Sc N

UNIVERSITÉ DE NANCY — FACULTÉ DES SCIENCES

Sc N 69
97

THESE

présentée à la

FACULTÉ DES SCIENCES DE L'UNIVERSITÉ DE NANCY

pour l'obtention du

DOCTORAT DE SPÉCIALITÉ MATHÉMATIQUES

par

Madame Odile FOUCAUT

AJUSTEMENT DE PARAMÈTRES VARIABLES SUR DES SYSTÈMES DIFFÉRENTIELS NON LINÉAIRES PAR LA PROGRAMMATION DYNAMIQUE.

UNIVERSITÉ DE NANCY
BIBLIOTHÈQUE

Soutenue publiquement le 15 Novembre 1969, devant la Commission d'Examen.

Membres du Jury :

Président : **M. J. LEGRAS**

Examinateurs : **MM. C. GILORMINI**
J. MARTIN

69
97

UNIVERSITÉ DE NANCY FACULTÉ DES SCIENCES

THESE

présentée à la

FACULTÉ DES SCIENCES DE L'UNIVERSITÉ DE NANCY

pour l'obtention du

DOCTORAT DE SPÉCIALITÉ MATHÉMATIQUES

par

Madame Odile FOUCAUT

AJUSTEMENT DE PARAMÈTRES VARIABLES SUR DES SYSTÈMES DIFFÉRENTIELS NON LINÉAIRES PAR LA PROGRAMMATION DYNAMIQUE.

Soutenue publiquement le 15 Novembre 1969, devant la Commission d'Examen.

Membres du Jury :

Président : **M. J. LEGRAS**

Examinateurs : **MM. C. GILORMINI**
J. MARTIN

Université de Nancy — Faculté des Sciences

ANNEE SCOLAIRE 1968/69

DOYEN : M. AUBRY

ASSESSEUR : M. GAY

Doyens honoraires : MM. CORNUBERT - ROUBAULT.

Professeurs honoraires : MM. RAYBAUD - LAFFITTE - LERAY - JOLY - LAPORTE - EICHBORN - CAPELLE - GODEMENT - L. SCHWARTZ - DIEUDONNE - DE MALLEMAN - LONGCHAMBON - LETORT - DODE - GAUTHIER - GOUDET - OLMER - CORNUBERT - CHAPELLE - GUERIN - WAHL.

Maîtres de Conférences honoraires : MM. LIENHART - PIERRET - Mle MATHIEU.

PROFESSEURS

MM.	ROUBAULT	Géologie		GAYET	Physiologie
	VEILLET	Biologie animale		HADNI	Physique
	BARRIOL	Chimie théorique	*	BASTICK	Chimie
	BIZETTE	Physique		DUCHAUFOUR	Pédologie
	GUILLIEN	Electronique		GARNIER	Agronomie
	LEGRAS	Mécanique rationnelle		NEEL	Chimie organique industrielle
	BOLFA	Minéralogie		BERNARD	Géologie appliquée
	NICLAUSE	Chimie	*	CHAMPIER	Physique
	FAIVRE	Physique appliquée	*	GAY	Chimie biologique
	AUBRY	Chimie minérale		STEPHAN	Zoologie
	COPPENS	Radiogéologie	*	CONDE	Zoologie
	DUVAL	Chimie	*	WERNER	Botanique
	FRUHLING	Physique		EYMARD	Calcul différentiel et intégral
	HILLY	Géologie			
	LE GOFF	Génie chimique		LEVISALLES	Chimie organique
	SUHNER	Physique expérimentale		FELDEN	Physique
	CHAPON	Chimie biologique	*	GOSSE	Mécanique physique
	HEROLD	Chimie minérale industrielle	*	DAVOINE	Physique (ENSMIN)
	SCHWARTZ B.	Exploitation minière		HORN	Physique (1° cycle)
	MALAPRADE	Chimie	*	ROCCI	Géologie
*	MANGENOT	Botanique	* Mme	LUMER	Mathématiques
				DELPUECH	Chimie physique

N...	Chimie biologique
N...	Mécanique appliquée
N...	Analyse supérieure
N...	Méthodes mathématiques de la physique
N...	Mécanique rationnelle

(*) Professeur titulaire à titre personnel

PROFESSEURS SANS CHAIRE

Mme	BASTICK	Chimie P. C. Epinal	MM.	FLECHON	Physique P. C.
MM.	GUDEFIN	Physique	Mle	HUET	Mathématiques C. B. G.
	VUILLAUME	Psychophysiologie		VIGNES	Métallurgie
	FRENTZ	Biologie animale		BALESDENT	Thermodynamique, Chimie appliquée (ENSIC
	MARI	Chimie (ISIN)		BLAZY	Minéralogie appliquée (ENSG)
	AUROUZE	Géologie		JANOT	Physique P. C. Epinal
	DEVIOT	Physique du solide		CACHAN	Entomologie appliquée (ENSA)

MAITRES DE CONFERENCES

MM	JACQUIN	Pédologie et chimie agricoles	MM.	BAVEREZ	Chimie (ENSIC)
	MAINARD	Physique M. P.		CHAMBON	Exploitation minière (Mines)
	MARTIN	Chimie P. C.		HUSSON	Physique (ENSEM)
	PAULMIER	Mécanique expérimentale		GERL	Physique
				WEISSLINGER	Physique
	PROTAS	Minéralogie		ROQUES	Chimie minérale
	JOZEFOWICZ	Physico-chimie		PERRIER	Mathématiques
	JURAIN	Géologie C. B. G.		N...	Mécanique des fluides (ISIN)
	RIVAIL	Chimie appliquée (CUCES)			
	VILLERMAUX	Génie chimique		N...	Mécanique (ISIN)
	METCHE	Biochimie appliquée (Brasserie)		N...	Mathématiques
				N...	Mathématiques P. C.
	PAIR	Mathématiques appliquées		N...	Mathématiques C. B. G.
	BAUMANN	Physique 1° cycle		N...	Physiologie animale
	DURAND	Physique		N...	Mathématiques M. P.
	GRANGE	Physique (ISIN)			
	DEPAIX	Probabilités et statistiques			

CHARGES D'ENSEIGNEMENTS

MM. AMARIGLIO, COEURE, DAVRAINVILLE, GILORMINI, GIRARDEAU, HILY, MAURIN, NOVERRAZ, OVAERT, RUYER, WEBER.

Je tiens à exprimer ma profonde reconnaissance à Monsieur le Professeur LEGRAS, Directeur de l'Institut Universitaire de Calcul Automatique de Nancy, pour l'attention avec laquelle il a dirigé mon travail et pour la formation qu'il m'a donnée.

J'exprime toute ma gratitude à Monsieur le Professeur MARTIN pour son accueil au sein de la section d'Informatique Médicale de la Faculté de Médecine de Nancy et pour l'intérêt qu'il a toujours porté à mes recherches.

Je remercie Monsieur GILORMINI, Maître de Conférence à la Faculté des Sciences de Nancy pour l'honneur qu'il m'a fait en acceptant de participer à ce jury.

Certains programmes, nécessaires à la réalisation de ce travail, ont été testés sur l'IBM 1800 de l'Ecole Nationale Supérieure d'Electricité et de Mécanique. Je tiens ici à remercier Monsieur le Professeur GOSSE, Directeur de l'Ecole, et Monsieur le Professeur FELDEN qui nous y ont autorisés.

Ce travail a fait l'objet d'un contrat de la Délègation Génèrale de la Recherche Scientifique et Technique (n° 67.01.006). Je remercie le Comité Calculateur et le Génie Biologique et Médical pour leur appui.

SOMMAIRE

INTRODUCTION

CHAPITRE I - FORMULATION DU PROBLEME

I Le problème d'ajustement vu sous l'angle de la commande optimale

I. 1 Définition d'un problème de commande optimale
I. 2 Application à l'ajustement de fonctions dans un système différentiel

II Méthode de résolution : programmation dynamique

II. 1 Définition - Principe d'optimalité.
II. 2 Application du principe d'optimalité à l'ajustement de fonctions.

II. 2. 1 Rappel du problème posé
II. 2. 2 Equation fonctionnelle fondamentale

II. 3 Différentes interprétations du principe d'optimalité

II. 3. 1 Forme discrète

II. 3. 1. 1 Partage de l'intervalle d'étude imposé
II. 3. 1. 2 Optimisation du partage
II. 3. 1. 3 Comparaison des deux points de vue.

II. 3. 2 Forme continue

II. 3. 2. 1 Rappels sur l'établissement de l'équation de Hamilton-Jacobi
II. 3. 2. 2 Manque d'intéret de cette forme dans le cas étudié.

III - Conclusion.

CHAPITRE II - AJUSTEMENT D'UNE FONCTION DE COMMANDE SUR UNE EQUATION DIFFERENTIELLE QUELCONQUE

I Méthode générale de résolution d'une équation fonctionnelle - Problèmes posés.

I. 1 Résolution d'une équation fonctionnelle - Obtention de la solution
I. 2 Principales phases de calcul

II Application à l'ajustement d'une fonction de commande sur une équation différentielle

II. 1 Etablissement de l'équation fonctionnelle dans le cas du partage imposé
II. 2 Différentes méthodes de résolution envisagées

II. 2. 1 Utilisation de la méthode du trapèze pour le calcul de l'intégrale et de la méthode d'Euler pour la résolution de l'équation fonctionnelle

Solution explicite
II. 2. 2 Résolution de l'équation différentielle par la méthode de Kutta-Runge
II. 2. 2. 1 Méthode d'optimisation à une variable
II. 2. 2. 2 Dénombrement sur les valeurs prises par la fonction d'état
II. 2. 2. 3 Utilisation de l'interpolation.

III Résultats numériques
III. 1 Premier exemple
III. 1. 1 Solution explicite
III. 1. 2 Méthode avec dénombrement
III. 1. 3 Méthode avec interpolation
III. 1. 4 Conclusion.
III. 2 Second Exemple
III. 3 Troisième exemple : équation non linéaire
III. 4 Remarque.

IV Conclusion.

CHAPITRE III - AJUSTEMENT DE DEUX FONCTIONS DE COMMANDE SUR UN SYSTEME DIFFERENTIEL D'ORDRE DEUX.

I Méthode de résolution dans le cas du partage imposé
I. 1 Système choisi et équation fonctionnelle associée
I. 2 Impossibilité d'obtenir une solution explicite
I. 3 Méthode du gradient
I. 3. 1 Principe de la méthode
I. 3. 2. Application à notre problème
I. 4 Dénombrement sur le vecteur d'état
I. 5 Utilisation de l'interpolation.

II Applications numériques dans le cas du partage imposé
II. 1 Système linéaire à coefficients variables
I. 1. 1 Méthode avec dénombrement
I. 1. 2 Méthode avec interpolation
II. 2 Système non linéaire
II. 3 Difficultés rencontrées
II. 3. 1 Difficultés matérielles
II. 3. 2 Autres difficultés.

III Méthode utilisant l'optimisation du partage

III. 1 Equation fonctionnelle

III. 2 Méthode de résolution

III. 3 Application numérique

III. 4 Comparaison avec la première méthode.

CONCLUSION

INTRODUCTION

Les physiciens comme les biologistes utilisent fréquemment des modèles mathématiques pour décrire et étudier les phénomènes réels. Ces modèles font intervenir, d'une part, des grandeurs directement accessibles à l'expérience et, d'autre part, des paramètres inconnus qui sont, suivant les cas, constants ou variables. Ces paramètres ont, en général, une signification physique et leur détermination apporte des connaissances supplémentaires sur le processus étudié. Le problème posé aux mathématiciens est donc la recherche de ces paramètres inconnus à partir du modèle mathématique et des résultats expérimentaux, c'est pour eux "un problème d'ajustement".

Des études précédentes ont traité l'ajustement de paramètres sur des systèmes d'équations différentielles linéaires. La méthode employée, basée sur l'utilisation de la transformation de Laplace a été appliquée à des phénomènes biologiques (7). La même méthode a permis l'ajustement de paramètres sur des systèmes d'équations aux dérivées partielles avec une application à un problème de diffusion de la chaleur (5). Enfin une étude théorique a montré (2) la convergence de cette méthode utilisée sur des systèmes différentiels non linéaires d'un type particulier. Dans tous ces travaux les paramètres ajustés étaient des constantes numériques.

Le problème qui est posé maintenant par les biologistes est un ajustement sur un système différentiel non linéaire, d'un type particulier, puisqu'il ne s'agit plus comme précédemment d'ajuster des paramètres constants mais des fonctions du temps ou d'une variable du système. La méthode fondée sur la transformation de Laplace n'étant plus applicable dans le cas où les paramètres sont des fonctions d'un type inconnu, nous avons considéré l'ajustement comme un problème particulier de commande optimale et nous l'avons donc abordé par une méthode de résolution propre à la commande optimale : La programmation dynamique.

Notre travail se présente de la façon suivante :

- Dans la première partie nous montrons comment l'ajustement de fonctions entre dans le cadre de la commande optimale ; nous établissons les équations fonctionnelles nécessaires à sa résolution à partir de la théorie de la programmation dynamique.

- Dans le second chapitre nous proposons plusieurs méthodes de résolution de ces équations fonctionnelles et nous les appliquons à l'ajustement d'une fonction de commande sur une équation différentielle.

- Dans le troisième chapitre nous traitons l'ajustement de deux fonctions de commande sur un système différentiel d'ordre deux et nous donnons quelques exemples numériques. Nous ne traitons pas le problème biologique qui est à l'origine de cette étude par insuffisance du matériel mis à notre disposition actuellement.

CHAPITRE I

FORMULATION DU PROBLEME

I - LE PROBLEME D'AJUSTEMENT VU SOUS L'ANGLE DE LA COMMANDE OPTIMALE.

I.1 Définition d'un problème de commande optimale

Cette définition est celle donnée par Naslin (8). Soit un processus décrit par un système d'équations différentielles, en général non linéaires. Un tel système peut toujours être mis sous la forme de système d'équations différentielles du premier ordre en introduisant éventuellement des variables supplémentaires.

Soit :

$$\begin{cases} \dfrac{dx_1}{dt} = f_1 (x_1, x_2 \cdots x_n, u_1, u_2 \cdots u_m, z_1, z_2 \cdots z_p, t) \\ \dfrac{dx_2}{dt} = f_2 (x_1, x_2 \cdots x_n, u_1, u_2 \cdots u_m, z_1, z_2 \cdots z_p, t) \\ \vdots \\ \dfrac{dx_n}{dt} = f_n (x_1, x_2 \cdots x_n, u_1, u_2 \cdots u_m, z_1, z_2 \cdots z_p, t) \end{cases}$$

Les variables sont classées de la façon suivante :

- $x_1, x_2 \cdots x_n$ désignent les variables d'état du système : leur connaissance à un instant donné définit complètement l'état du système
- $u_1, u_2 \cdots u_m$ désignent les variables de commande ou de contrôle
- $z_1, z_2 \cdots z_p$ désignent les perturbations extérieures agissant sur le système.

