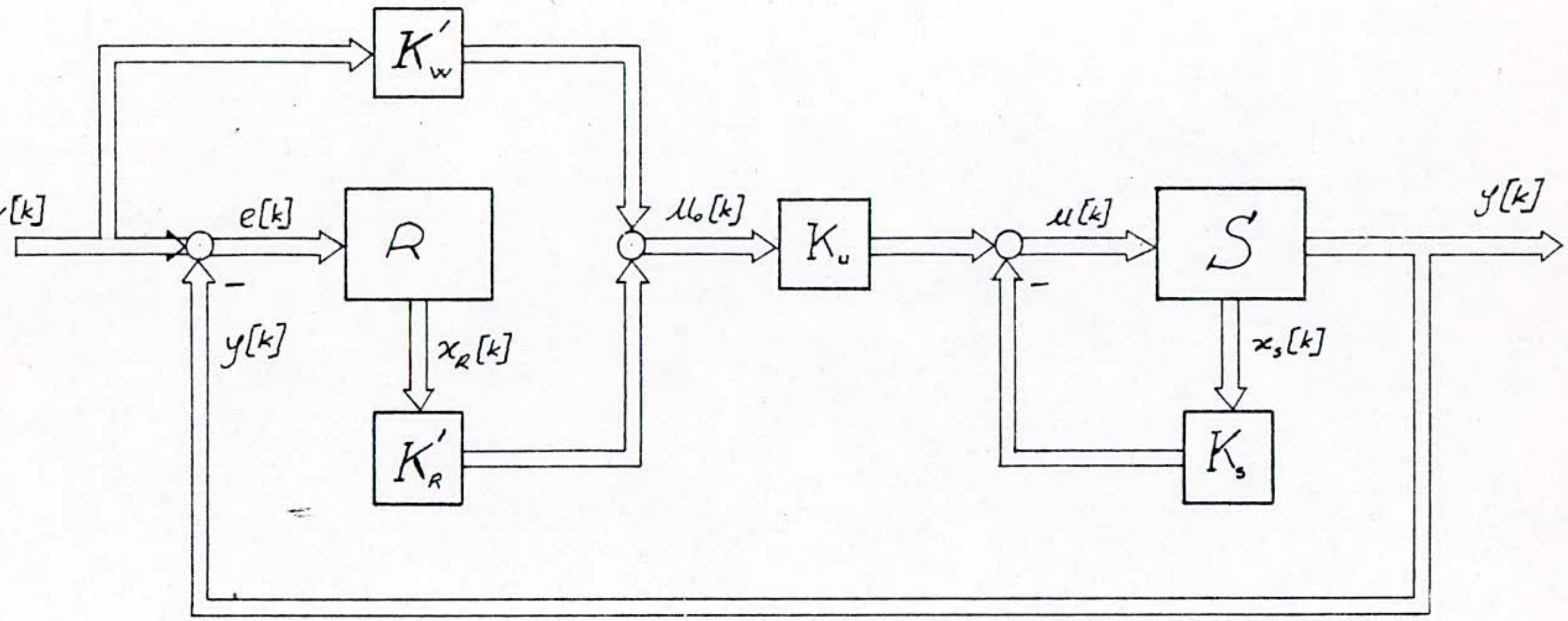


Fig. 4 Représentation schématique de la structure du réglage d'état avec découplage d'un système multivariable

Le découplage est assurer quand G[k] est diagonale.

* en imposant $G[1] = I \ (m \times m)$ on tire

$$K_u = (C_s \ H_s)^{-1}$$

* en imposant $G[2] = \Lambda$ (matrice diagonale $m \times m$)

on trouve:

$$K_s = (C_s \ H_s)^{-1} (C_s \ F_s - \Lambda \ C_s)$$

* on décompose alors le systéme en sous systémes du premier ordre et à chaque sous systéme on superpose un régulateur intégrateur [1];
* on calcul K_v par compensation d'un pôle de chaque sous systéme.

6-RESOLUTION DE L'EQUATION DE RICCATI DISCRETE:

L'équation de RICCATI donne la commande optimale sur la base de la minimisation d'un critére quadratique du type:

$$\bar{J} = \sum_{i=0} (U^T[i] \ R \ U[i] + E_i^T \ Q \ E_i)$$

L'algorithme suivant [3] donne la solution asymptotique et non pas la solution en temps réel (dans le cas de l'horizon infini) de l'équation :

$$K[i] = A^T \ K[i+1] \ A - A^T \ K[i+1] \ B \ (R + B^T \ K[i+1] \ B)^{-1} \times$$
$$B^T \ K[i+1] \ A + C^T \ Q \ C \quad \ldots\ldots\ldots\ldots\ldots(*)$$

La commande optimale est alors définie par :

$$\bar{U} = -L\, X[i]$$

$$L = (R + B^T K B)^{-1} B^T K A$$

K étant la solution de l'équation (*).

II-6 OBSÉRVATEUR D'ÉTAT :

1-INTRODUCTION :

La synthése da la contre réaction d'état traité au chapitre précédent, assume que toutes les grandeurs d'état du systéme à régler, sont mesurables ou peuvent étre génerer à partir des sorties du systéme.

Cependant dans la plupart des cas pratiques, il n'est pas possible de mesurer directement ces grandeurs, on doit alors employer un observateur d'état.

Un observateur consiste en un algorithme qui permet de déterminer la valeur des grandeurs d'états aux instants d'échantillonnage -fig 5-.

Dans ce chapitre nous présenterons la structure de l'observateur et ses différents constituants pour le cas monovariable, puis pour le cas multivariable, l'environnement étant déterministe.

Pour un environnement stochastique, nous présenterons le filtre de KALMAN.

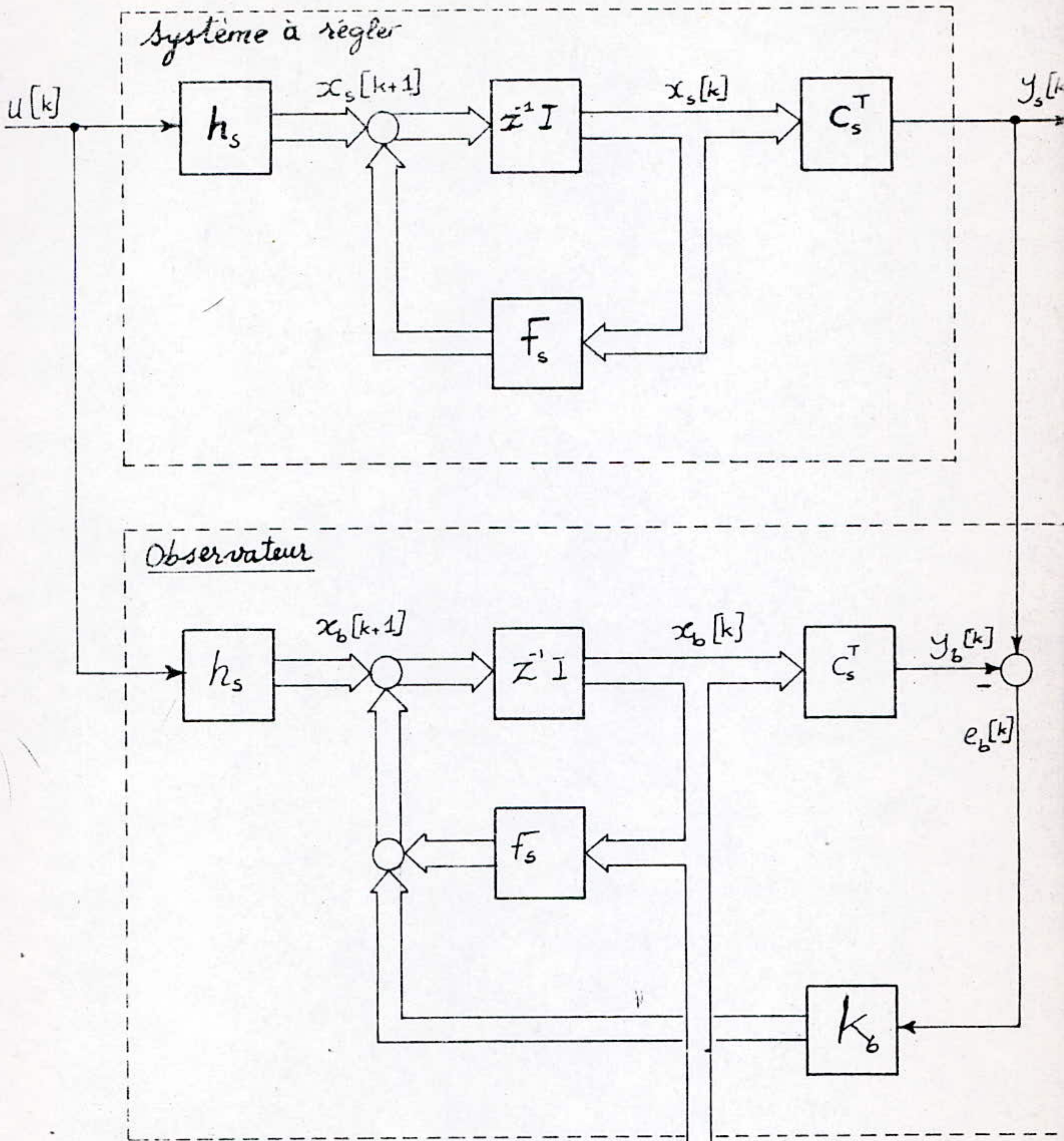


Fig. 5.