Pour simplifier l'écriture nous adoptons une notation vectorielle ainsi

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$ sera le vecteur d'état

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$$ sera le vecteur commande

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_p \end{bmatrix}$$ le vecteur de perturbation

Le système s'écrit alors sous forme vectorielle : $\frac{dx}{dt} = f\ (x, u, z, t)$.

Soit S (x) le vecteur de sortie et E (x) le vecteur d'entrée du système.

Le problème de commande optimale consiste à élaborer le vecteur de commande à partir de la connaissance du vecteur d'entrée, du vecteur de sortie, du vecteur de perturbation et du vecteur d'état de façon à satisfaire un critère d'optimalité, c'est-à-dire de façon à extrémaliser une certaine fonction des variables de commande et de sortie et éventuellement du temps.

Nous allons voir comment le problème d'ajustement entre dans le cadre de cette définition.

I. 2 Application à l'ajustement de fonctions sur un système différentiel

Soit un processus réel décrit par le système différentiel suivant :

$$\frac{dx}{dt} = f\ (x, t, u)$$

où t représente la variable temps, x est un vecteur de fonctions de t continues et dérivables sur un intervalle de R $[T_0, T]$, u est un vecteur de fonctions de t ou de x appartenant à un ensemble de fonctions noté U, enfin f est un vecteur de fonctions de t de u et de x.

Certaines variables $x_1, \ldots, x_r$, par exemple, peuvent être atteintes expérimentalement, nous noterons $g_1 \ldots g_r$ les solutions expérimentales correspondantes. Les fonctions $u_1 \ldots u_p$ sont inconnues ; le but de l'ajustement est de les déterminer à partir de la connaissance du système et des solutions g_i, c'est-à-dire de trouver un ensemble de fonctions $u_1^*, u_2^*, \ldots u_p^*$ tel que les solutions du

système différentiel $\frac{dx}{dt} = f(x, u^*, t)$ rendent les différences entre les fonctions x_i et g_i aussi petites que possible.

L'analogie avec le problème de commande optimale s'en déduit immédiatement :

- x est le vecteur d'état
- u est le vecteur de commande
- il n'y a pas de vecteur perturbation mais la présence d'un tel vecteur n'apporte pas de modifications à la théorie
- le vecteur d'entrée est le vecteur g des fonctions expérimentales connu
- le vecteur de sortie se confond avec le vecteur d'état
- le critère d'optimalité s'énonce ici : la différence entre le vecteur d'entrée et le vecteur de sortie est minimale.

Dans notre étude nous adopterons ces notations et nous prendrons la différence au sens des moindres carrés. Nous ne traiterons l'ajustement que sur un intervalle fini que nous noterons $[T_0, T]$.

II - METHODE DE RESOLUTION : LA PROGRAMMATION DYNAMIQUE

L'ajustement de fonctions sur des systèmes différentiels entrant, comme nous venons de le voir, dans le cadre des problèmes de commande optimale de processus non linéaires, nous avons cherché à le résoudre en appliquant une technique couramment utilisée dans ce genre d'étude, à savoir : "La programmation dynamique".

Nous avons été guidé dans le choix de cette méthode par les résultats qu'elle a donnés depuis un certain nombre d'années en commande optimale ainsi que le montrent les exemples rapportés dans plusieurs ouvrages (2, 3, 8).

Nous allons donc rappeler brièvement "Le principe d'optimalité" qui est à l'origine de la programmation dynamique et l'appliquer à notre problème particulier.

Ce principe est interprété de deux façons différentes :

- la forme discrète
- la forme continue.

Nous allons voir que la forme discrète conduit à deux modes de résolution différents suivant que la discrétisation est imposée ou optimisée alors que la forme continue ne peut donner de résultats dans notre cas particulier.

II.1 Principe d'optimalité

La programmation dynamique est une méthode d'optimisation s'appliquant aux processus à décisions séquentielles et par extension à tous les problèmes d'optimisation pouvant être mis sous cette forme. Cette technique est déduite du principe d'optimalité énoncé par Bellman (1).

Nous donnons ici deux versions du principe d'optimalité, le premier est l'énoncé classique de Bellman R., l'autre (3) est plus abstrait mais se prête mieux à la formulation mathématique.

- Premier énoncé

Une politique optimale est telle que, quels que soient l'état initial et la décision initiale, les décisions suivantes doivent constituer une politique optimale par rapport à l'état résultant de la première décision.

- Second énoncé

Considérons deux problèmes définis de la façon suivante :

$$\text{Problème } P_1 \quad \begin{cases} D_1 : \text{ensemble de données} \\ U_1 : \text{ensemble de décisions} \end{cases}$$

$$\text{Problème } P_2 \quad \begin{cases} D_2 : \text{ensemble de données} \\ U_2 : \text{ensemble de décisions} \end{cases}$$

U' : premières décisions

$$\text{Relation entre } P_1 \text{ et } P_2 \quad \begin{cases} U_2 \subset U_1 \text{ et } U' \cup U_2 = U_1 \\ D_1 \xrightarrow{U'} D_2 \end{cases}$$

Le principe d'optimalité s'énonce alors de la façon suivante :

Quelles que soient les données D_1 du problème P_1 et les premières décisions U', si l'ensemble des décisions U_1 est optimal alors il doit en être de même pour l'ensemble U_2 relatif au problème P_2.

II. 2 Application du principe d'optimalité à l'ajustement de fonctions

II. 2.1 Rappel du problème posé

Soit le système différentiel, noté sous forme vectorielle :

$$\frac{dx}{dt} = f\ (x, u, t)$$

x : vecteur d'état
u : vecteur des fonctions de commande
t : variable temps.

Nous nous proposons de déterminer u sur un intervalle fini de R $[T_0, T]$ de façon à minimiser la quantité $\int_{T\emptyset}^{T} q\ (x, u, t)\ dt$. Cette quantité est appelée en programmation dynamique : "La fonction de coût" du problème.

Dans notre étude q (x, u, t) sera de la forme suivante :

$$q\ (x, u, t) = \sum_{i=1}^{i=r} \alpha_i\ (x_i\ (t) - g_i\ (t))^2$$

x_i : $i^{ème}$ composante du vecteur d'état
g_i : fonction expérimentale correspondant à x_i
α_i : facteur de pondération
r : nombre de fonctions expérimentales connues.

Remarque :

q (x, u, t) n'est fonction de u que par l'intermédiaire des fonctions x_i (t). Nous verrons les conséquences de cette particularité de la fonction de coût lors de l'application de la forme continue du principe d'optimalité et au niveau de la résolution numérique du problème.

II. 2.2 Equation fonctionnelle fondamentale

Appliquons le principe d'optimalité à la recherche du vecteur de fonctions u^* (t) qui minimise la quantité $\int_{T\emptyset}^{T} q\ (x, u, t)\ dt$. Pour cela nous déterminons deux problèmes P_1 et P_2 qui satisfont aux conditions définies au paragraphe II. 1.

Problème P_1 $\left\{ \begin{array}{l} D_1 : x_0 \text{ vecteur d'état au temps } T_0 \\ U_1 : u(t) = u_1(t) \text{ vecteur de commande défini sur l'intervalle } [T_0, T[\end{array} \right\}$

Problème P_2 $\left\{ \begin{array}{l} D_2 : x \text{ vecteur d'état au temps } T_0 + \Delta t \\ \quad\quad \Delta t : \text{intervalle de temps fini} \\ U_2 : u(t) = u_2(t) \text{ vecteur de commande défini sur l'intervalle } [T_0 + \Delta t, T[\end{array} \right\}$

$U' \{ u(t) = u'(t)$ vecteur de commande défini sur $[T_0, T_0 + \Delta t[\}$.

Vérifions l'existence des relations suivantes entre les problèmes P_1 et P_2.

Soit $\left\{ \begin{array}{l} U_2 \subset U_1 \\ U' \cup U_2 = U_1 \\ D_1 \xrightarrow{U'} D_2 \end{array} \right.$

U_1 : choix du vecteur de commande sur $[T_0, T[$

U_2 : choix du vecteur de commande sur $[T_0 + \Delta t, T[$

U' : choix du vecteur de commande sur $[T_0, T_0 + \Delta t[$

$[T_0 + \Delta t, T[\subset [T_0, T[$

$[T_0, T_0 + \Delta t[\cup [T_0 + \Delta t, T] = [T_0 + \Delta t, T[$

les relations liant U_2, U_1 et U' sont ainsi vérifiées.

D'autre part, connaissant x_0 pour $t = T_0$ ainsi que le vecteur de commande u' sur $[T_0, T_0 + \Delta t[$, la résolution du système différentiel permet d'obtenir x pour $t = T_0 + \Delta t$; ce qui vérifie la dernière relation.

Nous pouvons établir les correspondances suivantes :

- D_1 correspond à x_0
- U' correspond au choix de $u(t)$ sur $[T_0, T_0 + \Delta t[$
- U_1 optimal se traduit par l'existence d'un vecteur de commande u_1 tel que :

$$\int_{T_0}^{T} q(x, u_1, t)\, dt = \min_u \left(\int_{T\emptyset}^{T} q(x, u, z)\, dz \right)$$

- U_2 optimal se traduit par l'existence d'un vecteur de commande u_2 tel que

$$\int_{T\emptyset+\Delta t}^{T} q\ (x, u_2, t)\ dt = \min_{u} \int_{T\emptyset+\Delta t}^{T} q\ (x, u, z)\ dz$$

le principe d'optimalité est équivalent à l'équation suivante :

$$\min_{\substack{u(t) \\ \text{défini sur } [\,T_0,\ T[}} \left(\int_{T\emptyset}^{T} q\ (x, u, t)dt \right) = \min_{\substack{u(t) \\ \text{défini sur} \\ [\,T_0,\ T_0+\Delta t[}} \left(\int_{T\emptyset}^{T\emptyset+\Delta t} q\ (x, u, z)\, dz + \min_{\substack{u(t) \\ \text{défini sur} \\ [\,T_0+\Delta t,\ T[}} \left(\int_{T\emptyset+\Delta t}^{T} q\ (x, u, z)\ dz \right) \right)$$

Notons $S\ (x_0,\ T_0)$ la quantité : $\min_{\substack{u(t) \\ \text{défini sur} \\ [\,T_0,\ T[}} \int_{T\emptyset}^{T} q\ (x, u, z)\ dz$

et posons $x\ (T_0 + \Delta t) = x_0 + \Delta x$

l'équation devient alors :

$$S\ (x_0, T_0) = \min_{\substack{u(t) \\ \text{défini sur} \\ [\,T_0,\ T_0+\Delta t[}} \left(\int_{T\emptyset}^{T\emptyset+\Delta t} q\ (x, u, z)\ dz + S\ (x_0 + \Delta x,\ T_0 + \Delta t) \right)$$

le raisonnement que nous venons de faire pour T_0 est valable quel que soit t appartenant à l'intervalle $]\,T_0,\ T[$. Nous obtenons alors l'équation :

$$S\ (x, t) = \min_{\substack{u(t) \\ \text{défini sur} \\ [\,t,\ t+\Delta t[}} \left(\int_{t}^{t+\Delta t} q\ (x, u, z)\ dz + S\ (x + \Delta x,\ t + \Delta t) \right)$$

S (x, t) est appelée la fonction de coût optimal sur l'intervalle $[\,t,\ T[$.

Nous appellerons par la suite cette équation l'équation fonctionnelle du problème.

Nous allons voir comment cette équation se transforme et comment elle peut conduire à la solution du problème d'ajustement lorsque l'on utilise la forme discrète ou la forme continue de la programmation dynamique.

II. 3 Différentes utilisations du principe d'optimalité

II. 3. 1 Forme discrète

L'équation fonctionnelle que nous venons d'établir au paragraphe précédent nous suggère de remplacer la recherche du vecteur $u^{\pm}(t)$ qui minimise la quantité

$$\int_{T\emptyset}^{T} q\ (x, u, z)\ dz$$

sur l'intervalle $[\ T_0, T[$ par une succession de recherches sur des intervalles partiels. Pour cela deux possibilités s'offrent à nous :

- partager $[\ T_0, T[$ en N intervalles, égaux ou non, et remplacer sur chacun de ces intervalles les fonctions de commande par des constantes, en se donnant les bornes des différents intervalles.

- approcher comme dans le premier cas les fonctions de commande par des fonctions en escalier, mais rechercher quel est le meilleur découpage de l'intervalle d'étude en N intervalles partiels, au lieu de l'imposer.

Etudions tout d'abord le premier cas :

II. 3. 1. 1 Partage de l'intervalle d'étude imposé

Soient $t_0 = T_0,\ t_1,\ t_2 \ldots t_N = T$ les bornes des différents intervalles partiels, et $u_1,\ u_2 \ldots u_n$ les vecteurs remplaçant le vecteur u (t) sur ces intervalles avec la convention que u_k est le vecteur de commande sur l'intervalle $[\ t_{N-k},\ t_{N-k+1}[$.

Reprenons l'équation fonctionnelle établie au paragraphe II. 2. 2 :

$$S\ (x, t) = \min_{\substack{u(t)\ \text{défini} \\ \text{sur}\ [\ t,\ t+\Delta t[}} \left(\int_t^{t+\Delta} q\ (x, u, z)\ dz + S\ (x + \Delta x,\ t + \Delta t) \right)$$

avec

$$S\ (x, t) = \min_{\substack{u(t)\ \text{défini} \\ \text{sur}\ [\ t,\ T\ [}} \left(\int_t^{T} q\ (x, u, z)\ dz \right)$$

Nous allons discrétiser la fonction S (x, t) par rapport à la variable t en ne la calculant que pour les bornes des intervalles partiels, nous noterons $S\ (x, t_{N-k}) : S_k\ (x)$.

Nous remplaçons donc la détermination de la fonction S (x, t) par le calcul de la suite de fonctions S_1 (x), S_2 (x) ... S_N (x), ces fonctions étant liées entre elles par une relation de récurrence déduite de l'équation fonctionnelle.