Diagramme structurel du système à régler $x_b[k]$ et de son obser

2-OBSERVATEUR POUR UN SYSTEME MONOVARIABLE:

Dans cette section trois cas seront traiter:

a- Perutrbations Nulles :

L'équation dynamique de l'observateur est donnée par :

$$X_b[k+1] = F_b\ X_b[k] + H_s\ U[k] + K_b\ Y_s[k]$$
$$F_b = F_s - K_b\ C_s$$

On rappelle que F_s, H_s, C_s sont données par le systéme à régler. Le vecteur de contre réaction de l'observateur sera déterminer de telle maniére que l'érreur d'observation

$$\bar{X}_b[k] = X_s[k] - X_b[k]$$

tend asymptotiquement vers zero, on montre [1],[3] que l'équation dynamique de l'erreur est donnée par la relation:

$$\bar{X}_b[k+1] = F_b\ \bar{X}_b[k]$$

Pour imposer une certaine dynamique à l'erreur il suffit d'assigner les valeurs propres de la matrice F_b

L'algorithme consiste donc à:

* passer à la forme canonique observable (voir chap:formes canoniques)
* calcul du vecteur K_b par l'algorithme de placement des poles traité dans le cas monovariable du chapitre précédent.

Remarque:

cet algorithme est valable pour les systémes discrét et continue b-Perurbations Constantes:

Lorsque des perurbations interviennent sur le systéme, le comportement de ce dernier est décrit par :

$$X_s[k+1] = F_s\ X_s[k] + H_s\ U[k] + H_{sv}\ V[k]$$

$$Y[k] = C_s\ X_s[k]$$

L'équation dynamique de l'erreur d'observation devient:

$$\bar{X}_b[k+1] = F_b\ \bar{X}_b + H_{sv}\ V[k]$$

On voit que l'erreur en régime établie ne s'annule pas, par conséquent il n'est pas possible d'observer correctement le vecteur d'état du systéme à régler, d'ou la necéssité de compléter l'observateur d'état, on aboutit alors à la structure représentée à la -fig 6-

On montre [1] que l'intégrateur introduit dans cette structure pérmet d'annuler l'écart d'observation en régime établie.

L'équation dynamique pour l'observateur globale est donnée par:

$$X_{bG}[k+1] = F_{bG}\ X_{bG}[k] + H_{bG}\ V[k]$$

$$\text{avec } X_{bG}[k] = \begin{vmatrix} \bar{X}_b[k] \\ V_b[k] \end{vmatrix} \quad , H_{bG} = \begin{vmatrix} H_{sv} \\ 0 \end{vmatrix}$$

$$F_{bG} = \begin{vmatrix} F_s - K_b\ C_s & -H_{sv} \\ 0 & 1 \end{vmatrix}$$

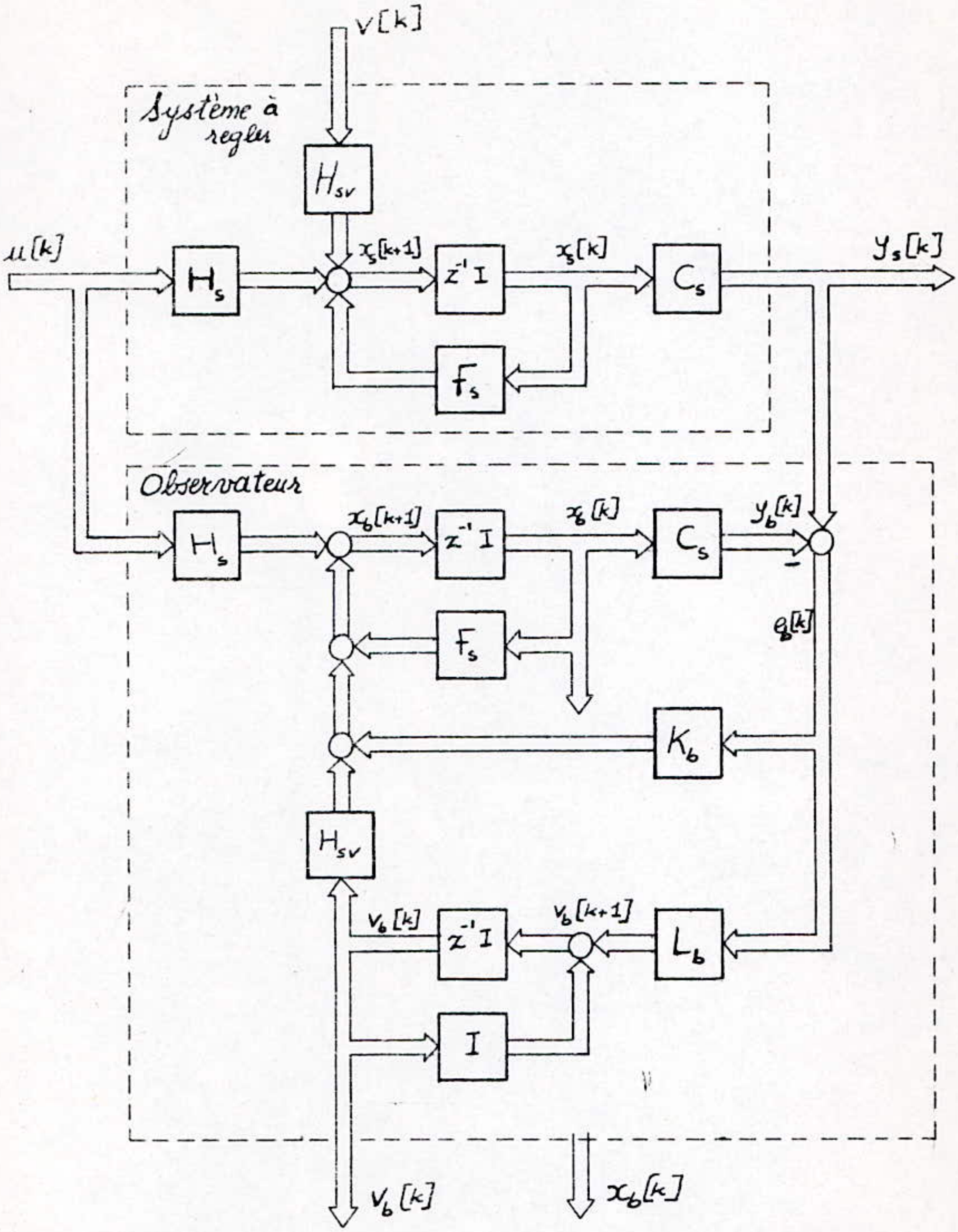


Fig 6. Diagramme structurel de l'observateur d'état et de perturbation

L'algorithme consiste à calculer les matrices F_{bo}, C_{bo} définie par:

$$F_{bo} = \begin{vmatrix} F_s & -H_{sv} \\ 0 & 1 \end{vmatrix} \quad , \quad C_{bo} = [\ C_s \quad 0\ ]$$

puis déterminer le vecteur K_{bo} par l'algorithme précedent et poser

$$K_{bo} = \begin{vmatrix} K_b \\ -L_b \end{vmatrix}$$

, K_b pour l'obsérvateur d'état
L_b pour la pérturbation

c-Perturbations Variables:

L'observateur d'état et de perturbation étudie précédement, ne pérmet plus une observation correcte de grandeur de perurbation losque celle ci varie rapidement et de maniére déterministe .

Dans un tel cas, on doit étendre la structure de l'observateur en tenant compte du modéle d'état de la pérturbation, on aboutit alors à la structure de la -fig 7-.

On montre pour un tel cas [1] que l'équation dynamique de l'erreur est donnée par:

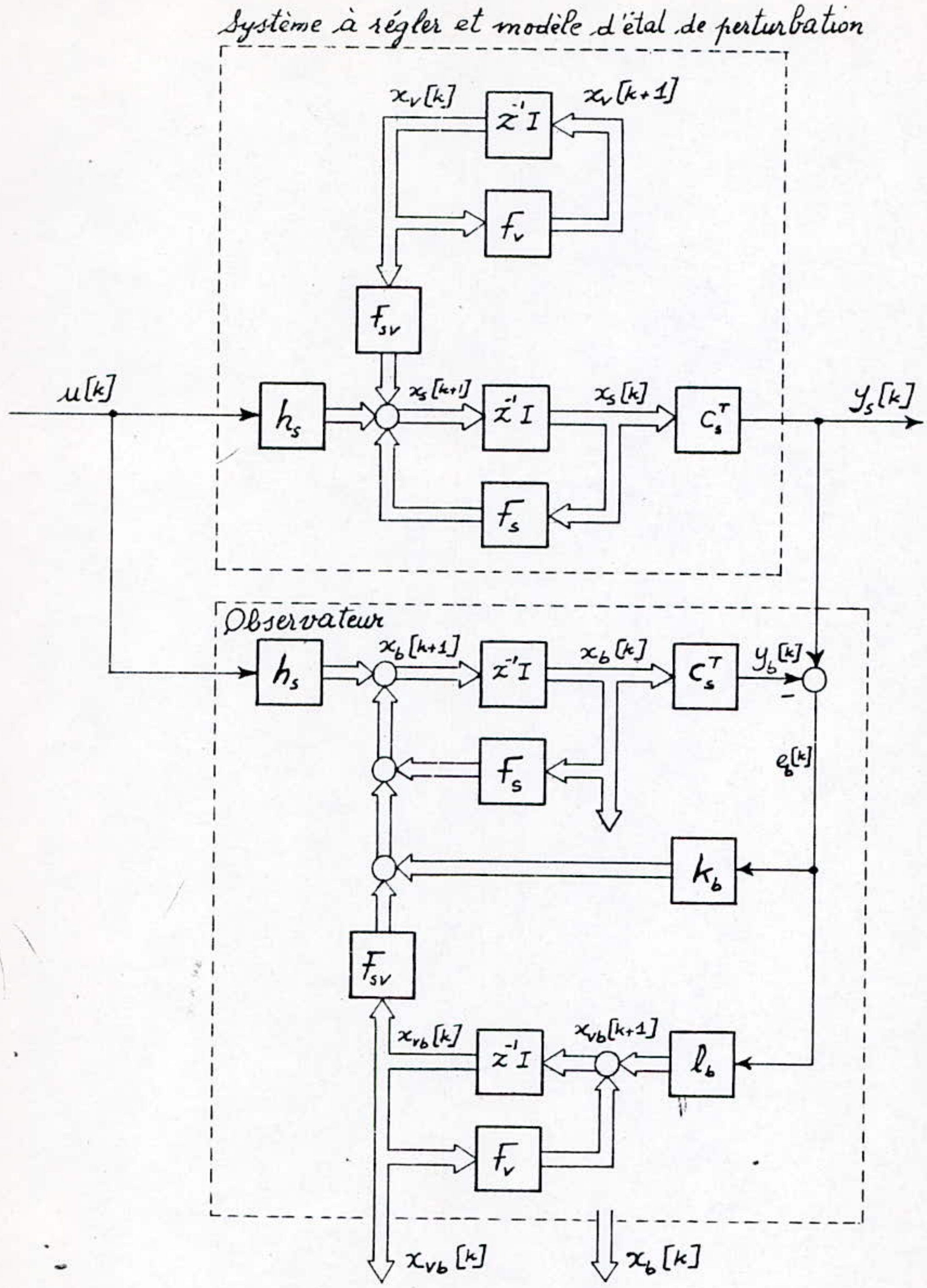


Fig 7. Diagramme structurel de l'observateur pour perturbation variable

$$\bar{X}_{bG}[k+1]=F_{bG}\ \bar{X}_{bG}[k]$$

$$F_{bG}=\begin{vmatrix} F_s-K_b\ C_s & F_{sv} \\ -L_b\ C_s^T & F_v \end{vmatrix}$$

F_v : matrice d'évolution des perturbations

F_{sv} : matrice d'intérvention des perturbations

L'algorithme consiste à calculer les matrices F_{bo}, C_{bo} par:

$$F_{bo}=\begin{vmatrix} F_s & F_{sv} \\ 0 & F_v \end{vmatrix} \quad , \quad C_{bo}=[\ C_s^T \qquad 0\ ]$$

puis appliquer l'algorithme décrit dans le cas des peturbations nulles en remplacant F_s par F_{bo} et C_s par C_{bo} on obtient alors, K_{bo} qui est sous la forme:

$$K_{bo}=\begin{vmatrix} K_b \\ -L_b \end{vmatrix}$$

3-OBSERVATEUR POUR LE CAS MULTIVARIABLE:

La strucure présentée dans 2-a peut étre étendue également pour les systémes multivariable, on a alors affaire à des matrices de réactions au lieu des vecteurs de réaction,ainsi K_b devient une matrice de contre réaction $(n \times r)$.
L'équation dynamique de l'obsérvateur s'écrit alors:

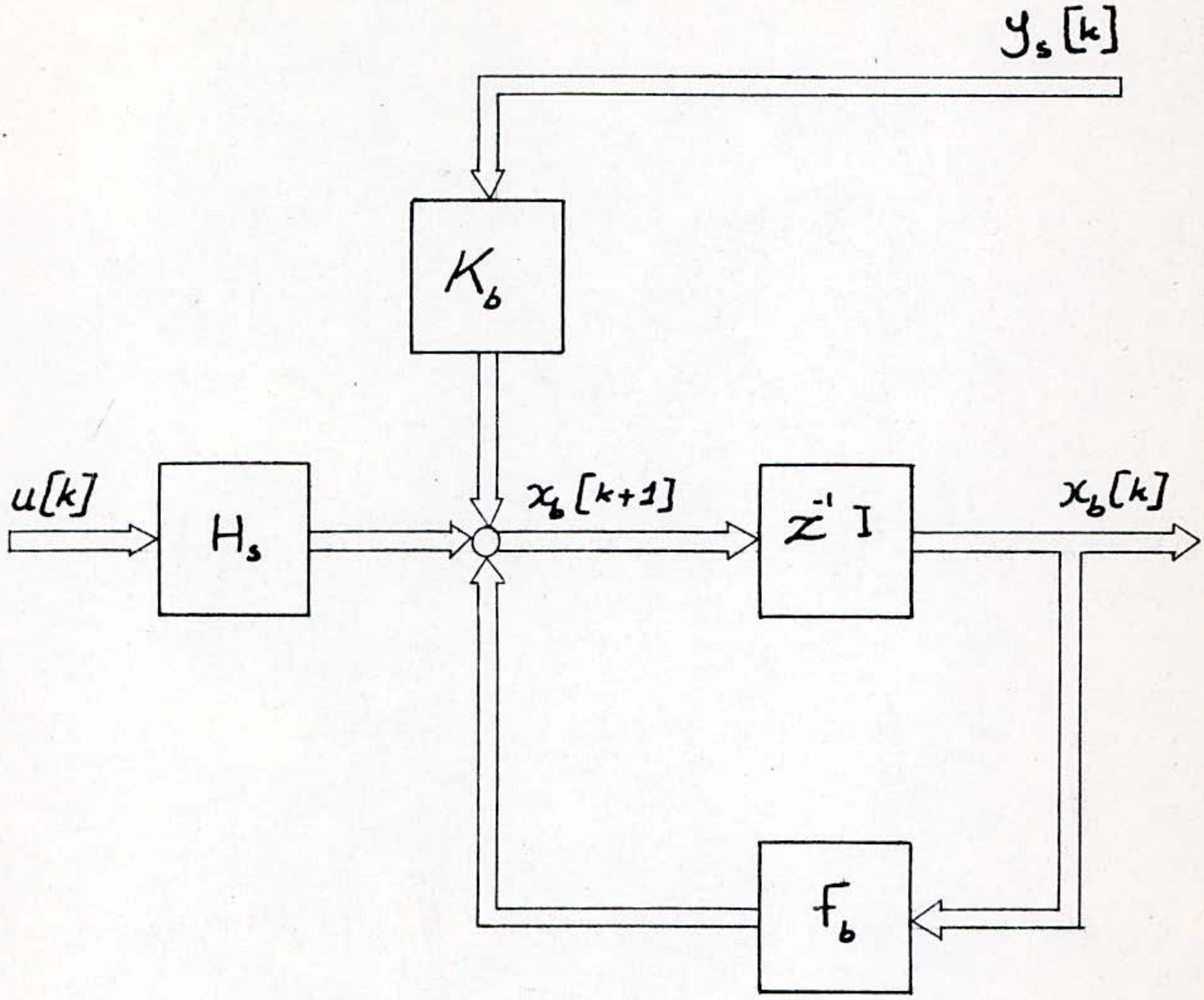


Fig 8. Diagramme structurel de l'observateur pour un système multivariable

$$X_b[k+1] = F_b\ X_b[k] + H_s\ U[k] + K_b\ Y_s[k]$$

avec $$F_b = F_s - K_b\ C_s$$

Le diagramme structurel pour l'observateur multivariable est présenté sur la -fig 8- .

le résultat concernant l'évolution de l'erreur reste valable pour le cas multivariable, à savoir, la dynamique de l'erreur d'observation peut étre imposer à l'aide de l'assignement des pôles de la matrice F_b.

L'algorithme en question consiste à :

* transposer les matrices F_s,C_s
* faire l'assignement des pôles par décomposition en P sous systémes (voir chap réglage par contre réaction d'état)
* transposer la matrice K obtenue.

4-ESTIMATEUR DE KALMAN:

L'estimateur de Kalman est un observateur asymptotique d'ordre n, il pérmet d'estimer les états d'un systéme dans un environnement stochastique (les perturbations varient de maniére aléatoire).

L'équation dynamique de l'estimateur est donnée par [3]:

$$\bar{X}[i] = F\ \bar{X}[i-1] + H\ U[i-1] + G(Y[i] - \bar{Y}[i])$$

G est calculé de maniére à minimiser la moyenne de l'erreur d'estimation, cela est obtenue par la solution du systéme:

$$\begin{cases} G[i] = K[i]\, C^T (\, C\, K[i]\, C^T + R)^{-1} \\ K[i+1] = A\, (\, I - G[i]\, C)\, K[i]\, A^T + Q \end{cases}$$

Q,R étant les variances des bruits influançant réspectivement l'équation d'état et l'équation de sortie.On suppose que ces bruits sont blancs, gaussiens et non corrélés. L'algorithme utilisé est valable pour le cas " horizon infini" c'est pourquoi, la solution qu'on obtient est asymptotique, ce qui est trés interéssant du fait qu'elle nous évite le calcul en temps réels.

II-H ORGANISATION GENERAL DU TRAVAIL:

Etant donné les objéctifs assignés, la méthodologie de conception adopté et la variétés des algorithmes, l'organisation globale du logiciel est présentée sous forme d'un schéma fonctionnel à la figure -fig 9-.

1-SHEMA FONCTIONNEL :

Le logiciel est constitué d'un programme principal soutenu par neuf unités.

- Le programme principal est axée éssentiellement sur les structures itératives ou (et) séléctives .

Le logiciel intéragit avec l'utilisateur par l'intermédiare d'un ensemble de menus et détérmine par l'analyse du mouvement

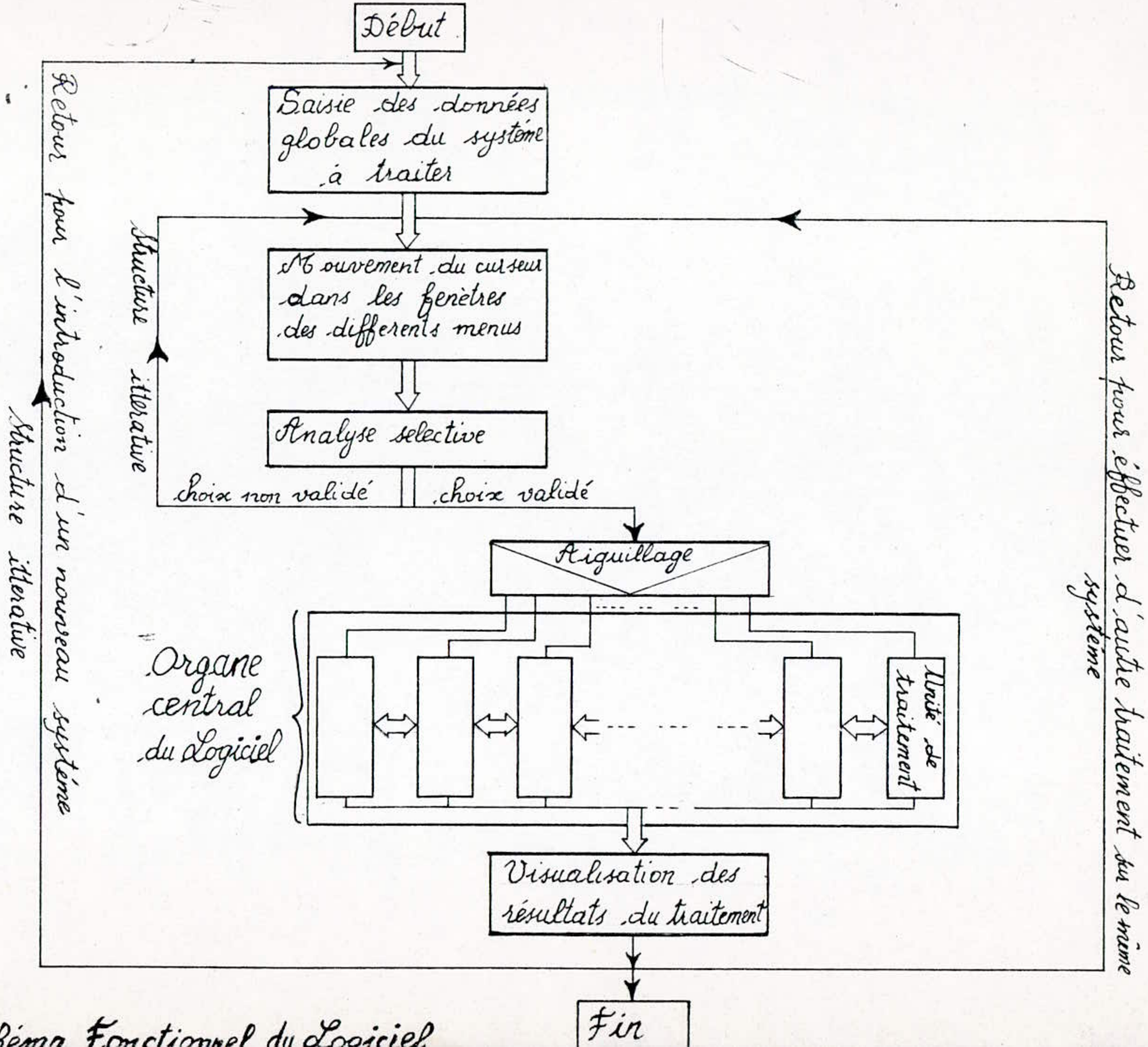


Fig 9 Shéma Fonctionnel du Logiciel

du curseur les caractéristiques génerale du probléme qu'on veut traiter.

Les résultats sont alors mis à la disposition de l'utilisateur par une visualisation sur l'écran ou par sortie sur imprimante.

Nous venons de décrir un cycle complét de fonctionnement du logiciel, se cycle est reprit en fonction du choix:

*traitement d'un autre systéme

*appliquer un autre traitement pour le même systéme

2-ORGANE CENTRAL DU LOGICIEL :

cet organe à pour tache:

* gérer l'écran graphique
* gérer les erreurs d'entrées/sorties
* faire le traitement nécéssaire
* gérer l'interaction avec le systéme d'éxploitation.

IL est constituée de neuf unités :

trois d'entre elles sont propre au turbo-pascal qui sont

-Crt:utilitaire contenant un ensemble d'objet, procédures donnant accés au controle des modes d'écran, du clavier et du curseur

-Dos:cette unité contient des déclarations de constantes, types variables, procédures et fonctions liés au systéme d'éxploitation et à la géstion des fichiers.

-graph:offre plus de 70 routines liés au graphisme.

Les unités réstantes sont :

-Win:unité d'éxtention fenétrage de l'unité crt, pérmet la géstion des fenétres en mode téxte.

-Alcrt: Permet la géstion des erreurs d'entrée-sorties et les erreurs logique.

-Almatl et Moh: Ces deux unités constituent le cerveau du logiciel, elles contiennent un ensemble trés importantes de routines spésialisées dans:

* le calcul matriciel
* l'analyse et la synthése des systémes de commande

-Alprint: Permet de copier sur l'imprimante l'écran graphique.

-Ken: Contient des routines diverses concernant l'ésthetique et l'animation graphique.

3eme

partie

COMMUNICATION HOMME-MACHINE

1-INTRODUCTION :

aprés avoir :

* fixés les objectifs et caractéristiques que doit satisfaire le logiciel

* décrit les algorithmes d'analyse et de synthése des systémes de commande, que ce logiciel doit mettre à la disposition de l'utilisateur

* présenter l'organisation globale du logiciel

Ce chapitre est consacré au mode d'exploitation du logiciel et de son interaction avec l'utilisateur (interface).
La validité du logiciel est mise en évidence par des éssais sur des exemples déja traités dans la bibliographie [1]..[3]

2-INTERFACE HOMME-MACHINE :

a-Caractéréstique de l'intérface :

L'intérface implanté posséde les caracteristiques suivantes :

- interactivité :

Dans la limite de nos moyens, nous nous sommes attachés à ce que le logiciel conçu soit, le plus transparent, ergonomique et conversa--tionnel possible.

- facilité d'utilisation :

concretiser par Le choix du mode "pilote" qui permet même à un utilisateur non initié d'étre guidé par des menus vérs l'opération qu'il désire efféctuer et l'option "help" qui offre la possibilité de demander à tout moment une assistance au logiciel.

b-Organe central de l'intérface:

Cet organe est constitué d'un ensemble de routines capables d'effectuer les taches suivantes:

* définir le jeux de caractére pour la construction des cadres entourant les fenétres;
* la construction de fenétres, notons que celle ci est défini par la nature du cadre, sa position, sa dimension, son attribus et l'attribus du titre qui lui est associé
* définir le mode de remplissage (couleur de fond) des surfaces delimités par le cadre de la fenétre.
* la sauvegarde et la restitution dans et du buffer écran des fenétres .
* la géstion des pointeurs et de l'éspace mémoire écran dans les opérations de transfert écran buffer ou buffer écran .

Ainsi on peut avoir la possibilité d'afficher autant de fenétres superposés l'une sur l'autre tout en assurant la sauvegarde des zones recouvertes.

Par conséquent on dispose grace à cet organe d'un confort et d'une ergonomie indéniable.

3-ENVIRONNEMENT DE TRAVAIL:

1-Introduction:

L'environnement de travail du logiciel propose à l'utilisateur plus d'une trentaines d'options groupés en menus deroulants .

L'accée à cet environnement se fait simplement en tapant KT90

2-Fonctionnement des Menus:

L'interaction entre le logiciel et l'utilisateur passe par l'intermédiare de choix éffectués à l'aide de menus deroulants à structure arborescente .Un menu arborescant possede pluisieurs niveaux, dont chacun constitue une subdivision du niveau qu'il lui est superieur .

Pour le logiciel le niveau le plus elevé appelé "menu principal" , est constitué de six options affichées sur la premiere ligne de l'écran.

La derniere ligne de l'écran contient la description de deux commandes obtenues à l'aide de touches de fonction ou de combinaison de touches.

Comme on peut le constater, l'écran se divise en trois zones:

* le menu principal
* la zone de travail
* la ligne de commande help et quit

Le curseur peut se trouver dans les deux premieres zones .