En effet écrivons l'équation fonctionnelle pour :

- $t = t_{N-k}$
- $\Delta t = t_{N-k+1} - t_{N-k}$
- u (t) défini sur $[\, t, t+\Delta t\, [$ est, d'après nos notations, u_k, ce

qui donne :

$$S\ (x,\ t_{N-k}) = \min_{u_k} \left(\int_{t_{N-k}}^{t_{N-k+1}} q\ (x, u_k, t)\ dt + S\ (x + \Delta x,\ t_{N-k+1}) \right)$$

Soit

$$\boxed{S_k\ (x) = \min_{u_k} \left(\int_{t_{N-k}}^{t_{N-k+1}} q\ (x, u_k, t)\ dt + S_{k-1}\ (x+\Delta x) \right)}$$

le début de la récurrence est donné par :

$$S_1\ (x) = \min_{u_1} \left(\int_{t_{N-1}}^{T} q\ (x, u_1, t)\ dt \right)$$

Dans le second chapitre nous étudierons les différentes méthodes numériques permettant de calculer ces fonctions S_k (x) et d'obtenir les vecteurs U_k associés, à partir de cette relation de récurrence.

II. 3. 1. 2 Optimisation du partage de l'intervalle d'étude

Au paragraphe précédent nous nous sommes ramené à un problème discret en partageant l'intervalle d'étude $[\, T_0, T\,]$ en N intervalles partiels et en considérant les fonctions de commande comme des fonctions en escalier.

Une autre possibilité consiste maintenant à rechercher, parmi tous les découpages possibles de $[\, T_0, T\,]$, lequel est le meilleur, c'est-à-dire celui qui minimise la fonction de coût.

La quantité à minimiser devient alors :

$$\sum_{i=1}^{i=N} \left(\int_{t_{i-1}}^{t_i} q\ (x, u_i, \tau)\ d\tau \right) \quad \text{avec} \quad \begin{matrix} t_0 = T_0 \\ t_N = T \end{matrix}$$

le minimum est recherché sur l'ensemble des variables u_i et t_i (u_i étant un vecteur et t_i un scalaire).

Le temps devient donc une variable de commande particulière ce qui ne permet pas d'appliquer le principe d'optimalité de la même façon qu'au paragraphe II.2.2, nous allons donc établir une nouvelle équation fonctionnelle.

Introduisons la fonction $S_k\ (t, x_0)$ qui sera le coût minimum lorsque l'on partage $[\ T_0, T[$ en k intervalles partiels et lorsque $x\ (T_0) = x_0$ c'est-à-dire :

$$S_k\ (t, x_0) = \min_{\substack{u_1 \cdots u_k \\ t_1 \cdots t_{k-1}}} \left(\int_{T\emptyset}^{t_1} q\ (x, u_1, z)\ dz + \int_{t_1}^{t_2} q\ (x, u_2, z)\ dz + \ldots \int_{t_{k-1}}^{t} q\ (x, u_k, z) dz \right)$$

Ce problème, aller de T_0 à t en k étapes, peut se décomposer de la façon suivante : "aller de T_0 à t_{k-1} en k-1 étapes et aller de t_{k-1} à t en 1 étape" ; la décision initiale étant le choix de t_{k-1}. Le principe d'optimalité s'écrit alors :

$$S_k\ (t, x_0) = \min_{\substack{u_k \\ T_0 < t_{k-1} \leq t}} \left(S_{k-1}\ (t_{k-1}, x_0) + \int_{t_{k-1}}^{t} q\ (x, u_k, z)\ dz \right)$$

La fonction $S_{k-1}\ (t_{k-1}, x_0)$ ne dépend pas de u_k, ce qui nous donne :

$$S_k\ (t, x_0) = \min_{T\emptyset \leq t_{k-1} \leq t} \left(S_{k-1}\ (t_{k-1}, x_0) + \min_{u_k} \int_{t_{k-1}}^{t} q\ (x, u_k, z)\ dz \right)$$

La fonction $S_1\ (t, x_0)$, qui correspond au coût minimum lorsque l'on remplace chaque fonction de commande par une constante sur l'intervalle $[\ T_0, t[$ et lorsque $x\ (T_0) = x_0$, a la forme suivante :

$$S_1\ (t, x_0) = \min_{u_1} \left(\int_{T\emptyset}^{t} q\ (x, u_1, z)\ dz. \right)$$

Dans de nombreux problèmes d'ajustement x_0, vecteur d'état au temps T_0, est connu, on peut alors considérer les fonctions S_k comme des fonctions de t seul.

La relation de récurrence devient alors :

$$S_k(t) = \min_{T\emptyset \leq t_{k-1} \leq t} \left(S_{k-1}(t_{k-1}) + \min_{u_k} \int_{t_{k-1}}^{t} q(x, u_k, z)\, dz \right)$$

$$S_1(t) = \min_{u_1} \left(\int_{T\emptyset}^{t} q(x, u_1, z)\, dz \right)$$

II. 3. 1. 3 Comparaison des deux méthodes

Dans la première méthode, c'est-à-dire lorsque le partage de l'intervalle $[T_0, T[$ est imposé, l'équation fonctionnelle comprend la recherche d'un minimum à p variables, p étant le nombre de variables de commande, dans la seconde méthode nous sommes en présence d'une double minimisation à p+1 variables au total.

Cependant dans le premier cas, la fonction de coût optimal $S_k(x)$ est une fonction de n variables, n étant le nombre de variables d'état, tandis que dans le second, lorsque x_0 est connu, $S_k(t)$ est une fonction d'une variable. La résolution du problème d'ajustement, tel que nous le traitons, consiste à calculer successivement les fonctions de coût optimal S_k, ainsi, dès que l'ordre du système différentiel à ajuster sera supérieur à deux, et lorsque x_0 sera connu, la seconde méthode sera plus rapide que la première, que le calcul de S_k soit une simple tabulation ou une interpolation.

Pour cette raison, nous n'avons appliqué la méthode avec optimisation du partage qu'au cas d'un système différentiel. Nous verrons d'autres différences entre ces deux façons d'aborder le problème dans le chapitre III où nous traiterons la résolution numérique de ces deux types d'équations fonctionnelles.

II. 3. 2 Forme continue

II. 3. 2. 1 Rappels sur l'établissement de l'équation de Hamilton Jacobi*

Reprenons l'équation fonctionnelle établie au paragraphe II. 2. 2.

Soit :

$$S(x, t) = \min_{\substack{u(t) \text{ défini} \\ \text{sur } [t, t+\Delta t[}} \left(\int_t^{t+\Delta t} q(x, u, \tau)\, d\tau + S(x + \Delta x,\ t + \Delta t) \right)$$

Dans la version discrète du principe d'optimalité nous avons considéré Δt, comme un accroissement fini de la variable t, considérons le maintenant comme un infiniment petit, ce qui nous permet d'écrire :

$$S(x, t) = \min_{u(t)} (q(x, u, t)\, \Delta t + S(x, t) + \Delta S)$$

$$\Delta S = S(x + \Delta x,\ t + \Delta t) - S(x + t)$$

S (x, t) ne dépend pas de u (t) par définition, l'équation se réduit donc à :

$$0 = \min_{u(t)} (q(x, u, t)\, \Delta t + \Delta S)$$

Faisons tendre Δt vers zéro, ce qui donne :

$$0 = \min_{u(t)} \left(q(x, u, t) + \frac{dS}{dt}\right) \qquad (1)$$

S est une fonction de x et de t donc :

$$\frac{dS}{dt} = \frac{\partial S}{\partial t} + \sum_{i=1}^{n} \frac{\partial S}{\partial x_i} \frac{dx_i}{dt}$$

or x est solution du système différentiel : $\frac{dx}{dt} = f(x, u, t)$ cette notation vectorielle

* Ces rappels sont en grande partie empruntés à Naslin (8).

étant équivalente au système :

$$\begin{cases} \dfrac{dx_1}{dt} = f_1 \ (x_1 \dots x_n, \ u_1 \dots u_p, t) \\ \dfrac{dx_2}{dt} = f_2 \ (x_1 \dots x_n, \ u_1 \dots u_p, \ t) \\ \vdots \\ \dfrac{dx_n}{dt} = f_n \ (x_1 \dots x_n, \ u_1 \dots u_p, t \ ) \end{cases}$$

donc $\dfrac{dx_i}{dt} = f_i \ (x, u, t)$

l'équation ① devient

$$0 = \min_{u(t)} \ (q \ (x, u, t) + \frac{\partial S}{\partial t} + \sum_{i=1}^{i=n} f_i \ (x, u, t) \ \frac{\partial S}{\partial x_i} \ )$$

$\dfrac{\partial S}{\partial t}$ est indépendant de u, car S (x, t) l'est par définition, donc :

$$\frac{\partial S}{\partial t} = - \min_{u(t)} \left(q \ (x, u, t) + \sum_{i=1}^{i=n} f_i \ (x, u, t) \ \frac{\partial S}{\partial x_i} \right)$$

Notons $p_0 = \dfrac{\partial S}{\partial t} \qquad p_i = \dfrac{\partial S}{\partial x_i}$

Soit

$$p_0 = - \min_{u(t)} \left(q \ (x, u, t) + \sum_{i=1}^{n} p_i \ f_i \ (x, u, t) \right)$$

Cette équation définit le vecteur de commande optimal à l'instant t, que nous noterons $\bar{u}$. La fonction de coût optimal S (x, t) satisfait alors l'équation aux dérivées partielles suivante qui est appelée "équation de Hamilton-Jacobi" :

$$q \ (x, \bar{u}, t) + \sum_{i=1}^{i=n} p_i \ f_i \ (x, \bar{u}, t) + p_0 = 0$$

La somme $q \ (x, u, t) + \sum_{i=1}^{n} p_i \ f_i \ (x, u, t)$ est la fonction hamiltonienne que nous noterons H (x, u, t).

Le problème comporte donc deux phases :

- déterminer le vecteur de fonctions $\bar{u}$ (t) qui minimise la fonction H (x, u, t)
- obtenir la solution du système différentiel $\frac{dx}{dt} = f(x, \bar{u}, t)$ et la fonction de coût optimal associée.

Cette solution ainsi que les dérivées p_i de la fonction S (x, t) sont données par les "équations de Pontryagin", dont nous ne rappelerons pas ici l'établissement à partir de l'équation de Hamilton Jacobi.

Equations de Pontryagin

$$\begin{cases} \frac{dx_i}{dt} = \frac{\partial \bar{H}}{\partial p_i} & \text{pour } i = 1 \ldots n \qquad \text{avec } \bar{H} = H(x, \bar{u}, t) \\ \frac{dp_i}{dt} = \frac{\partial \bar{H}}{\partial x_i} & \text{pour } i = 1 \ldots n \end{cases}$$

L'application de cette méthode aux problèmes de commande optimale n'est possible que si la première phase du problème peut être résolue, c'est-à-dire si la commande optimale $\bar{u}$ (t) peut être obtenue analytiquement.

Nous allons voir que dans notre problème particulier la forme de la fonction hamiltonienne ne permet pas cette détermination de la fonction $\bar{u}$ (t).

II. 3. 2. 2 Manque d'intérêt de cette forme dans le cas étudié

Reprenons l'hamiltonien H (x, u, t)

$$H(x, u, t) = q(x, u, t) + \sum_{j=1}^{j=n} p_j f_j (x, u, t)$$

Nous avons remarqué au paragraphe II. 2. 1 que dans le problème d'ajustement que nous traitons q à la forme suivante :

$$q(x, t) = \sum_{j=1}^{j=r} \alpha_j (x_j(t) - g_j(t))^2 \qquad (r \leq n)$$

et ne dépend pas explicitement de u ; le vecteur optimal $\bar{u}$ minimisant H (x, u, t) vérifiera nécessairement le système d'équations

$\left(\frac{\partial H}{\partial u_i}\right) = 0$. Soit puisque p_i ne dépend pas de u également :

$$\left(\sum_{j=1}^{n} p_j \frac{\partial f_j}{\partial u_i} \right) = 0$$

ce qui peut encore s'écrire sous forme matricielle :

$$\begin{bmatrix} \frac{\partial f_1}{\partial u_1} \cdots \frac{\partial f_i}{\partial u_1} \cdots \frac{\partial f_n}{\partial u_1} \\ \vdots \\ \frac{\partial f_1}{\partial u_i} \cdots \frac{\partial f_i}{\partial u_i} \cdots \frac{\partial f_n}{\partial u_i} \\ \vdots \\ \frac{\partial f_1}{\partial u_n} \cdots \frac{\partial f_i}{\partial u_n} \cdots \frac{\partial f_n}{\partial u_n} \end{bmatrix} \quad \begin{bmatrix} p_1 \\ \vdots \\ p_i \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Une solution particulière de ce système est, quelques soient les fonctions f_i, $p_i = 0$ pour i = 1 à n.

Etudions cette solution :
les équations de Pontryagin s'écrivent alors :

$$\frac{dp_i}{dt} = \frac{\partial \bar{H}}{\partial x_i} \quad \text{pour } i = 1 \ldots n$$

$$p_i = 0 \quad \text{donc} \quad \frac{dp_i}{dt} = 0$$

or $\bar{H} = q\,(x, t) = \sum_{j=1}^{j=r} \alpha_j\,(x_j - g_j)^2$ soit $\frac{\partial \bar{H}}{\partial x_i} = 2\,\alpha_j\,(x_j - g_j)$

soit : $2\,\alpha_j\,(x_j - g_j) = 0$ pour $j = 1 \ldots r$.

Nous obtenons donc la solution optimale :

$$x_j\,(t) = g_j\,(t) \quad \text{pour} \quad j = 1 \ldots r.$$

la fonction de coût optimale, $S\,(x, t) = \min_u \int_t^T q\,(x, u, t)\,dt$ étant alors identiquement nulle.

Cette solution est effectivement la solution optimale du problème d'ajustement tel que nous l'avons défini au paragraphe I.2 : obtenir des solutions du système différentiel représentant le phénomène étudié, aussi proches que possible des fonctions connues expérimentalement, c'est-à-dire, à la limite, confondues.

La forme continue de la programmation dynamique nous a donc conduit à la solution optimale du problème mais sans déterminer les fonctions de commande associées dont la connaissance est le but réel de notre étude. Nous ne pourrons donc pas utiliser cette méthode dans les applications que nous étudierons aux chapitres suivants.

III - CONCLUSION

Cette étude théorique nous a montré qu'il est possible de considérer le problème d'ajustement de paramètres variables sur des systèmes différentiels comme un problème particulier de commande optimale.

En effet nous avons pu appliquer les différentes méthodes de la programmation dynamique ; cependant la trop grande simplicité de la fonction de coût à minimiser ne permet pas la détermination des paramètres inconnus, à l'aide de l'équation de Hamilton Jacobi et des équations de Pontryagin. Nous ne pourrons donc obtenir ces fonctions de commande que de façon approchée et discrète à l'aide des équations fonctionnelles établies à partir du principe d'optimalité.

Aux chapitres suivants nous allons donc étudier la résolution numérique des deux types d'équations fonctionnelles construites ci-dessus, tout d'abord sur une équation différentielle, afin de tester l'ensemble de la méthode et de déterminer les principales difficultés d'ordre numérique, puis sur un système différentiel d'ordre deux.

CHAPITRE II

AJUSTEMENT D'UNE FONCTION DE COMMANDE SUR UNE EQUATION DIFFERENTIELLE QUELCONQUE

Dans ce chapitre, nous allons tout d'abord étudier les différents problèmes numériques posés par la résolution d'une équation fonctionnelle du type de celles que nous avons établies au chapitre précédent.

Ensuite nous appliquerons notre méthode à l'ajustement d'un paramètre variable sur une équation différentielle d'un type simple. Cette étude aura un double but :

- vérifier que la méthode d'ajustement utilisant la programmation dynamique discrète conduit à la solution optimale.

- tester le programme de résolution numérique sur une équation fonctionnelle simple avant de l'appliquer à l'ajustement sur un système différentiel.

Enfin nous donnerons quelques résultats numériques obtenus sur trois équations différentielles différentes.

I - METHODE GENERALE DE RESOLUTION - PROBLEMES POSES

I-1 Résolution d'une équation fonctionnelle - Obtention de la solution

Soit un problème de commande optimale du type de ceux que nous avons vus au chapitre précédent :

- Déterminer le vecteur de commande u sur l'intervalle fini $[T_0, T]$ tel que la quantité $\int_{T_0}^{T} q\,(y, u, t)\,dt$ soit minimale ; y étant solution du système différentiel :

$$\frac{dy}{dt} = f\,(y, u, t).$$

Nous avons vu que la forme discrète du principe d'optimalité conduit, dans le cas où le partage de l'intervalle d'étude est imposé, à l'équation fonctionnelle suivante :

$$S_k\ (y) = \min_{u_k} \left(\int_{t_{N-k}}^{t_{N-k+1}} q\ (y, u_k, t)\ dt + S_{k-1}\ (y + \Delta y) \right)$$

Avec

$$S_1\ (y) = \min_{u_1} \left(\int_{t_{N-1}}^{T} q\ (y, u_1, t)\ dt \right)$$

N : nombre d'étapes

$t_0 = T_0, t_1, t_2 \cdots t_N = T$ bornes des différents intervalles partiels.

Soit y_0 le vecteur d'état au temps T_0

y_0 sera : - soit imposé

- soit calculé comme étant le vecteur initial optimal, c'est-à-dire tel que :

$$S_N\ (y_0) = \min_y\ (S_N(y))$$

Par définition nous avons :

$$S_N\ (y) = \min_u \left(\int_{T_0}^{T} q\ (y, u, t)\ dt \right)$$

La solution du problème est donc donnée par la connaissance de $S_N\ (y_0)$.

Mais pour obtenir $S_N\ (y)$ nous devons calculer successivement les fonctions $S_k\ (y)$ pour les valeurs de k allant de 1 à N-1. A chaque fonction $S_k\ (y)$ est associé un vecteur de fonctions de commande, que nous noterons $u_k^*\ (y)$, faisant correspondre à chaque valeur de y le vecteur de commande optimal sur l'intervalle $[\, t_{N-k},\ t_{N-k+1}[$.

Connaissant ces fonctions $S_k\ (y)$ et $u_k^*\ (y)$ la solution sera obtenue de la façon suivante :

La résolution du système différentiel $\frac{dy}{dt} = f\ (y, u_N^*\ (y_0), t)$ aux conditions initiales $y = y_0$ pour $t = T_0$, sur l'intervalle $[\, T_0, t_1]$ donnera la valeur

y_1 pour $t = t_1$; de même la résolution du système $\frac{dy}{dt} = f(y, u^*_{N-1}(y_1), t)$ sur $[t_1, t_2]$ avec $y = y_1$ pour $t = t_1$ conduira à une valeur y_2 pour $t = t_2$ et en procédant ainsi jusqu'à la résolution sur $[t_{N-1}, T[$ nous pourrons mettre la solution dans le tableau suivant :

t	Y	u
T_0	Y_0	$u^*_N(Y_0)$
t_1	Y_1	$u_{N-1}(Y_1)$
t_2	Y_2	⋮
⋮	⋮	⋮
t_{N-1}	Y_{N-1}	$u^*_2(Y_{N-1})$
T	Y_N	$u^*_1(Y_N)$

Cette méthode de résolution met en évidence les problèmes qui vont se poser lors d'une recherche de cette solution par les techniques de l'analyse numérique.

I-2 Principales phases de calcul

L'obtention de la solution, d'après l'étude précédente, suppose la connaissance des fonctions $S_k(y)$ et $u^*_k(y)$; c'est-à-dire, si N est le nombre d'étapes et p le nombre de fonctions de commande, de Np fonctions.

Ces fonctions ne pourront pas, en général, être connues sous forme analytique, le premier problème posé sera donc leur stockage en mémoire.

Deux solutions peuvent être envisagées :

- connaissance discrète sous forme d'un tableau de nombre
- connaissance sur un intervalle en utilisant une interpolation ou une approximation.

Intéressons nous maintenant à l'obtention du vecteur $u_k^{\star}(y_i)$, y_i étant une des valeurs de y pour laquelle S_k est calculée.

$u_k^{\star}(y_i)$ est le vecteur qui minimise la fonction de u_k suivante :

$$h(u_k) = \int_{t_{N-k}}^{t_{N-k+1}} q(y, u_k, t)\, dt + S_{k-1}(y_i + \Delta y_i)$$

où y est la solution du système différentiel :

$$\frac{dy}{dt} = f(y, u_k, t)$$

aux conditions initiales $y = y_i$ pour $t = t_{N-k}$.

Le minimum de $h(u_k)$ sera recherché par une méthode numérique d'optimisation. Toutes les méthodes numériques d'optimisation sont fondées sur le calcul de la fonction à optimiser en des points particuliers dont le choix dépend de la technique utilisée.

Voyons donc maintenant comment obtenir une valeur de la fonction $h(u_k)$ pour $u_k = u_{k_1}$.

$h(u_{k_1})$ est la somme d'une intégrale et d'une valeur de S_{k-1}.

- L'intégrale $\int_{t_{N-k}}^{t_{N-k+1}} q(y, u_{k_1}, t)\, dt$ sera obtenue par une méthode d'intégration numérique, la plus simple est la méthode du trapèze qui revient à remplacer l'intégrale par la quantité suivante :

$$(q(y_i, u_{k_1}, t_{N-k}) + q(y_i + \Delta y_i, u_{k_1}, t_{N-k+1}))(t_{N-k+1} - t_{N-k})/2$$

Δy_i sera obtenu en résolvant le système différentiel $\frac{dy}{dt} = f(y, u_{k_1}, t)$ sur l'intervalle (t_{N-k}, t_{N-k+1}) avec comme condition initiale : $y = y_i$ pour $t = t_{N-k}$; nous aurons donc besoin à ce niveau d'une méthode de résolution numérique de systèmes différentiels.

- l'utilisation d'une autre méthode que la méthode du trapèze pour le calcul de l'intégrale nécessitera la connaissance de y en quelques points de l'intervalle $[t_{N-k}, t_{N-k+1}[$, donc à nouveau la résolution du système différentiel.

D'autre part, nous n'aurons pas en général la valeur de la fonction S_{k-1} en mémoire pour $y = y_i + \Delta y_i$, nous remplacerons alors $S_{k-1}(y_i + \Delta y_i)$

- soit par $S_{k-1}(y_j)$ y_j étant, parmi les points pour lesquels nous avons précédemment calculé S_{k-1}, le plus proche de $(y_i + \Delta y_i)$
- soit par $W(y_i + \Delta y_i)$, W étant la fonction approchant S_{k-1}, déterminée à l'étape précédente.

Les méthodes de calcul numérique nécessaires à la résolution de l'équation fonctionnelle sont donc les suivantes :

- méthode d'optimisation à une ou plusieurs variables
- méthode d'intégration
- méthode de résolution de systèmes différentiels
- méthode d'interpolation à une ou plusieurs variables.

Cette énumération montre la complexité du programme à établir lors de problèmes d'ajustement de plusieurs fonctions de commande dans des systèmes différentiels, c'est pourquoi nous avons jugé nécessaire de tester la méthode de résolution dans le cas de l'ajustement d'une fonction de commande sur une équation différentielle.

Remarque

Nous n'avons présenté ici que la résolution du premier type d'équation fonctionnelle établie au chapitre précédent.

En effet le second type, lorsque le partage de l'intervalle d'étude est optimisé, se résoud d'une façon analogue.

Rappelons cette équation :

$$S_k(z) = \min_{T\emptyset \le t_k \le z} \left(S_{k-1}(t_k) + \min_{u_k} \int_{t_k}^{z} q(y, u_k, z)\, dz \right)$$

Les fonctions S_k seront calculées pour certaines valeurs de z et nous aurons les fonctions $t_k^{\pm}(z)$ et $u_k^{\pm}(z)$ associées. Le calcul d'une valeur $S_k(z_i)$ nécessitera, tout d'abord l'emploi d'une méthode d'optimisation à une variable, puis nous aurons les mêmes étapes numériques que dans le cas précédent. Ce n'est qu'au niveau de l'application à un exemple précis que nous verrons apparaître nettement les différences entre les méthodes de résolution des deux types d'équations fonctionnelles.

II - APPLICATION A L'AJUSTEMENT D'UNE FONCTION DE COMMANDE SUR UNE EQUATION DIFFERENTIELLE.

Considérons l'équation différentielle suivante :

$$y'(t) + u(t)\, y(t) = b(t) \qquad (1)$$

où - $b(t)$ est une fonction connue

- $u(t)$ est une fonction inconnue.

Soit $g(t)$ une fonction connue expérimentalement sur $[T_0, T]$ correspondant à la solution de l'équation (1) sur $[T_0, T]$.

On se propose de déterminer $u(t)$ de façon à minimiser l'écart entre la solution $y(t)$ de (1) et $g(t)$ sur l'intervalle $[T_0, T]$, nous prenons cette différence au sens des moindres carrés, c'est-à-dire que nous voulons minimiser la quantité :

$$\int_{T\emptyset}^{T} (y(t) - g(t))^2 \, dt.$$

Dans ce cas particulier, la solution du problème d'ajustement est immédiate, en effet, de l'équation ① nous pouvons tirer u (t), pour tout valeur de t où y (t) est différent de zéro :

$$u\ (t) = \frac{b\ (t) - y'\ (t)}{y\ (t)}$$

Soit en remplaçant y (t) par g (t)

$$u^{*}\ (t) = \frac{b\ (t) - g'\ (t)}{g\ (t)}$$

Nous allons voir que la résolution de l'équation fonctionnelle lorsque le partage de l'intervalle d'étude est imposé conduit à la limite, c'est-à-dire lorsque la longueur des intervalles partiels tend vers zéro, à cette solution optimale.

II - 1 Etablissement de l'équation fonctionnelle dans le cas du partage imposé.

Partageons l'intervalle $[T_0, T]$ en N intervalles partiels et appliquons les résultats du paragraphe II.3.1.1 du chapitre I.

L'équation fonctionnelle du problème posé ci-dessus est la suivante :

$$S_k\ (y) = \min_{u_k} \left(\int_{t_{N-k}}^{t_{N-k+1}} (y - g(t))^2\ dt + S_{k-1}\ (y + \Delta y) \right)$$

le début de la relation de récurrence étant donnée par :

$$S_1\ (y) = \min_{u_1} \left(\int_{t_{N-1}}^{T} (y - g\ (t))^2\ dt \right)$$

Rappelons la signification des grandeurs u_k :

- u_k est la valeur prise par la fonction en escalier approchant u (t), sur l'in-

tervalle $[t_{N-k}, t_{N-k+1}[$.

Nous allons voir les différentes méthodes permettant de résoudre cette équation.

II - 2 Différentes méthodes de résolution envisagées

Nous nous intéressons tout d'abord à la méthode de résolution la plus simple qui nous conduit à une solution explicite de l'équation fonctionnelle. Nous obtenons ainsi une justification de l'emploi de la programmation dynamique pour ces problèmes d'ajustement. Puis nous envisageons une méthode de résolution numérique plus complexe afin de la tester avant de l'appliquer à l'ajustement sur un système différentiel.

II. 2.1 Utilisation de la méthode du trapèze pour le calcul de l'intégrale et de la méthode d'Euler pour la résolution de l'équation différentielle : Solution explicite.

Nous avons commencé l'étude de la résolution de l'équation fonctionnelle en utilisant des méthodes numériques simples et nous allons voir que dans le cas d'une équation différentielle, ces méthodes conduisent directement à la solution optimale.

Reprenons l'équation fonctionnelle établie ci-dessus

$$S_k(y) = \min_{u_k} \left(\int_{t_{N-k}}^{t_{N-k+1}} (y - g(t))^2 dt + S_{k-1}(y + \Delta y) \right)$$

L'intégrale est calculée par la méthode du trapèze, c'est-à-dire qu'elle est remplacée par la quantité :

$$\frac{(t_{N-k+1} - t_{N-k})}{2} \left((y(t_{N-k+1}) - g(t_{N-k+1}))^2 + (y(t_{N-k}) - f(t_{N-k}))^2 \right)$$

avec $y\ (t_{N-k+1}) = y\ (t_{N-k}) + \Delta y$.

L'équation différentielle est résolue par la méthode d'Euler, c'est-à-dire que la dérivée y' est remplacée par le rapport des accroissements respectifs de y et de t.

$$\text{Soit} \quad \frac{\Delta y}{\Delta t} = - u\ (t_{N-k})\ y\ (t_{N-k}) + b\ (t_{N-k})$$

$$\text{avec} \quad \Delta t = (t_{N-k+1} - t_{N-k})$$

L'équation fonctionnelle devient dans ces conditions :

$$S_k\ (y) = \min_{u_k} \left(\frac{\Delta t}{2} ((y - g\ (t_{N-k}))^2 + (y + \Delta y - g\ (t_{N-k+1}))^2) + S_{k-1}\ (y + \Delta y) \right)$$

$$\text{avec} \quad \Delta y = (-u_k\ y + b\ (t_{N-k}))\ \Delta t$$

$$\text{et} \quad \Delta t = (t_{N-k+1} - t_{N-k})$$

Pour obtenir la solution du problème nous devons calculer successivement les fonctions $S_1,\ S_2 \ldots S_N$.