Dans le menu principal le curseur est reconnaissable au fait qu'il indique toujours une option, en la mettant en évidence.

Dans la zone de travail, le curseur a le même aspéct qu'au niveau du DOS .

```
SAISIE OPTIONS ORME-CANONIQUE R^Etat OBSERVATEUR REPONSE

Load                                        Estimateur_Kalman
Echantillonnage                             Observateur_Mono
File                                        Observateur_Multi
Exp(Matrice)
Inv(Matrice)
Solution Ax = b


F1-Help                   Ctrl-End Quit environnement menu
```

MENU PRINCIPAL DE L'ENVIRONNEMENT DE TRAVAIL

SAISIE OPTIONS ORME-CANONIQUE R^Etat OBSERVATEUR REPONSE

Estimateur_Kalman

Observateur_Mono

lti

Perturbations_Nulles

Perturbartions_Ctes

Perturbations_Vbles

F1-Help Ctrl-End Quit environnement menu

EXEMPLE DE RECOUVREMENT DES FENETRES DANS LE MENU

A l'aide du menu principal,l'utilisateur peut executer les differents commandes de l'environnement .

Le parcours du menu principal s'effectue à l'aide des touches de déplacement du curseur :les fléches vers la droite et vers la gauche déplacent le curseur à travers les options du menu principal, alors que les fléches vers le haut et vers le bas parcourt les differentes options d'un menu ou d'un sous menu.
Dans la zone de travail l'utilisateur peut introduire les données de son probleme ou recevoir par l'intermédiare de la fenetre d'affichage les resultats produits par le logiciel.
La maniere la plus simple de sortir d'un menu secondaire est d'appuiyer sur la touche <ESC> autant de fois que nécessaire , selon le niveau dans le quel on se trouve.
Si on se trouve dans la fenetre de travail, on ne peut en sortir qu'aprés avoir introduit tout les données necéssaire , ou aprés l'affichage de tous les resultats attendus.
IL suffit alors d'appuiyer sur la touche <ENTER> pour les quitter.

Notons que pour quitter l'environnement de travail du logiciel et rendre le controle au systéme (Disque dur ou Diskette) il suffit de taper la combinaison <CTRL>-<END>.

3- Menu Saisie :

```
SAISIE
-----------------
Load
Echantillonage
File
Exp(Matrice)
Inv(Matrice)
Solution Ax=b
-----------------
```

le menu saisie comporte six options :

Load :

permet de specifier les données du systéme à traiter, qui sont : son ordre, nombre d'entrée et les matrices A,B,C,D

En effet en apuiyant sur la touche <RETURN>, une fenétre apparait à l'utilisateur lui demandant d'introduire l'ordre du systéme puis le nombre d'entrée, des que cette opération est effectué une autre fenetre plus grande apparait donnant ainsi à l'utilisateur la possibilité d'introduire les équations de son systéme.

Echantillonnage:

grace à cette option l'utilisateur peut echantillonnée son systéme, le logiciel demande à l'utilisateur l'introduction de la periode d'échantillonnage,ensuite il doit spécifier si son systéme est avec élement de maintient ou sans, des lors les matriçes d'évolution F et d'entrée H corréspondant son affichées.

File :

réaffiche les données globale du systéme courant .

Exp(Matrice):

évalue l'exponentiel d'une matrice donnée.

Inv(Matrice):

évalue l'inverse et informe si la matrice est singuliere

Solution Ax=b:

donne la solution du systéme linéaire définie par la matrice A et le vecteur b

4-Menu Options:

```
                 Options
        ------------------------------
        Gilbert(M F T ----> R.Etat)
        R.Etat -----> M F T
        Com-Globale
        Com-Partielle
        Com-Sortie
        Obs-Globale
        Obs-Partielle
        Stabilité
        ------------------------------
```

Ce menu comporte huit options permettant de specifier certaines caractéristiques du systéme ainsi que le passage d'une représentation à une autre, autrement dit ce menu permet d'éffectuer l'analyse des systémes de commande.

Gilbert(M F T -----> R.Etat):

permet le passage de la matrice de transfert à la représentation d'état

R.Etat ------>M F T :

permet le passage de la représentation d'état à la représentation par matrice de transfert

- Com-Global:

permet d'éffectuer le test de commandabilité globale du systéme en renvoiyant un méssage.

Com-Partielle:

effectue la même fonction que l'option précedente mais le test se fait par rapport à chacunes des entrées du systéme .

Com-Sortie:

test la commandabilité en sortie du systéme.

Obs-Globale:

test l'observabilité globale du systéme

Obs-Partielle:

test l'observabilité du systéme par rapport à chacunes de ses sorties.

Stabilité:

effectue l'analyse de la stabilité d'un systéme échantillonnée ou continue.Les pôles du systéme sont affichés ainsi que le nombre de pôles instables.

5-Menu Formes-Canoniques:

Forme-Canonique

```
---->  to F C C -Mono
Décomp-en-P-S.Sys-Com
Décomp-en-M-S.Sys-Com
---->  to F C O -Mono
Décomp-en-P-S.Sys-Obs
Décomp-en-R-S.Sys-Obs
```

Le menu "Forme-Canonique" propose un ensemble de six options permettant le passage aux formes canoniques de commandabilité ou d'observabilité.

----> to F C C -Mono:

concerne les systémes monovariables, elle permet le passage à la forme canonique de réglage

Décomp-en-P-S.Sys-Com :

effectue pour les systémes multivariables le passage à la forme canonique de réglage en appliquant l'algorithme de décomposition en P sous systémes Commandables.

Décomp-en-M-S.Sys-Com :

permet le passage pour les systémes multivariables,à la forme canonique commandable via l'algorithme de décomposition en M sous systémes.

----> to F C O -Mono :

permet le passage à la forme canonique observable pour les systémes monovariables

Décomp-en-R-S.Sys-Obs:

permet le passage, pour les systémes multivariables à la forme canonique d'observation par décomposition en R sous systémes.

6-Menu R-d^Etat :

R-d^Etat

R.Etat-Mono
R.Etat-Multi
Découplage-par-R-E
Equa-Riccati

Ce menu propose quatre options permettant de faire la synthése des circuits de réglage par contre réaction d'état

R-Etat-Mono :

permet de calculer le vecteur de retour d'état pour les systémes monovariables

R-Etat-Multi:

concerne les systémes multivariables, elle donne accée à un sous menu de trois options:

qK-transpose
Décomp-en-P-S-S
Décomp-en-M-S-S

qK-transpose :

determine la matrice de retour d'état par l'algorithme qK^T

Décomp-en-P-S-S:

calcul la matrice de retour d'état via l'algorithme de décomposition en P sous systémes commandables

Décomp-en-M-S-S:

determine la matrice de retour d'état par décomposition du systéme globale fermé en M sous systémes commandables.

Découplage-par-R-Etat:

permet le découplage d'un systéme donné continue ou échantilloné par contre réaction d'état.

Equa-Riccati:

donne la solution asymptotique de l'équation de RICCATI dans le cas discrét.

7-Menu Observateur :

Observateur
Estimateur-Kalman
Observateur-Mono
Observateur-Multi

Ce menu offre trois options permettant l'estimation des états du systéme dans un environnement stochastique ou deterministe.

Estimateur-Kalman:

donne la solution asymptotique (gain de kalman) de l'estimateur de kalman pour un environnement stochastique.

Observateur-Mono :

donne accée au sous menu composé de trois options traitant le cas de l'observateur monovariable.

Perturbations-Nulles
Perturbations-Connues-Cte
Perturbations-Connues-Vble

Perturbations-Nulles:

determine le vecteur de retour de l'observateur sans tenir compte des perturbations

Perturbations-Connue-Cte :

determine le vecteur de retour de l'observateur pourvu que les perturbations demeurent constantes pendant la periode d'échantillonnage.

Perturbations-Connues-Vble:

determine le vecteur de retour de l'observateur dans le cas oû les perturbations varient de maniere deterministe.

Observateur-Multi :

determine la matrice de retour de l'observateur pour les systémes multivariables dans le cas ou les perturbations sont nulles.

8-Menu Réponse:

RPONSE

Indicielle

cette options aboutit au sous menu

Systéme-Mono
Systéme-Multi

Systéme-Mono:

permet de determiner la réponse indicielle d'un systéme monovariable échantillonné

Systéme-multi :

permet de determiner la réponse indicielle d'un systéme multivariable échantillonnée.

4-EXEMPLES D'APPLICATIONS

Nous consacrons cette section aux exemples d'applications, à titre de comparaison nous avons juger utile de faire l'essaie sur six exemples deja traités dans la litterature [1] ..[3]

Exemple d'Application N=1 :[1]

C'est un systéme à deux états et une seule entrée (monovariable).

Le systéme en question ainsi que les traitements effectués sont illustrés ci-aprés, classés par ordre chronologique est :

* échantillonnage.
* test de commandabilité.
* passage à la forme canonique de réglage.
* test d'observabilité.
* passage à la forme canonique observable.