Intéressons nous tout d'abord au calcul de S_1

$$S_1\ (y) = \min_{u_1} \left(\frac{(T - t_{N-1})}{2} ((y - g\ (t_{N-1}))^2 + (y + \Delta y - g\ (T))^2 \right)$$

$$\text{avec} \quad \Delta y = (-u_1\ y + b\ (t_{N-1}))\ (T - t_{N-1})$$

$S_1\ (y)$ est le minimum d'une fonction de u_1 qui est la somme de deux carrés u_1 n'intervient que dans le second carré, le minimum est donc atteint pour la valeur $u_1^*\ (y)$ annulant le second carré soit :

$$y + (T - t_{N-1})\ (-u_1^*\ (y)\ y + b\ (t_{N-1})) - g\ (T) = 0$$

Soit

$$u_1^*\ (y) = \frac{- g\ (T) + y + b\ (t_{N-1})\ (T - t_{N-1})}{y\ (T - t_{N-1})}$$

$S_1(y)$ vaut alors :

$$S_1(y) = \left(\frac{T - t_{N-1}}{2}\right)(y - g(t_{N-1}))^2$$

Calculons $S_2(y)$

$$S_2(y) = \min_{u_2}\left[\frac{(t_{N-1} - t_{N-2})}{2}\left((y - g(t_{N-2}))^2 + (y + \Delta y - g(t_{N-1}))^2\right) + S_1(y+\Delta y)\right]$$

or d'après le calcul précédent nous avons

$$S_1(y + \Delta y) = (y + \Delta y - g(t_{N-1}))^2 \left(\frac{T - t_{N-1}}{2}\right)$$

ce qui donne

$$S_2(y) = \min_{u_2}\left[\frac{(t_{N-1} - t_{N-2})}{2}(y - g(t_{N-2}))^2 + (y + \Delta y - g(t_{N-1}))^2 \left(\frac{T - t_{N-2}}{2}\right)\right]$$

avec $\Delta y = (-u_2 y + b(t_{N-2}))(t_{N-1} - t_{N-2})$

En raisonnant comme pour S_1, le minimum de la fonction entre crochets est atteint pour la valeur $u_2^*(y)$ annulant le second carré, soit :

$$u_2^*(y) = \frac{-g(t_{N-1}) + y + b(t_{N-2})(t_{N-1} - t_{N-2})}{y(t_{N-1} - t_{N-2})}$$

Dans ce cas nous avons

$$S_1(y + \Delta y) = 0$$

et $$S_2(y) = (y - g(t_{N-2}))^2 (t_{N-1} - t_{N-2})/2$$

Poursuivons le calcul par récurrence

Supposons que jusqu'à l'étape k-1 nous avons l'ensemble de relations suivant :

a) $$u_i^*(y) = \frac{-g(t_{N-i+1}) + y + b(t_{N-i})(t_{N-i+1} - t_{N-i})}{y(t_{N-i+1} - t_{N-i})}$$

Pour i = 1, 2 ... (k-1)

b) $S_i(y) = 0$ Pour $i = 1, 2 \ldots (k-2)$

c) $S_{k-1}(y) = (t_{N-k+2} - t_{N-k+1})(y - g(t_{N-k+1}))^2/2$

Montrons que ces relations a), b), c) sont encore vérifiées à l'étape k.

Compte tenu de l'hypothèse c), $S_k(y)$ s'écrit :

$$S_k(y) = \min_{u_k} \left[\frac{(t_{N-k+1} - t_{N-k})}{2} (y - g(t_{N-k}))^2 + \frac{(t_{N-k+2} - t_{N-k})}{2} (y+\Delta y-g(t_{N-k+\right.$$

avec $\Delta y = (-u_k y + b(t_{N-k}))(t_{N-k+1} - t_{N-k})$

le même raisonnement que pour S_1 et S_2 nous donne

$$S_k(y) = (t_{N-k+1} - t_{N-k})(y - g(t_{N-k}))^2/2$$

$$u_k^*(y) = \frac{-g(t_{N-k+1}) + y + b(t_{N-k})(t_{N-k+1} - t_{N-k})}{y(t_{N-k+1} - t_{N-k})}$$

et $S_{k-1}(y) = 0$

ces trois équations sont les relations a), b), et c) pour l'étape k, elles ont été démontrées aux étapes 1 et 2, elles sont donc vérifiées pour toute valeur de k, entier appartenant à $[1, N]$.

La solution du problème est donc la suivante :

$$S_k(y) = 0 \quad \text{pour} \quad k = 1 \ldots N$$

$$y(t_k) = g(t_k) \quad \text{pour} \quad k = 0 \ldots N$$

$$u(t) = \frac{g(t_k) - g(t_{k+1})}{(t_{k+1} - t_k) g(t_k)} + \frac{b(t_k)}{g(t_k)} \quad \text{pour} \quad t \in [t_k, t_{k+1}[$$

Remarque 1

Remarquons que lorsque $(t_{k-1} - t_k)$ tend vers zéro $\frac{g(t_{k+1}) - g(t_k)}{(t_{k+1} - t_k)}$ tend vers $g'(t_k)$

nous retrouvons donc à la limite la valeur de u (t) établie en introduction du paragraphe II soit :

$$u(t) = (b(t) - g'(t)) / g(t)$$

Nous venons ainsi, en retrouvant ce résultat, de justifier l'emploi de la programmation dynamique discrète pour les problèmes d'ajustement ; nous avions déjà donné une première vérification en montrant que la forme continue du principe d'optimalité conduit à la solution optimale du problème, ce calcul en fournit une seconde.

Remarque 2

Envisageons maintenant une application de cette méthode, où nous ne connaissons la fonction expérimentale g (t) que de façon discrète c'est-à-dire en des points fixes t_0, t_1, ... t_N de l'intervalle $[T_0, T]$. Si ces points sont assez espacés, l'erreur faite en calculant à chaque étape l'accroissement Δy par la méthode d'Euler peut être amoindrie en utilisant une méthode de résolution de l'équation différentielle plus élaborée. Nous allons voir, dans le paragraphe suivant, que l'obtention de la solution devient alors plus compliquée ; mais nous avons tenu à étudier et à programmer cette nouvelle méthode de résolution de l'équation fonctionnelle, car elle est un cas particulier de la méthode générale que nous utiliserons sur un système différentiel.

II-2.2 Résolution de l'équation différentielle par la méthode de Kutta - Runge

Pour les raisons que nous venons d'exposer, nous nous intéressons maintenant à la résolution de l'équation fonctionnelle lorsque l'accroissement Δy est calculé par une méthode de résolution d'équations différentielles plus élaborée que la méthode d'Euler.

Nous avons choisi la méthode de Kutta - Runge à quatre approximations, technique classique de grande efficacité, dont nous ne rappellerons ici que quelques notions pratiques (6).

Pour une équation différentielle notée : $y' = f(t ; y)$, les formules de Kutta - Runge donnant $\Delta y = y(t + \Delta t) - y(t)$ sont les suivantes :

$$k_1 = \Delta t \, f(t ; y(t))$$

$$k_2 = \Delta t \, f\left(t + \frac{\Delta t}{2} ; y(t) + k_{1/2}\right)$$

$$k_3 = \Delta t \, f\left(t + \frac{\Delta t}{2} ; y(t) + k_{2/2}\right)$$

$$k_4 = \Delta t \, f(t + \Delta t ; y(t) + k_3)$$

$$\Delta y = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Dans l'exemple que nous avons choisi c'est-à-dire :

$$y' = b(t) - u(t)\, y$$

lorsque la fonction $u(t)$ est approchée par une constante u sur l'intervalle $[t, t + \Delta t[$, Δy devient un polynôme de degré quatre en u. Il n'est donc plus possible dans ce cas, et à plus forte raison lors d'ajustement sur des équations différentielles plus complexes que celle que nous avons choisie, de rechercher le minimum de la fonction de coût de façon analytique, nous utiliserons alors une méthode numérique de minimisation à une variable.

II. 2. 2. 1 Méthode d'optimisation à une variable

Nous avons utilisé la méthode de Fibónacci (4) qui est une méthode "minimax" donnant l'optimum d'une fonction d'une variable réelle sur un intervalle [a, b] de R lorsque l'on s'impose le nombre n de points, en lesquels on calcule la fonction à optimiser, et la séparation ε. Cette méthode présente l'avantage par rapport aux autres méthodes d'optimisation à une variable, de permettre de calculer le nombre minimum de points n, en lesquels il est nécessaire de calculer la fonction pour obtenir l'optimum.

La fonction que nous avons à minimiser est très complexe puisque la connaissance de l'une de ses valeurs nécessite la résolution d'une équation différentielle, aussi est-il intéressant de la calculer le moins de fois possible.

II. 2. 2. 2 Dénombrement sur les valeurs prises par la fonction d'état

Reprenons l'équation fonctionnelle établie au paragraphe II. 2. 1

$$S_k(y) = \min_{u_k} \left[\frac{(t_{N-k+1} - t_{N-k})}{2} \left((y - g(t_{N-k}))^2 + (y + \Delta y - g(t_{N-k+1}))^2 \right) + S_{k-1}(y + \Delta y) \right]$$

l'utilisation d'une méthode numérique pour la détermination du minimum de la fonction "entre crochets" ne permet d'obtenir qu'une connaissance discrète des fonctions $S_k(y)$ et $u_k^*(y)$.

Chaque fonction S_k sera donc calculée pour un certain nombre de valeurs de y. Nous aurons donc à conserver à la fois les valeurs $S_k(y)$ et $u_k^*(y)$ correspondantes. Dans le calcul de $S_k(y_i)$, nous remplacerons $S_{k-1}(y_i + \Delta y_i)$

par $S_{k-1}(y_j)$ où y_j est la valeur de la fonction d'état la plus proche de $(y_i + \Delta y_i)$ pour laquelle nous avons calculé S_{k-1}.

Cette méthode élémentaire a pu être employée, car par la nature même du problème nous connaissons le domaine où nous devons choisir les valeurs de y à chaque étape. En effet, notre but étant de trouver une solution de l'équation différentielle aussi proche que possible de la fonction expérimentale g, nous prenons à la $k^{ième}$ étape des valeurs de y situées dans le voisinage de la valeur $g(t_{N-k})$.

Le nombre de points utilisés ainsi que leur répartition autour de cette valeur dépendra de chaque cas particulier.

II. 2. 2. 3 Utilisation de l'interpolation

Cette façon de résoudre l'équation fonctionnelle par dénombrement sur la fonction d'état est la "méthode exhaustive" utilisée par Bellman (1) en programmation dynamique discrète. Lors d'une résolution à l'aide d'un ordinateur ce procédé présente l'inconvénient de nécessiter la mise en mémoire d'un grand nombre de valeurs pour chaque fonction S_k et d'entraîner une erreur systématique en remplaçant $(y_i + \Delta y_i)$ par une valeur y_j approchée.

Comme nous l'avons vu au début de ce chapitre, nous pouvons gagner à la fois de la place en mémoire et de la précision en calculant chaque fonction S_k en un petit nombre de points et en l'approchant par une fonction d'un type donné en utilisant une technique d'interpolation.

Les fonctions S_k étant ainsi calculées, nous avons deux possibilités pour déterminer la fonction de commande optimale :

- la première consiste à conserver en mémoire toutes les valeurs optimales u_k^i associées aux points y_i^k et de donner la solution en interpolant également les fonctions de commandes optimales.
- La seconde consiste à ne garder que les fonctions approchant les fonctions S_k et à recalculer les valeurs optimales u_k^* à partir de la solution optimale y_k^* ce qui nécessite à nouveau N recherches de minimum.

Ce second procédé nécessite un temps de calcul plus long que le précédent mais permet d'obtenir la fonction de commande optimale avec plus de précision, surtout lorsque la méthode d'interpolation utilise peu de points.

Nous allons voir maintenant quelques applications numériques de ces trois méthodes :

- solution explicite
- méthode avec dénombrement
- méthode avec interpolation.

III - RESULTATS NUMERIQUES

III - 1 Premier exemple

Nous avons tout d'abord appliqué la méthode à une équation différentielle simple avec une fonction de commande simple également.

Nous avons choisi l'équation :

$$y' + \frac{t}{100}\, y = \frac{2t}{100}$$

avec la solution $g(t) = 20 \exp(-t^2/200) + 2$

et nous avons cherché la fonction $u(t)$ telle que la solution de l'équation différentielle $y' + u(t)\, y = \frac{2t}{100}$ soit aussi proche que possible de la fonction $g(t)$.

Nous avons appliqué successivement les trois méthodes envisagées ci-dessus.

III.1.1 Solution explicite

Nous avons ajusté la fonction $u(t)$ sur l'intervalle $[0, 5]$ partagé en 50 intervalles égaux.

Notons u_i la valeur approchée de la fonction de commande sur l'intervalle $[t_i, t_{i+1}]$.

Nous avons obtenu en moyenne sur cet intervalle une valeur u_i égale à la valeur exacte de la fonction $u(t)$ au milieu de l'intervalle, c'est-à-dire $u((t_{i+1} + t_i)/2)$, à 10^{-3} près en valeur relative. La précision est meilleure sur la première partie de l'intervalle c'est-à-dire pour $t \leq 2$ que sur la seconde, ceci s'expliquant par la décroissance exponentielle de $g(t)$.

III. 1. 2 Méthode avec dénombrement

Nous avons appliqué la méthode avec dénombrement sur l'intervalle [1, 2]. Partagé en 10 intervalles égaux et nous avons calculé chaque fonction S_k (y) en 10 points. Nous avons obtenu les résultats suivants :

$\frac{t_{i+1} + t_i}{2}$	u_i
1.05	0.01054
1.15	0.01127
1.25	0.01273
1.35	0.01345
1.45	0.01455
1.55	0.01527
1.65	0.01673
1.75	0.01745
1.85	0.01854
1.95	0.01927

la valeur approchée u_i est égale à la valeur exacte au milieu de l'intervalle à 10^{-2} près en valeur relative ;

le coût total S_N (y) est alors égal à 10^{-7}

l'obtention de cette solution nécessite la mise en mémoire de 300 nombres ; en effet il y a 10 étapes, à chaque étape on calcule 10 fois la fonction S_k et pour chaque calcul on a trois grandeurs à conserver :

- la valeur de S_k
- la valeur de u_k associée
- la valeur de y à l'étape précédente.

III. 1. 3 Méthode avec interpolation

Nous avons utilisé une méthode d'interpolation très simple soit l'interpolation d'une fonction sur trois points par un polynôme de degré deux ; à l'étape k les points d'interpolation sont les suivants :

$$y_1^k = g\,(t_{N-k})\,(1 - h)$$

$$y_2^k = g\,(t_{N-k}) \qquad \text{avec } 0 < h < 1$$

$$y_3^k = g\,(t_{N-k})\,(1 + h)$$

La méthode d'interpolation étant très élémentaire, nous n'avons pas interpolé les fonctions de commande mais nous les avons recherchées à partir de la connaissance des fonctions S_k.

Nous avons appliqué cette méthode comme précédemment sur l'intervalle $[1, 2]$ partagé en 10 intervalles égaux ; nous avons obtenu des résultats comparables aux précédents, c'est-à-dire $u_i = u\,\frac{(t_{i+1} + t_i)}{2}$ à 10^{-3} près en valeur relative mais avec un coût total de 10^{-8}.

L'obtention de la solution ne nécessite cette fois que la mise en mémoire de 30 nombres, soit 10 fois moins que précédemment, ce qui nous a invité à affiner les résultats obtenus en prenant des intervalles partiels de longueur $\Delta t = 0{,}05$ au lieu de 0.1

les résultats obtenus sont alors les suivants :

$\frac{t_{i+1} - t_i}{2}$	u_i
1.525	0.015249
1.575	0.015751
1.625	0.016248
1.675	0.016750
1.725	0.017250
1.775	0.017750
1.825	0.018249
1.925	0.019249
1.975	0.019751

coût total $= 10^{-13}$

$u_i = u \frac{(t_{i+1} + t_i)}{2}$ à 10^{-4} près en valeur relative

III.1.4 Conclusion

Au vu de ce premier exemple, la méthode utilisant l'interpolation apparaît comme la plus performante, en ce qui concerne, tant la qualité des résultats que l'encombrement en mémoire.

La méthode discrète pourrait également conduire à d'excellents résultats en diminuant la longueur des intervalles de temps Δt, mais nous n'avons pas jugé utile de poursuivre des essais dans cette voie car cette méthode ne se généralise pas à l'ajustement sur un système.

Nous n'avons donc appliqué que la méthode avec interpolation aux deux autres exemples que nous avons traités.

III - 2 Second exemple

Pour le second exemple nous avons pris également une équation différentielle linéaire à coefficients variables mais avec une fonction de commande u (t) plus compliquée que précédemment.

Soit l'équation :

$$y' + u(t)\, y = 0 \qquad \text{avec } u(t) = t^2 - 3t$$

et avec la solution $g(t) = \frac{1}{2} \exp\left(-\frac{t^3}{3} + \frac{3t^2}{2}\right)$.

Nous avons ajusté $u(t)$ sur l'intervalle $[0, 2]$ en prenant un pas $\Delta t = 0,1$ sur l'intervalle $[0,1]$ et un pas $0,05$ sur l'intervalle $[1, 2]$.

Nous avons obtenu en moyenne $u_i = u\,\frac{(t_{i+1} + t_i)}{2}$ à 10^{-3} près en valeur relative avec un coût total de 10^{-5}.

Le choix du pas Δt plus petit sur l'intervalle $[1, 2]$ que sur l'intervalle $[0,1]$ est du à la variation de la fonction $g(t)$.

III - 3 Troisième exemple

Comme troisième exemple d'application nous avons choisi une équation non linéaire avec une variable de commande fonction de y et non plus de t.

Soit l'équation

$$t\, y' - y\, u(y) = 0$$

avec $u(y) = y - 1$ et la solution $g(t) = \frac{i}{1-2t}$ pour $t > 0.5$.

Sur l'intervalle $[2.4,\ 2.5]$ avec un pas $\Delta t = 0.01$ nous avons obtenu $u_i = u\,\frac{(t_{i+1} + t_i)}{2}$ à 10^{-2} près en valeur relative avec un coût total de 10^{-9}.

III - 4 Remarque

Nous n'avons pas précisé sur les exemples traités les valeurs données aux différents paramètres dont dépend la mise en œuvre pratique du programme de résolution , à savoir le pas d'interpolation h, l'intervalle $[A, B]$ sur lequel on applique la méthode de Fibonacci ainsi que la précision

demandée lors de l'emploi de cette méthode. En effet le choix de ces grandeurs dépend de chaque application et ne peut se faire qu'expérimentalement. Il sera donc en général nécessaire de faire une simulation pour obtenir des ordres de grandeur de la fonction de commande avant d'envisager la résolution par la méthode que nous venons de proposer.

Nous verrons que dans l'ajustement sur un système nous ne pourrons pas nous passer de cette phase préliminaire de simulation, car les paramètres dont dépendra le programme seront plus nombreux et le temps de résolution du problème beaucoup plus long.

IV - CONCLUSION

Les résultats obtenus sur les exemples traités nous ont semblé assez satisfaisants pour envisager l'extension de la méthode à l'ajustement de plusieurs fonctions de commande sur des systèmes différentiels. Cependant, de même que nous avons commencé l'étude générale par le cas simple d'une équation différentielle, nous avons tout d'abord traité l'ajustement sur un système d'ordre deux ; nous n'avons pas pu, avec le matériel mis à notre disposition (CAE 510), effectuer ensuite des applications sur des systèmes d'ordre supérieur à deux, mais le passage d'un problème sur un système d'ordre deux à un problème sur un système d'ordre supérieur ne pose que des difficultés pratiques, alors que, comme nous allons le voir au chapitre suivant, la méthode employée sur une équation différentielle ne peut pas s'étendre directement à l'ajustement sur un système.

CHAPITRE III

AJUSTEMENT DE DEUX FONCTIONS DE COMMANDE SUR UN SYSTEME DIFFERENTIEL D'ORDRE DEUX

Dans le premier chapitre nous avons vu que le problème d'ajustement sur un système différentiel peut se ramener, en utilisant les techniques de la programmation dynamique, à la résolution d'une équation fonctionnelle ; dans le chapitre précédent nous avons étudié le cas simple de l'ajustement d'une fonction de commande sur une équation différentielle non linéaire et nous avons mis au point plusieurs techniques numériques pour déterminer la fonction inconnue ; nous nous proposons maintenant d'étendre ces méthodes à l'ajustement sur un système différentiel.

Cette extension ne peut pas se faire sans une nouvelle étude car il existe des différences fondamentales entre l'ajustement sur une équation et l'ajustement sur un système, nous allons mettre en évidence les principales.

- Dans les applications pratiques, nous ne connaitrons jamais autant de fonctions expérimentales que nous aurons de variables d'état, alors que dans le cas précédent nous avions une fonction expérimentale et une variable d'état.

- Sur des systèmes différentiels nous chercherons à déterminer plusieurs fonctions de commande. Le nombre de fonctions que nous pourrons ajuster dépendra de la quantité d'information que nous posséderons c'est-à-dire du nombre de fonctions expérimentales connues. Nous n'avons pas résolu ici le problème général de la détermination du nombre maximal de fonctions de commande que l'on peut ajuster à partir d'une quantité d'information donnée ; ceci dépasse le cadre de notre étude et nous n'avons travaillé que sur un système différentiel d'ordre deux.

- Ces deux points montrent que les techniques nécessaires à la résolution de l'équation fonctionnelle vont être différentes de celles utilisées

précédemment : nous aurons à rechercher le minimum d'une fonction de plusieurs variables et à mettre en mémoire des fonctions de coût optimal de plusieurs variables également.

Nous avons tout d'abord étudié la résolution de l'équation fonctionnelle dans le cas d'ajustement sur un système simple avant d'envisager une application plus générale.

Nous avons donc travaillé sur un système d'ordre deux de la forme suivante :

$$\begin{cases} \dfrac{dx_1}{dt} = a_1 x_1 + a_2 x_2 \\ \dfrac{dx_2}{dt} = a_3 x_1 + a_4 x_2 \end{cases}$$

où a_1, a_2, a_3, a_4 sont des fonctions de t ou de x_1.

- Nous étudierons deux cas :
  - nous connaissons une fonction expérimentale g_1 correspondant à x_1
  - nous connaissons une fonction expérimentale g_3 correspondant à une combinaison linéaire de x_1 et x_2.
- Dans chacun de ces cas nous ajusterons deux fonctions de commande u_1 et u_2 qui seront deux fonctions a_i inconnues, sur un intervalle fini $[T_0, T[$.
- Nous établirons des méthodes de résolution numérique des équations fonctionnelles. Pour les deux cas envisagés au chapitre I :
  - partage de l'intervalle d'étude imposé
  - partage de l'intervalle d'étude optimisé.
- Nous donnerons enfin quelques applications numériques, sur un système où les fonctions de commande sont des fonctions de t et sur un autre système où elles sont des fonctions de x_1.

I - METHODE DE RESOLUTION DANS LE CAS DU PARTAGE IMPOSE

I - 1 Système choisi et équation fonctionnelle associée

Soit le système différentiel :

$$\begin{cases} \dfrac{dx_1}{dt} = u_1 x_1 + a_2 x_2 \\ \dfrac{dx_2}{dt} = a_3 x_1 + u_2 x_2 \end{cases}$$

- a_2 et a_3 : fonctions connues
- u_1 et u_2 : fonctions de t ou de x_1 inconnues
- g_1 (t) fonction expérimentale connue sur l'intervalle $[T_0, T[$ correspondant à x_1.

Nous partageons $[T_0, T[$ en N intervalles et sur chacun de ces intervalles nous approchons les fonctions u_1 et u_2 par des constantes, que nous noterons u_1^k et u_2^k, sur l'intervalle $[t_{N-k}, t_{N-k+1}[$.

Nous ne précisons pas à ce niveau la fonction $q(x_1, x_2, u_1, u_2, t)$ dont nous cherchons à minimiser l'intégrale de T_0 à T, nous verrons en effet que sa forme dépend de la méthode employée pour la résolution de l'équation fonctionnelle.

D'après les résultats du chapitre I l'équation fonctionnelle du problème est la suivante :

$$S_k(x_1, x_2) = \min_{u_1^k, u_2^k} \left(\int_{t_{N-k}}^{t_{N-k+1}} q(x_1, x_2, u_1^k, u_2^k, t)\, dt + S_{k-1}(x_1+\Delta x_1, x_2+\Delta x_2) \right)$$

$$S_1(x_1, x_2) = \min_{u_1^1, u_2^1} \left(\int_{t_{N-1}}^{T} q(x_1, x_2, u_1^1, u_2^1, t)\, dt \right)$$

Nous allons voir que la façon la plus simple de résoudre l'équation fonctionnelle, que nous avions utilisée pour l'ajustement sur une équation, et que nous avions appelée solution explicite n'est pas applicable pour ce nouveau problème.

I - 2 Impossibilité d'obtenir une solution explicite

Rappelons que nous avons obtenu une solution explicite au chapitre précédent en résolvant le système différentiel par la méthode d'Euler et en calculant l'intégrale par la méthode du trapèze.

Choisissons, comme précédemment, la fonction q, c'est-à-dire

$$q\ (x_1,\ x_2,\ u_1^k,\ u_2^k, t) = (x_1\ (t) - g_1\ (t))^2$$

l'équation fonctionnelle s'écrit alors :

$$S_k\ (x_1, x_2) = \min_{u_1^k,\, u_2^k} \left(\frac{\Delta t}{2} \left((x_1 - g\ (t_{N-k}))^2 + (x_1 + \Delta x_1 - g\ (t_{N-k+1}))^2 \right) + S_{k-1}(x_1 + \Delta x_1, x_2 + \Delta x_2) \right)$$

avec

$$\Delta t = (t_{N-k+1} - t_{N-k})$$

$$\Delta x_1 = (u_1^k\ x_1 + a_2\ x_2)\ \Delta t$$

$$\Delta x_2 = (a_3\ x_1 + u_2^k\ x_2)\ \Delta t$$

et

$$S_1\ (x_1, x_2) = \min_{u_1^1,\, u_2^1} \left[\frac{(T - t_{N-1})}{2} \left((x_1 - (t_{N-1}))^2 + (x_1 + \Delta x_1 - g\ (T))^2 \right) \right]$$

avec $\Delta x_1 = (u_1^1\ x_1 + a_2\ x_2)\ (T - t_{N-1})$.

En raisonnant comme dans le paragraphe II. 2. 1 du précédent chapitre nous pouvons affirmer que le minimum de la fonction entre crochets est atteint si le second carré est nul, c'est-à-dire si

$$x_1 + (u_1^1 x_1 + a_2 x_2)(T - t_{N-1}) - g(T) = 0$$

Deux remarques s'imposent :

1) cette équation ne fait pas intervenir u_2^1

2) nous ne possédons qu'une relation pour déterminer deux inconnues u_1^{1*} et u_2^{1*}

Il y a donc une indétermination qui se répète de la même façon à la seconde étape et à toutes les autres jusqu'à la dernière. Nous avons fait alors la remarque suivante :

Un système différentiel est, lorsqu'il est linéaire, toujours équivalent à une équation différentielle, cette équivalence se présente aussi pour certaines formes de systèmes différentiels non linéaires.

Pour un système du second ordre l'équation, du second ordre, est du type $f(x''_1, x'_1, x_1, t) = 0$

cette équation est elle-même équivalente au système différentiel II suivant :

$$\text{II} \begin{cases} \dfrac{dx_1}{dt} = z \\ f\left(\dfrac{dz}{dt}, z, x_1, t\right) = 0 \end{cases}$$

Si on connait une fonction expérimentale $g_1(t)$, correspondant à $x_1(t)$ on peut obtenir également sa dérivée par rapport à t, $g'_1(t)$, ce qui revient à connaître deux fonctions expérimentales pour le système II et donc à minimiser dans un problème d'ajustement la somme suivante :

$$(g_1 - x_1)^2 + (g'_1 - z)^2.$$

Pour remédier à l'indétermination à laquelle nous aboutissons en minimisant uniquement la différence entre x_1 et g_1, nous avons décidé, sous l'influence de la remarque précédente, de prendre comme fonction de coût :

$$\int_{T\emptyset}^{T} (x_1(t) - g_1(t))^2 + \left(\frac{dx_1}{dt} - g'_1(t)\right)^2 \quad dt$$

L'équation fonctionnelle devient alors, en utilisant toujours la méthode du trapèze pour le calcul de l'intégrale :

$$S_k(x_1, x_2) = \min_{u_1^k, u_2^k} \left[\frac{\Delta t}{2} \left((x_1 - g(t_{N-k}))^2 + (u_1^k x_1 + a_2 x_2 - g'(t_{N-k}))^2 + \right.\right.$$

$$\left.\left. (x_1 + \Delta x_1 - g(t_{N-k+1}))^2 + (u_1^k (x_1 + \Delta x_1) + a_2 (x_2 + \Delta x_2) - g'(t_{N-k+1}))^2 \right) + \right.$$

$$\left. S_{k-1}(x_1 + \Delta x_1,\ x_2 + \Delta x_2) \right]$$

avec

$$S_1(x_1, x_2) = \min_{u_1^1, u_2^1} \left[\frac{\Delta t}{2} \left((x_1 - g(t_{N-1}))^2 + (u_1^1 x_1 + a_2 x_2 - g'(t_{N-1}))^2 + (x_1 + \Delta x_1 - g(T)) \right.\right.$$

$$\left.\left. + (u_1^1 (x_1 + \Delta x_1) + a_2 (x_2 + \Delta x_2) - g(T))^2 \right) \right]$$

Si on utilise comme précédemment la méthode d'Euler pour le calcul de Δx_1 et Δx_2, u_2^1 intervient explicitement dans la fonction à minimiser et dans ce cas on peut encore déterminer analytiquement les valeurs u_1^{1*} et u_2^{1*} correspondant au minimum, mais les calculs deviennent très compliqués dès la seconde étape et ne se généralisent pas simplement comme dans le cas d'ajustement sur une équation. Nous rechercherons donc le minimum par une méthode d'optimisation numérique.