Matrice A :

0.00000	1.00000
0.00000	-0.20000

Matrice B :

0.00000
1.00000

Matrice C :

1.00000	0.00000

Matrice D:

0.00000

Entrer La Periode D'Echantionnage...(T)

1.500

Systeme Avec Element De Maintient ?...(O/N) O

****
Matrice d'Evolution F:

1.00000	1.29591
0.00000	0.74082

****
Matrice d'Entree H :

1.02046	1.29591

************ Systeme Commandable ********

Matrice de Transformation :

0.82341	1.02046
-1.29591	1.29591

Matrice A Sous Forme Canonique Commandable:

0.00000	1.00000
-0.74082	1.74082

Vecteur D'entree Sous Forme Canonique Commandable:

0.00000	1.00000

Vecteur De Sortie Sous Forme Canonique Commandable:

0.82341	1.02046

******** Systeme Observable *******

Matrice de Transformation :

-0.74082	1.29591
1.00000	0.00000

Matrice A Sous Forme Canonique Observable:

0.00000	-0.74082
1.00000	1.74082

Vecteur D'entree Sous Forme Observable:

0.92341	1.02046

Vecteur De Sortie Sous Forme Observable:

0.00000	1.00000

Systeme Commandable En Sortie...!

Expemle d'Application N=2 : [1]

C'est un systéme multivariable échantillonné à trois états et deux entrées .

Les traitements qu'il a subit sont :

* passage à la représentation par matrice de fonction de transfert.

* calcul numerique de la réponse indicielle.

Matrice A :

0.78500	0.08700	0.08000
0.04400	0.88700	0.04400
0.08000	0.08700	0.78500

Matrice B :

0.33300	0.01100
0.00900	0.23500
0.01600	0.01100

Matrice C :

1.00000	-0.40000	-0.40000
-0.40000	1.00000	-0.40000

Matrice D:

0.00000	0.00000
0.00000	0.00000

La Matrice Multiple De Z^2 Est:

0.323	-0.087
-0.131	0.226

La Matrice Multiple De Z^1 Est:

-0.555	0.149
0.223	-0.370

La Matrice Multiple De Z^0 Est:

0.236	-0.062
-0.092	0.149

Les Coefficients Du Denominateur Par Degre Decroissant Sont:

1.000
-2.457
1.995
-0.536

```
Enther Le Vecteur D'Etat Initial X(0)

 0.00000            0.00000            0.00000

 Vecteur De Sortie Pour La Periode    0Est:
 0.00000            0.00000

Vecteur D'Etat Pour La Periode     1 Est:

 0.34400            0.24400            0.02700
 Vecteur De Sortie Pour La Periode    1Est:
 0.23560            0.09560

Vecteur D'Etat Pour La Periode     2 Est:

 0.63743            0.47675            0.09694
 Vecteur De Sortie Pour La Periode    2Est:
 0.40795            0.18300

Vecteur D'Etat Pour La Periode     3 Est:

 0.89361            0.69919            0.19557
 Vecteur De Sortie Pour La Periode    3Est:
 0.53571            0.26352

Vecteur D'Etat Pour La Periode     4 Est:

 1.12196            0.91211            0.31284
 Vecteur De Sortie Pour La Periode    4Est:
 0.63198            0.33818

Vecteur D'Etat Pour La Periode     5 Est:

 1.32912            1.11617            0.44169
 Vecteur De Sortie Pour La Periode    5Est:
 0.70598            0.40785

Vecteur D'Etat Pour La Periode     6 Est:

 1.51960            1.31196            0.57716
```

Exemple d'Application N=3 : [1]

Systéme du quatrieme ordre multivariable possedant deux entrées, on l'a soumit à :

* échantillonnage.

* retour d'état qK^T.

* retour d'état par décomposition en M sous systémes commandables.

* découplage par retour d'état.

* Observateur multivariable.

Matrice A :

-1.00000	2.00000	1.20000	-2.40000
0.00000	-2.00000	0.00000	0.00000
1.20000	-2.40000	-2.00000	4.00000
0.00000	0.00000	0.00000	-4.00000

Matrice B :

0.00000	0.00000
2.00000	0.00000
0.00000	0.00000
0.00000	4.00000

Matrice C :

1.00000	0.00000	0.00000	0.00000
0.00000	0.00000	1.00000	0.00000

Matrice D:

0.00000	0.00000
0.00000	0.00000

Entrer La Periode D'Echantionnage...(T)

0.500

Systeme Avec Element De Maintient ?...(O/N) O

****

Matrice d'Evolution F:

0.70230	0.34385	0.30381	-0.20274
0.00000	0.36788	0.00000	0.00000
0.30381	-0.33679	0.44913	0.38514
0.00000	0.00000	0.00000	0.13510

****

Matrice d'Entree H :

0.25156	-0.40544
0.63212	0.00000
-0.27085	0.71749
0.00000	0.86462

1- Avec regulateur integrateur...(A)
2- Sans regulateur integrateur...(S)

S

Entrer L'emplacement des Poles Desiree

On Commencera Par Les Parties Reels

0.20000	0.20000	0.20000	0.20000

Entrer Les Parties Imaginaire

0.31000	-0.31000	0.31000	-0.31000

Le Vecteur Q Est Donne Par :

1.00000 1.00000

Matrice De Retour D'Etat Est:

3.64873	-0.10765	2.64049	0.35222
3.64873	-0.10765	2.64049	0.35222

```
Structure Optimale...(O/N)        O
  Sous Systeme S1 D'Ordre:3

 Entrer Les Parties Reéls Du Sous Systeme S1

0.20000          0.20000          0.25000
 Entrer Les Parties Imaginaire Du Sous Systeme S1

0.31000         -0.31000          0.00000

 Sous Systeme S2 D'Ordre:3

 Entrer Les Parties Reels Du Sous Systeme S2

0.20000          0.20000          0.25000
 Entrer Les Parties Imaginaire Du Sous Systeme S2

0.31000         -0.31000          0.00000

   Matrice De Retour D'Etat Est:

 8.55253           1.03532           5.17064          0.18155
 4.02255           0.15465           3.23500          0.29936
   Matrice De regulateur integrateur :

-3.26591          -1.77770
-1.46117          -1.39025
   Voulez-Vouz Calculer Les  Matrices KC Et KP...(O/N)

   Entrer La Matrice d'Intervention Des Perturbations
  (Meme Dimension Que La Matrice B)

 0.20000           0.00000
 0.00000           0.10000
 0.25000           0.00000
 0.00000           0.20000

  Matrice D'Intervention De La Consigne Est  :

 9.57016           5.26167
 4.10519           3.88433
```

Matrice D'Intervention De La Perturbation Est :

2.81350	0.02370
1.43807	0.03911

Entrer les partis reels ..... ?

0.20000 0.20000

Entrer les partis reels ..... ?

0.00000 0.00000

Quel Pole Desirez Vous Compenser (Tapez 1 ou 2)....?

Entrer les partis reels ..... ?

0.20000 0.20000

Entrer les partis reels ..... ?

0.00000 0.00000

Quel Pole Desirez Vous Compenser (Tapez 1 ou 2)....?

Matrice d'Entree de Decouplage

10.15073	5.73599
3.83180	3.55903

Matrice d'Anticipation :

8.12059	4.58879
3.06544	2.84722

Matrice De Decouplage Globale:

14.96193	1.55851	9.10164	0.15117	-6.49647
-3.67103				
6.07141	0.11893	4.89800	0.59385	-2.45235
-2.27778				

Sous Systeme S1 De Dimension: 4

Entrez Les Poles Desirer Du Sous Systeme:1

Partie Reeles...?

0.00000	0.00000	0.00000	0.00000

Partie Imaginaire...?

0.00000	0.00000	0.00000	0.00000

Matrice de Retour d'Etat :

-1.65440	0.60406	-3.05885	0.00792
0.00000	0.00000	0.00000	0.00000

Matrice de Gain de L'Observateur :

-1.65440	0.00000
0.60406	0.00000
-3.05885	0.00000
0.00792	0.00000

Exemple d'Application N=4 :

Systéme monovariable continue à trois états, il est tiré du [2]

On l'a soumit à:

* Analyse de la stabilité.

* test de la commandabilité.

* test de la commandabilité en sortie.

* retour d'état monovariable.

* Observateur monovariable pour le cas ou les perturbations n'influent pas sur le systéme.

Matrice A :

-2.0	1.0	2.0
-1.0	-2.0	2.0
-2.0	0.0	2.0

Matrice B :

0.0
1.0
0.0

Matrice C :

1.0	0.0	0.0

Matrice D:

0.0

Systeme Ech/Cont ...(E ou C)...?

C

les poles du système sont :

0.000+ 1*-1.000

0.000+ 1* 1.000

-2.001

Systeme Stable...!

************ Systeme Commandable ********

Systeme Commandable En Sortie...!

1- Avec regulateur integrateur...(A)
2- Sans regulateur integrateur...(S)

S

Entrer L'emplacement des Poles Desiree

On Commencera Par Les Parties Reels

-1.00000	-1.00000	-2.00000

Entrer Les Parties Imaginaire

0.00000	0.00000	0.00000

Vecteur De Retour d'Etat :

4.00000	2.00000	-4.00000

Entrez Les Poles Désireé...

Partie Reels...?

0.00000	0.00000	0.00000

Partie Imaginaires...?

0.00000	0.00000	0.00000

Vecteur De Gain :

-2.00000	1.40000	-1.20000

Exemple d'Application N=5 :[3]

Systéme du quatrieme ordre possedant quatre entrées, il a subit le traitement:

* resolution de l'équation de RICCATI discréte.

Ordre du Systeme....?

Matrice Q....?

20.00000	0.00000	0.00000	0.00000
0.00000	1.00000	0.00000	0.00000
0.00000	0.00000	10.00000	0.00000
0.00000	0.00000	0.00000	5.00000

Matrice R....?

5.00000	0.00000	0.00000	0.00000
0.00000	1.00000	0.00000	0.00000
0.00000	0.00000	20.00000	0.00000
0.00000	0.00000	0.00000	10.00000

Matrice A....?

0.50000	0.00000	0.00000	0.00000
0.00000	0.50000	0.00000	0.00000
0.00000	0.00000	0.50000	0.00000
0.00000	0.00000	0.00000	0.50000

Matrice B....?

1.00000	0.00000	0.00000	0.00000
0.00000	1.00000	0.00000	0.00000
0.00000	0.00000	1.00000	0.00000
0.00000	0.00000	0.00000	1.00000

Matrice C....?

1.00000	0.00000	0.00000	0.00000
0.00000	1.00000	0.00000	0.00000
0.00000	0.00000	1.00000	0.00000
0.00000	0.00000	0.00000	1.00000

Matrice De Retour de Ricatti :

21.00970	0.00000	0.00000	0.00000
0.00000	1.13276	0.00000	0.00000
0.00000	0.00000	11.85950	0.00000
0.00000	0.00000	0.00000	5.92975

Exemple d'Application N=6 :[3]

Systéme multivariable du 6^{eme} ordre, à deux entrées nous lui avons appliquer les traitements:

* Décomposition en P sous systémes commandables (P = 1)

* Décomposition en M sous systémes commandables (M = 2)

* décomposition en R sous systémes observables (R = 1)

-*-*-*-*-*-*-*-

EN comparant les resultats que nous avons obtenu par l'intermédiaire du logiciel, avec les resultats donnés dans la litterature, on constate qu'elles sont identiques.Ce qui nous permet de dire que les programmes realisés sont valables, par consequent on peut affirmer que le logiciel est fiable et performant, au moins jusqu'a un certain ordre.

Matrice A :

1.0	1.0	1.0	1.0	1.0	1.0
0.0	1.0	1.0	1.0	1.0	1.0
0.0	0.0	1.0	1.0	1.0	1.0
0.0	0.0	0.0	1.0	1.0	1.0
0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.0	0.0	0.0	0.0	1.0

Matrice B :

1.0	0.0
1.0	0.0
1.0	1.0
1.0	0.0
0.0	1.0
1.0	1.0

Matrice C :

1.0	1.0	1.0	1.0	1.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0

Matrice D:

0.0	0.0
0.0	0.0

Matrice de Transformation :

1.0	5.0	16.0	41.0	91.0	182.0
1.0	4.0	11.0	25.0	50.0	91.0
1.0	3.0	7.0	14.0	25.0	41.0
1.0	2.0	4.0	7.0	11.0	16.0
0.0	1.0	2.0	3.0	4.0	5.0
1.0	1.0	1.0	1.0	1.0	1.0

Matrice A :

0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0
-1.0	6.0	-15.0	20.0	-15.0	6.0

Matrice B :

0.0	3.0
0.0	1.0
0.0	1.0
0.0	1.0
0.0	0.0
1.0	-1.0

Matrice C:

5.0	16.0	41.0	91.0	182.0	336.0
0.0	0.0	0.0	0.0	0.0	0.0

Matrice de Transformation :

1.0	5.0	16.0	0.0	3.0	14.0
1.0	4.0	11.0	0.0	3.0	11.0
1.0	3.0	7.0	1.0	3.0	8.0
1.0	2.0	4.0	0.0	2.0	5.0
0.0	1.0	2.0	1.0	2.0	3.0
1.0	1.0	1.0	1.0	1.0	1.0

Matrice A :

0.0	0.0	2.0	0.0	0.0	0.0
1.0	0.0	-6.0	0.0	0.0	-1.5
0.0	1.0	5.0	0.0	0.0	2.5
0.0	0.0	0.0	0.0	0.0	0.5
0.0	0.0	1.0	1.0	0.0	-1.5
0.0	0.0	-1.0	0.0	1.0	1.0

Matrice B :

1.0	0.0
0.0	0.0
0.0	0.0
0.0	1.0
0.0	0.0
0.0	0.0

Matrice C:

5.0	16.0	41.0	3.0	14.0	42.0
0.0	0.0	0.0	0.0	0.0	0.0

Matrice A sous Forme Compagne :

0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0
-1.0	6.0	-15.0	20.0	-15.0	6.0

Matrice B :

5.0	3.0
16.0	14.0
41.0	42.0
91.0	101.0
182.0	211.0
336.0	399.0

Matrice C:

1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0

<Conclusion>

CONCLUSION GENERALE

L'objectif essentiel de ce projet est de realiser un logiciel d'analyse et de synthése englobant veritablement tout l'etendue de l'approche " systéme " . Dans notre travail, Nous nous somme restreint aux systémes linéaires deterministes representés dans l'espace d'état.

Dans la mesure ou le but etait de concevoir un logiciel fiable et performant, nous avons par une approche relevant de l'analyse structuré descendante (particulierement mise en valeur par le langage Turbo-Pascal), decomposé les problemes posés en modules et sous modules.

Nous avons résolu le probleme de la communication " HOMME-MACHINE " par l'intermédiaire de menus deroulants à structure arborescente. Nous nous somme tenu en fait à mettre à la disposition de l'utilisateur un environnement de travail confortable, ergonomique et conversationnel. De cette façon, le travail pourra etre mis à profit, à la fois par des débutants et par des utilisateurs plus avertis.

Nous avons pu montré la validité du logiciel en comparant les résultats obtenue pour des exemples deja traités dans la littérature.

Pour les perspectives d'extentions eventuels, nous suggerons l'incorporation au logiciel d'autres unites :

* Completant les traitements non existant dans notre travail
* etendant l'etude faite aux modeles dynamique stochastiques
* traitant les problemes de commande en utilisant les techniques de l'algebre polynomiales.

Au terme de ce mémoire nous esperons avoir mis à la disposition des utilisateurs interessé par la commande des systémes un outils performant et efficace d'aide à l'analyse et la synthése dans l'éspace d'état.

< ANNEXE >

ANNEXE 2

Cet annexe est consacré à la presentation de deux sous programmes.

Le premier sous programme permet de determiner le nombre de colonnes indépendantes d'une matrice donnée.

Le second correspond à l'algorithme donnant une matrice de retour d'etat à rang maximal .

REMARQUE :