Une autre possibilité pour éviter l'indétermination que nous avons en calculant l'intégrale par la méthode du trapèze est de la calculer par une méthode numérique d'intégration faisant intervenir la fonction $(x_1(t) - g_1(t))^2$ en plusieurs points de l'intervalle $[t_{N-k}, t_{N-k+1}[$. Dans ce cas encore nous ne pouvons pas espérer déterminer les valeurs u_1^{k*} et u_2^{k*} de façon analytique, nous ferons appel à une méthode d'optimisation numérique.

Le choix de la fonction de coût à minimiser dépendra de chaque application pratique ;

- Nous préférerons la première méthode lorsque la fonction expérimentale sera connue de façon discrète aux seuls points t_i et lorsque les dérivées en ces points pourront être obtenues sans entraîner une erreur trop importante sur le reste du calcul.
- La seconde méthode sera utilisée lorsque la fonction expérimentale sera connue
  - soit en tout point de l'intervalle $[T_0, T[$
  - soit de façon discrète mais assez précisément pour pouvoir être remplacée par une fonction d'interpolation.
- D'autre part le temps de calcul d'une valeur de la fonction à minimiser, à chaque étape, sera plus long avec la seconde méthode qu'avec la première car il nécessitera le calcul du vecteur d'état (x_1, x_2) en des points intermédiaires. Ce facteur pourra influencer également sur le choix de la fonction de coût.

Nous allons voir maintenant les méthodes d'analyse numérique que nous avons employées pour la résolution de l'équation fonctionnelle.

I - 3 Méthode du gradient

Nous avons utilisé la méthode du gradient comme méthode d'optimisation à plusieurs variables.

I. 3.1. Principe de la méthode

Notons f (x) la fonction dont nous cherchons le maximum (la recherche d'un minimum se déduit de celle d'un maximum par la relation : $\min_x f(x) = - \max_x (-f(x))$.

La méthode du gradient est une méthode itérative qui, à partir d'un point x_0, construit la suite de points x_0, $x_1 \ldots x_n$ tels que

$$f(x_0) < f(x_1) < \ldots f(x_i) < \ldots f(x_n)$$

La détermination d'un point x_i à partir du point x_{i-1} se fait de la façon suivante :

- x_i est le point correspondant au maximum de la fonction f sur la direction du gradient en x_{i-1}.

Notons α_{i-1} le vecteur des cosinus directeurs du gradient en x_{i-1} ; la recherche de x_i se ramène à la recherche du maximum de la fonction d'une variable r soit : $f(x_{i-1} + r\,\alpha_{i-1})$.

Soit r_m la valeur de r correspondant au maximum de cette fonction nous aurons $x_i = x_{i-1} + r_m\,\alpha_{i-1}$.

I. 3.2 Application à notre problème

La fonction que nous avons à minimiser à chaque étape, c'est à dire lors du calcul d'une fonction de coût optimal $S_k(x_1, x_2)$, est de la forme suivante :

$$h_k(u_1, u_2) = \int_{t_{N-k}}^{t_{N-k+1}} q(x_1, x_2, u_1, u_2)\, dt + S_{k-1}(x_1 + \Delta x_1, x_2 + \Delta x_2)$$

Cette fonction n'étant connue que de façon numérique, nous calculons le gradient en un point (u_1^k, u_2^k) de façon approchée en donnant des accroissements finis Δu_1^k et Δu_2^k aux deux variables.

D'autre part, la recherche du minimum sur la direction du gradient est faite également par une méthode numérique ; nous avons pour cela le choix entre deux possibilités :

- rechercher le minimum de façon grossière en ne calculant la fonction qu'en des points particuliers et conserver comme solution celui qui correspond à la valeur la plus faible.
- utiliser une méthode numérique d'optimisation à une variable.

Dans les deux cas nous ferons l'hypothèse d'unimodalité de la fonction à minimiser. Cette hypothèse est essentielle pour pouvoir appliquer une méthode "minimax", et n'entraîne pas d'erreur trop importante lorsqu'elle n'est pas vérifiée ; en effet, si nous n'obtenons qu'un minimum local sur la direction du gradient, nous ferons peut-être quelques itérations supplémentaires mais nous arriverons au même résultat final.

D'autre part l'application d'une méthode "minimax" nécessite la connaissance d'un intervalle sur lequel se trouve la valeur de la variable donnant le minimum de la fonction à minimiser.

Pour déterminer cet intervalle nous avons utilisé la méthode suivante : notons α_1^k et α_2^k les cosinus directeurs du gradient au point (u_1^k, u_2^k). Nous avons vu que minimiser la fonction h était équivalent à maximiser la fonction (-h), nous avons donc à rechercher le maximum de la fonction $\ell(r)$

$$\ell\ (r) = -\ h_k\ (u_1^k - r\ \alpha_1^k,\ u_2^k - r\ \alpha_2^k) \quad \text{pour } r \geq 0$$

Fixons nous deux nombres α et s, α étant supérieur à 1 et calculons successivement la fonction ℓ pour $r = s\ \alpha^p$, p étant un entier positif ou nul.

D'après l'hypothèse d'unimodalité de la fonction ℓ le maximum se trouvera sur l'intervalle :

- $[0, s]$ si $\ell\ (s) < \ell\ (0)$
- $[s\ \alpha^{p-1},\ s\ \alpha^{p+1}]$ si, pour $i = 1, 2 \ldots p$, $\ell\ (s\ \alpha^{i-1}) \leq \ell\ (s\ \alpha^i)$ et $\ell\ (s\ \alpha^{p+1}) < \ell\ (s\ \alpha^p)$.

- Dans le cas où nous nous contentons d'une estimation du minimum nous prenons comme nouveau point $(u_1,\ u_2)$ le point $(u_1^k - (s\ \alpha^p)\ \alpha_1^k,\ u_2^k - (s\alpha^p)\alpha_2^k)$; si $\ell\ (s)$ est inférieur à $\ell\ (0)$ nous recommençons la recherche avec une autre valeur $s_1 < s$.
- Pour obtenir le minimum avec plus de précision, nous recherchons le maximum de la fonction $\ell\ (r)$ sur l'intervalle déterminé ci-dessus par la méthode des parties proportionnelles qui est la méthode "minimax" la plus efficace et la plus économique d'après des études antérieures (4).

Nous obtiendrons ainsi une valeur r_m et le point suivant de l'itération sera $(u_1^k - r_m\ \alpha_1^k,\ u_2^k - r_m\ \alpha_2^k)$.

Pour commencer la recherche itérative du minimum de $h_k\ (u_1,\ u_2)$ nous prendrons à chaque étape comme point initial le point correspondant au minimum de h_{k-1} trouvé à l'étape précédente. Nous serons, de cette manière, assuré de débuter la recherche dans le voisinage de la solution et d'obtenir ainsi une convergence rapide. Nous n'aurons donc à nous fixer des valeurs initiales arbitraires que pour la première étape ; le choix de ces va-

leurs sera délicat lorsque nous n'aurons aucune précision sur l'ordre de grandeur des variables de commande à l'instant T.

Nous arrêterons les itérations lorsque les deux critères suivants seront vérifiés ensemble :

- les dérivées de la fonction h_k sont inférieures en valeur absolue à une quantité ε_1
- la distance séparant les deux derniers points est inférieure à une quantité ε_2.

La méthode du gradient, ainsi utilisée, nous a donné de bons résultats bien que n'étant pas, d'après des études antérieures (4) la meilleure méthode numérique d'optimisation à plusieurs variables. Ces résultats s'expliquent par le fait que nous avons toujours des valeurs initiales assez proches des valeurs optimales.

L'utilisation de la méthode des parties proportionnelles a donné de meilleurs résultats que la recherche simplifiée ; en effet cette dernière technique est moins précise et nécessite davantage de calculs de gradient ainsi le temps de calcul gagné lors de la recherche du minimum de la fonction d'une variable est perdu en calculs de gradient. D'autre part le choix des paramètres s et α a plus d'importance lors de la recherche simplifiée et, comme nous ne possédons que très peu d'indications pour les déterminer, nous avons préféré utiliser la méthode du gradient comprenant la méthode des parties proportionnelles.

Nous nous sommes attardé sur la méthode du gradient, car lors de la résolution de l'équation fonctionnelle, l'étape numérique la plus importante et aussi la plus délicate est l'obtention du minimum d'une fonction de

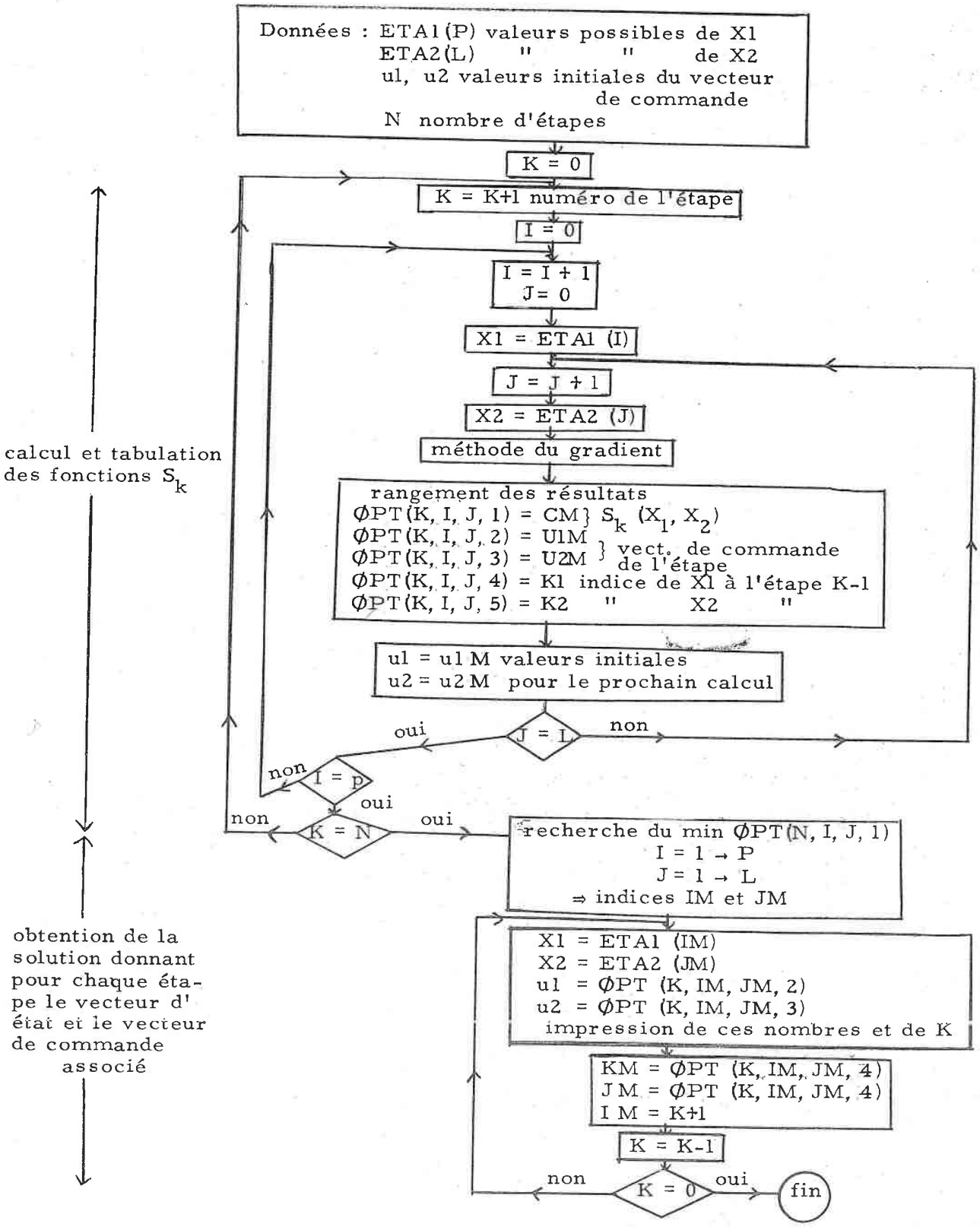
plusieurs variables. Le problème de stockage des fonctions de coût optimal est traité de la même façon que dans l'étude précédente de l'ajustement sur une équation, nous ne ferons donc qu'un rappel rapide des deux méthodes utilisées.

I - 4 Dénombrement sur le vecteur d'état

Comme dans l'étude de l'ajustement sur une équation, nous avons tout d'abord tabulé les fonctions S_k en ne les calculant qu'en un certain nombre de points et en remplaçant, dans le calcul de $S_k (X_1, X_2)$, $S_{k-1} (X_1 + \Delta X_1, X_2 + \Delta X_2)$ par une des valeurs mises en mémoire à l'étape précédente.

L'organigramme du programme principal nous montre que toutes les grandeurs à conserver à chaque étape sont des résultats de la méthode du gradient utilisée en sous-programme. En réalité, seules les valeurs des fonctions de commande sont des résultats de ce sous-programme qui utilise en paramètre la fonction à minimiser appelée "coût", nous donnons également l'organigramme du calcul d'une valeur de cette fonction.