Ces deux sous programmes contiennent plusieurs procedures et fonctions definies dans le logiciel

```
procedure INLI(var phi:tmatrice;nl,nc:integer;VAR k2:integer);
{***************************************************************
  Cette procedure permet de detecter le nombre total
       de vecteurs indépendants.
  -PHI:Matrice à traiter (chaque collonne est considérée
       comme un vecteur
  -NL,NC:sont respectivement le nombre de ligne et
        le nombre de collonne de la matrice PHI.
  -K:le nombre de collonnes indépendantes detectée.
  **************************************************************}
var
  s5,i,j,kl,l:integer;
   v:tvect;{****** v est de type vecteur(à definir)*****}
procedure tt(v:tvect;n:integer;var k:integer);
label 50;
var
   i,l,j,icm,ik,ind2,imm,ja,jj:integer;
   b,c,t,x,vli:tvect;
   e,a:tmatrice;
   rnorm,anorm,s:real;
   function ij(i,j:integer):integer;
   begin
   ij:=round(j*(j-1)/2+i);

   end;
begin
     ind2:=0;rnorm:=0;k:=k+1;
     for i:=1 to nl do   begin
     rnorm:=rnorm+abs(v[i])/n ;
     e[i,k]:=v[i];end;
     if k=1 then
     50:  begin
          anorm:=0;
          for i:=1 to nl do
          anorm:=anorm+e[i,k]*e[i,k];
          rnorm:=sqrt(anorm);
          for i:=1 to nl do
          begin
               if rnorm<>0 then
               e[i,k]:=e[i,k]/rnorm;
               vli[i]:=e[i,k];
           end;
           t[ij(k,k)]:=rnorm;
           exit;
      end
      else
          begin
               icm:=k-1;
               for i:=1 to icm do
               t[ij(i,k)]:=0;
               for i:=1 to icm do begin
               for j:=1 to nl do
               t[ij(i,k)]:=t[ij(i,k)]+v[j]*e[j,i];
               for ik:=1 to nl do
               e[ik,k]:=e[ik,k]-t[ij(i,k)]*e[ik,i];
```

```
            end;
            for i:=1 to nl do
            begin
            s:=abs(e[i,k])-0.00001*rnorm;
            if s>0 then goto 50 ;
            end;
            ind2:=1;
        end ;
        if ind2=1 then k:=k-1;
  end;
  {**********Debut du programme principale***************}
        begin
        k1:=0;
for i:=1 to nc do begin
    s5:=0;
      for j:=1 to nl do  begin
    v[j]:=phi[j,i]  ;
    if abs(v[j])<1e-8 then s5:=s5+1;
    end;
    if s5<nl then
         tt(v,nl,k1);
    if i=nc  then k2:=k1;
    end;
    end;
{*********Fin de la procedure INL1**************}
```

```
procedure remu( a,b,c:tmatrice;n,m:integer;var re,im:tvect;
var kk,q,q1,phi,ni,e1,q2,w,kw,kv,hn:tmatrice;
var e,d,r,a1,g:tvect1);

{***********************************************************
  Cette procedure calcul la matrice de retour d'etat
       dont le rang est egal au nombre d'entree du systeme.
       (voir à ce propos,la description de la methode
             dans [1] )
  ***********************************************************}

label 10;
var
   i,j,j1,i1,k,k1,l,l3,s3,k2,k3,l4,s4,s6:integer;
   ALF,e5,d1,de:TVECT;
   ks,id,bsv,w1,bb,qq,w3:tmatrice;
   ind:array[1..10] of integer;
   ts:boolean;
   rr:real;
   x:char;
begin
     repeat
     gotoxy(20,30);
     delline;
     write('Structure Optimale...(O/N)'):{*** systeme echantillonnees
     ou continue.}
     x:=upcase(readkey);
     until x in['O','N'];
     ts:=false;
     if x='O' then begin
     ts:=true;
     for i:=n+1 to n+m do
     for j:=1 to n do begin
      a[i,j]:=-c[i-n,j];
      a[j,i]:=0; end;
      for i:=n+1 to n+m do
      for j:=n+1 to n+m do begin
      b[j,i-n]:=0;
      if i=j then a[i,j]:=1
      else a[i,j]:=0;end;
     n:=n+m; end;
     for i:=1 to n do
     for j:=1 to m do begin
     kk[i,j]:=b[i,j];
     w3[i,j]:=b[i,j];end;
     for k:=0 to n-1 do
     begin
          for i:=1 to m do begin
          i1:=i+m*k ;
          for j:=1 to n do
          phi[j,i1]:=kk[j,i];
          end;
          prodmat(a,kk,kk,n,n,n,m);
      end;
      incor(a,b,n,m,phi,ni);
      for i:=1 to m do
```

```
for j:=1 to n do
kk[i,j]:=0;
  for i:=1 to m do
          for j:=1 to m do
           if i=j then hn[i,j]:=1 else hn[i,j]:=0;
for k:=1 to m do
begin
     l:=round(ni[1,k]);clrscr;
     writeln('Sous Systeme S',k,' D''Ordre:',l);
     writeln;
     writeln('Entrer Les Parties Reels Du Sous Systeme S',k);
     readvect(5,re,l);clrscr;
     writeln('Entrer Les Parties Imaginaire Du Sous Systeme S',k);
     readvect(6,im,l);clrscr;
     coef(re,im,l,ai);
     l:=round(ni[1,k]*m);
     if ni[1,k]=n/m then begin
     for i:=1 to l do
     for j:=1 to n do
     q[j,i]:=pni[j,i]   ;
          for i:=1 to l do
          if i<>(l-m+k)  then d[i]:=0;
          d[l+k-m]:=1;
          res1(q,d,n,e,q2);
      end;
      if ni[1,k]>n/m then
      begin
           for j:=1 to n do begin
           q[j,1]:=pni[j,l+k-m];
           q1[j,1]:=q[j,1];end;
           k2:=1;s3:=1;l3:=1;s4:=n;ind[1]:=l+k-m;
           for i:=1 to l do
           begin
                k3:=k2;
                l3:=l3+1;
                if i<>l+k-m then begin
                for j:=1 to n do
                q1[j,l3]:=pni[j,i];
                inli1(q1,n,l3,k2);
                if k2>k3 then
                begin
                     s3:=s3+1;
                     for j:=1 to n do
                     q[j,s3]:=q1[j,l3];
                     ind[s3]:=i;
                end;
                if k2=k3 then begin
                s4:=s4+1;
                for j:=1 to n do
                qq[j,s4-n]:=q1[j,l3];
                ind[s4]:=i;
                end;
                if ((k2=n) and (k2<>k3)) then
                begin
                for i1:=2 to n do
                d[i1]:=0;
                d[1]:=1;
```

```
            res1(q,d,n,e,q2);end;
            if i=l then begin
            ovm(qq,e,s4-n,n,de);
            for j:=n+1 to l do
            e5[j]:=de[j-n];
            for j:=1 to n do
            e5[j]:=d[j];
            for j1:=l downto 1 do
            a1[ind[j1]]:=e5[j1];
            for j1:=l-m+1 to l do
            nn[k,j1-l+m]:=a1[j1];
            end; end; end;
            goto 10;
end;
      if ni[1,k]<n/m then
   begin
            s3:=0;k2:=0;
            for i:=1 to l do
            if i<>(l-m+k) then d[i]:=0;
            d[l-m+k]:=1;
            for i:=1 to n do
            begin
                 k3:=k2;
                 for j:=1 to l do
                 a1[j,i]:= pni[i,j];
                 inli1(a1,l,i,k2);
                 if k2>k3 then
                 begin
                 s3:=s3+1;
                 for j:=1  to l do
                 q[j,s3]:=a1[j,i];
                 ind[s3]:=i;end;
                 if k2=l then
                 begin
                 mattranspose(q,l,l,q);
                 res1(q,d,l,e,q2);
                 for i:=l+1 to n do
                      e[i]:=0;
                      for j:=l downto 1 do
                      begin
                            rr:=e[j];
                            e[j]:=e[ind[j]];
                            e[ind[j]]:=rr;
                       end;
                      goto 10;
                      end;
                      end;
   end;
      10: l:=round(ni[1,k]+1);
          for i:=1 to l do
          begin
               for j:=1 to n do
               e1[i,j]:=e[j];
               ovm(a,e,n,n,e);
               if i<=ni[1,k] then r[i]:=ai[l-i];
```

```
            end;
            r[l]:=1;
            pvm(et,r,n,l,g);
            for i:=1 to n do
            kk[k,i]:=g[i];
      end;
      if ts then n:=n-m;
      matinverse(hh,m,hh);
      prodmat(hh,kk,kk,m,m,m,n);
      writeln;
     writeln('Matrice De Retour D''Etat Est:');
     writemat(kk,m,n);
     x:=readkey;
    clrscr;
    repeat
    clrscr;
    gotoxy(10,40);
    writeln('Voulez-Vouz Calculer Les  Matrices KC Et KP...(O/N)');
    x:=upcase(readkey);
    until x in['O','N'] ;
    if x='O' then begin
    writeln;
    writeln('Entrer La Matrice d''Intervention Des Perturbations');
    writeln('(Meme Dimension Que La Matrice B)');
    writeln;
    readmatc(5,bsv,n,m);
      for i:=1 to m do
      for j:=1 to n do
      ks[i,j]:=kk[i,j];
      prodmat(w3,ks,w,n,m,m,n);
      diff2mat(w,a,w,n,n);
      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;
      somme2mat(w,id,w,n,n);
      matinverse(w,n,w1);
      prodmat(c,w1,w,m,n,n,n);
      prodmat(w,w3,w,m,n,n,m);
      matinverse(w,m,kw);writeln;
      writeln('Matrice D''Intervention De La Consigne Est  :');
      writemat(kw,m,m);
      x:=readkey;
      prodmat(kw,c,w,m,m,m,n);
      prodmat(w,w1,w,m,n,n,n);
      prodmat(w,bsv,kv,m,n,n,m) ;writeln;
      writeln('Matrice D''Intervention De La Perturbation Est :');
      writeln;
      writemat(kv,m,m);end;

end;
{*******Fin de la procedure REMU **********************}
```

BIBLIOGRAPHIE

* [1] REGLAGE ECHANTILLONNE T2: TRAITEMENT DANS L'ESPACE D'ETAT

HANSRUED BUHLER 1982 (P P R)

* [2] COMMADE DES SYSTEMES MULTIDIMENSIONNELS

FOSSARD (PARIS 1972 DUNOD)

* [3] COMMANDE ET REGULATION PAR CALCULATEUR NUMERIQUE

Claude FOULARD SANDAZ GENTIL

5eme EDITION EYROLLES

* [4] COMMANDE NUMERIQUE DES SYSTEMES

OUVRAGE COLLECTIF PUBLIE SOUS LA DIRECTION DE

C.FARGEON MASSON 1986

* [5] LINEAR CONTROL SYSTEM ANALYSIS AND DESIGN

J.D'AZZO C.H.HOUPIS

Mc GRAW HILL 2nd EDITION 1984

* [6] MATHEMATICAL DESCRIPTION OF LINEAR DYNAMICAL SYSTEMS

KALMAN, R.E

J.SOC IND APPL.MATH, SER.A,CONTROL,VOL 1,No2,1963

METHODOLOGIE DE CONCEPTION DES LOGICIELS

* [7] LA PROGRAMMATION EN PASCAL

PETER GROGONO I.EDITION PARIS 1986

* [8] PROBLEM SOLVING AND COMPUTER PROGRAMMING

PETER GROGONO, NELSON 1982

* [9] STRUCTURED PROGRAMMING

TAHL, DIJKSTRA , HOARO 1972

ANALYSE NUMERIQUE

* [10] A FIRST COURSE IN NUMERICAL ANALYSIS (2end EDITION)

RALSTON AND P.RABI NOWITZ 1978

* [11] NUMERICAL ANALYSIS

R.L.BURDEN , J.D.FAIRS,A.C.REYNOLDS 2^{nd} EDITION

PWS PUBLISHERS 1981

LANGAGE TURBO PASCAL

* [12] TURBO PASCAL V 5 REFERENCE AND USER MANUAL

BORLAND INTERNATIONAL 1988

* [13] TURBO PASCAL VERSION 4 MANUEL DE PROGRAMMATION

J.C.ARMICI EDITIONS LISA 1986