Organigramme du Programme Principal



La fonction "coût" a la forme suivante

$$\int_{t_{N-k}}^{t_{N-k+1}} q\,(x_1, x_2, u_1, u_2, t)\,dt + S_{k-1}(x_1 + \Delta x_1,\ x_2 + \Delta x_2)$$

ses paramètres sont :

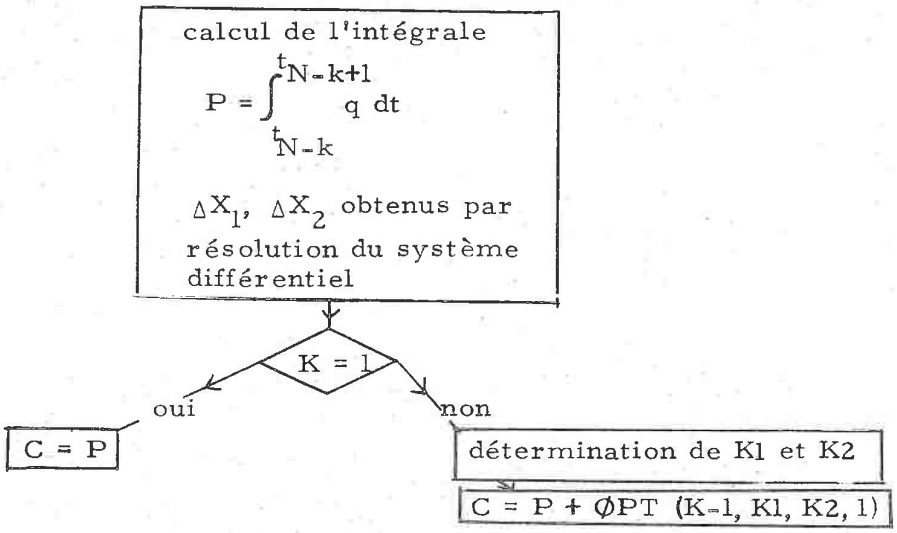
- K numéro de l'étape
- I indice donnant la valeur de X_1 dans le tableau ETA1
- J indice donnant la valeur de X_2 dans le tableau ETA2
- u_1, u_2 } valeurs des vecteurs de commande

ses résultats sont :

- C sa valeur
- K1, K2 } indices de $X_1+\Delta X_1$ dans ETA1 et de $X_2+\Delta X_2$ dans ETA2

Le calcul de l'intégrale sera fait dans le sous-programme lorsqu'on utilisera la méthode du trapèze, sinon on fera appel à un sous-programme d'intégration.

Organigramme :



I - 5 Utilisation de l'interpolation

L'utilisation d'une méthode d'interpolation n'apporte pas beaucoup de modifications à la méthode de résolution proposée ci-dessus.

En effet, dans le programme principal nous aurons, tout d'abord, à déterminer les points (X_1, X_2) d'interpolation. Pour chacun de ces points nous conserverons, le coût, les valeurs du vecteur de commande et du vecteur d'état à l'étape précédente. Dans la fonction coût, la détermination de K1 et K2 sera remplacée par l'appel du sous-programme d'interpolation qui donnera la valeur approchée de $S_{k-1}(X_1 + \Delta X_1, X_2 + \Delta X_2)$.
L'obtention de la solution se fera également en utilisant l'interpolation.

Pour des raisons matérielles nous n'avons pas pu utiliser une méthode d'interpolation sur plus de 9 points ; nous avons donc pris la méthode suivante :

- Soit une fonction de deux variables f (x, y) connue en neuf points d'un réseau rectangulaire.
- Soit [F] le vecteur colonne des valeurs prises par la fonction.
- Soit $[L_x]$ et $[L_y]$ les matrices de Lagrange correspondant respectivement aux points x_i et y_i choisis.
- Le polynôme d'interpolation est alors donné par la formule suivante :

$$P(x, y) = [1 \; x \; x^2] \otimes [1 \; y \; y^2] \cdot [L_x] \otimes [L_y] \; [F]$$

$\otimes$: produit de Kronecker

En résumé l'ajustement, tel que nous le présentons, dépend de deux facteurs :

- la fonction q

- l'utilisation du dénombrement ou de l'interpolation

Dans les applications, au paragraphe suivant, nous préciserons toujours dans quel cas nous nous trouvons.

III - APPLICATIONS NUMERIQUES DANS LE CAS DU PARTAGE IMPOSE

II - 1 Système linéaire à coefficients variables

Nous avons choisi le système suivant :

$$\begin{cases} \frac{dX_1}{dt} = \frac{t}{100} X_1 + \frac{(3t - 1)}{100} X_2 \\ \frac{dX_2}{dt} = \frac{4t}{100} X_1 + \frac{2}{100} X_2 \end{cases}$$

les fonctions de commande étant $u_1 = \frac{t}{100}$ et $u_2 = \frac{3\,t-1}{100}$ la fonction g_1 connue a été calculée en résolvant le système pour les conditions initiales

$$\begin{cases} X_1\ (0) = 1 \\ X_2\ (0) = 3 \end{cases}$$

Nous avons calculé l'intégrale par la méthode du trapèze et avons donc minimisé

$$\int_{T\emptyset}^{T} (X_1 - g_1)^2 + \left(\frac{dX_1}{dt} - g_1' \right)^2 \ dt.$$

II. 1. 1 Méthode avec dénombrement

Nous avons ajusté u_1 et u_2 sur l'intervalle $[1.1,\ 1.6]$ partagé en 5 intervalles partiels égaux.

Nous comparons la valeur u_1^i trouvée sur l'intervalle $[t_i,\ t_{i+1}]$ à la valeur exacte de la fonction de commande au milieu de l'intervalle c'est-à-di-

re pour $t_{mi} = \frac{t_i + t_{i+1}}{2}$

Résultats :

t	$u_1 \times 100$	$u_1(t_m) \times 100$	$u_2 \times 100$	$u_2(t_m) \times 100$
1.55	1.57	1.55	3.64	3.65
1.45	1.47	1.45	3.34	3.35
1.35	1.37	1.35	3.04	3.05
1.25	1.28	1.25	2.75	2.75
1.15	1.17	1.15	2.42	2.45

coût total : 10^{-4}

Dans ce tableau $u_1(t_m)$ et $u_2(t_m)$ sont les valeurs exactes et u_1 et u_2 les valeurs approchées.

II. 1. 2 Méthode avec interpolation

Dans les mêmes conditions, la méthode avec interpolation a donné des résultats identiques mais avec un coût total de 10^{-5}.

II - 2 Système non linéaire

Nous avons travaillé sur le système suivant où les fonctions de commande sont des exponentielles d'une des variables d'état. Nous avons choisi un tel système pour sa ressemblance avec un modèle mathématique de phénomène biologique.

$$\begin{cases} \frac{dX_1}{dt} = - u_1 X_1 + u_2 X_2 \\ \frac{dX_2}{dt} = - (u_1 - a) X_1 - u_2 X_2 \end{cases}$$

avec $a = 0.7 \exp(X_1)\, 10^{-2}$

$u_1 = 1.5 \exp(X_1)\, 10^{-2}$

$u_2 = 8 \exp(-X_1/10)\, 10^{-2}$

Nous avons supposé que nous connaissons une fonction expérimentale correspondant à une combinaison linéaire de X_1 et de X_2, soit $2X_1 + 3X_2$.

Sur l'intervalle $[1, 1.5]$ partagé en cinq intervalles partiels égaux, nous avons obtenu les résultats suivants en calculant l'intégrale par la méthode du trapèze.

u_1^D, u_2^D } résultats par la méthode avec dénombrement

u_1^I, u_2^I } résultats par la méthode avec interpolation

u_1^R, u_2^R } valeurs exactes pour $t = t_m$.

t_m	u_1^D	u_1^I	u_1^R	u_2^D	u_2^I	u_2^R
1.05	0.1657	0.1652	0.1685	0.0641	0.0609	0.0628
1.15	0.1595	0.1593	0.1622	0.0640	0.0617	0.0630
1.25	0.1538	0.1539	0.1566	0.0627	0.0624	0.0632
1.35	0.1490	0.1490	0.1514	0.0635	0.0631	0.0634
1.45	0.1445	0.1445	0.1466	0.0644	0.0639	0.0636

Dans la méthode avec dénombrement le coût total est de 10^{-2}

Dans la méthode avec interpolation le coût total est de 10^{-4}.

Nous avons également résolu ce problème en calculant l'intégrale

$\int_{T\emptyset}^{T} (2X_1 + 3X_2 - g_1(t))^2 \, dt$ par une méthode d'intégration numérique.

Nous avons choisi la méthode d'intégration suivante :

L'intégrale $I = \int_a^b F(x)\,dx$ est approchée par la quantité :

$I^* = \int_a^b P_N(x)\,dx$ où $P_N(x)$ est le polynôme d'interpolation de $F(x)$ sur $[a, b]$ dans une base de Tchebycheff et sur un support de Tchebycheff.

Nous nous sommes contenté, dans notre application, d'un polynôme d'interpolation de degré cinq, car les intervalles sur lesquels nous travaillons à chaque étape sont de longueur réduite.

Les résultats obtenus alors par la méthode utilisant l'interpolation sont équivalents aux précédents mais nécessitent un temps de calcul plus long ; cette différence provient du calcul des fonctions X_1 et X_2 en des points supplémentaires, calcul qui demande la résolution du système différentiel.

II - 3 Difficultés rencontrées

II. 3. 1 Difficultés dues au matériel utilisé

Nous ne disposions pour cette étude que d'un calculateur de faible capacité (CAE 510, 8K de mémoire). Le programme de résolution utilisant l'interpolation sur 9 points tels que nous l'avons présenté atteignait la limite de capacité de l'ordinateur. Nous n'avons donc pas pu utiliser des méthodes d'approximation plus élaborées pour calculer les fonctions de coût optimal, ni des méthodes d'optimisation plus performantes que la méthode du gradient. D'autre part la lenteur d'exécution de la machine a limité les essais effectués, à titre indicatif nous signalerons que l'obtention du tableau de résultats du paragraphe II - 2 a nécessité trois

heures de calcul.

II.3.2 Autres difficultés

Nous nous sommes tout d'abord heurté à des difficultés propres à tous les problèmes d'ajustement, c'est-à-dire :

- le choix des valeurs initiales à donner aux fonctions de commande
- le choix des points (X_1, X_2) en lesquels on calculera chaque fonction de coût optimal.

Nous avons vu, lors de l'étude de la méthode du gradient, que les valeurs initiales arbitraires des fonctions de commande n'interviennent que dans le calcul de la première valeur de S_1 ; nous pourrons donc déterminer ces valeurs expérimentalement en répétant ce calcul.

Dans l'étude de l'ajustement sur une équation différentielle, le problème du choix du domaine de définition de chaque fonction S_k est résolu par la connaissance de la fonction expérimentale. Mais, lorsque nous travaillons sur un système, nous connaissons moins de fonctions expérimentales qu'il n'existe de fonctions d'état ; nous ne possédons donc aucune indication sur les ordres de grandeur de certaines variables d'état. Afin d'éviter de calculer, pour cette raison, chaque fonction S_k en un très grand nombre de points, il sera nécessaire de faire, avant chaque résolution de problème d'ajustement, une simulation : c'est-à-dire une étude du modèle mathématique permettant de déterminer des ordres de grandeur des variables d'état et de commande. Cette étude présentera également l'intérêt de mettre en évidence l'influence des différentes fonctions de commande sur chacune des solutions du système et de déterminer ainsi quelle précision il est permis d'espèrer obtenir sur chaque variable de commande.

D'autres difficultés, d'ordre plus technique, sont apparues lors des applications numériques ; elles concernent le stockage des fonctions S_k (y) et u_k (y) et la précision de l'interpolation ; ces problèmes n'ont été résolus qu'au niveau de chaque cas particulier car en raison de l'insuffisance du matériel utilisé nous ne pouvions pas employer des méthodes d'approximation plus élaborées.

Cependant une étude récente (9) a montré que l'utilisation des méthodes d'approximation, en particulier des approximations dans une base de Tchebycheff avec support de Tchebycheff, en programmation dynamique à plusieurs variables, permettait de résoudre ces problèmes. Nous pouvons donc espérer, qu'avec un matériel plus puissant, ces difficultés techniques disparaîtront.

III - METHODE UTILISANT L'OPTIMISATION DU PARTAGE

III - 1 Equation fonctionnelle

D'après les remarques faites au paragraphe II. 3. 1. 3 du chapitre I, nous envisageons l'application de cette méthode uniquement dans les problèmes où nous connaissons le vecteur d'état à l'instant initial T_0.

Comme nous l'avons vu au paragraphe II. 3. 1. 2 du chapitre I, l'équation fonctionnelle du problème est alors la suivante :

$$S_k(t) = \min_{T\emptyset \le t_{k-1} \le t} \left(S_{k-1}(t_{k-1}) + \min_{u_k} \int_{t_{k-1}}^{t} q(x, u_k, \tau)\, d\tau \right)$$

avec $$S_1(t) = \min_{u_1} \left(\int_{T\emptyset}^{t} q(x, u_1, \tau)\, d\tau \right)$$

Rappelons la signification de $S_k(t)$:

$S_k(t)$ est le coût minimum obtenu en approchant les fonctions de commande sur $[T_0, t]$, par des fonctions en escaliers s'appuyant sur k intervalles partiels de $[T_0, t]$.

Voyons maintenant en quoi différent les méthodes de résolution de cette équation par rapport à celles employées précédemment.

III - 2 Méthodes de résolution

III. 2. 1 Méthode avec dénombrement

Chaque fonction S_k est calculée en un certain nombre de points t_i et la recherche du minimum sur la variable t_{k-1} est faite également par dénombrement ; ainsi $S_{k-1}(t_{k-1})$ est toujours une valeur obtenue

à l'étape précédente.

Le minimum sur u_k est obtenu comme précédemment par la méthode du gradient, la fonction à minimiser étant plus simple puisqu'elle ne comporte plus que l'intégrale.

Pour chaque valeur $S_k\ (t_i)$ nous mettrons en mémoire :

- la valeur t_{k-1}
- le vecteur u_k
- le vecteur d'état $x\ (t_i)$

les autres phases de la résolution, calcul de l'intégrale et résolution du système différentiel, seront les mêmes que précédemment.

III. 2. 2 Méthode avec interpolation

La recherche du minimum sur t_{k-1} se fait, cette fois par une méthode d'optimisation à une variable. Nous remarquerons, que sur cette équation fonctionnelle, l'emploi d'une méthode "minimax" est facilité par la connaissance de l'intervalle $[T_0, t]$ sur lequel doit se trouver t_{k-1}.

Chaque fonction S_k est approchée par une fonction $S_k^*\ (t)$. Pour le calcul de l'intégrale $\int_{t_{k-1}}^{t} q\ (x, u_k, \tau)\ d\tau$ nous devons connaître le vecteur d'état à l'instant t_{k-1}, ce vecteur sera obtenu également par l'interpolation sur les valeurs trouvées à l'étape précédente.

Cette détermination du vecteur d'état par interpolation constitue une différence par rapport à la méthode précédente où nous le choisissions.

III - 3 Application numérique

Nous avons appliqué la méthode avec dénombrement au système non linéaire étudié au paragraphe II. 2.

Dans les mêmes conditions, c'est-à-dire sur l'intervalle $[1, 1.5]$ et en connaissant la même fonction expérimentale, nous avons du utiliser 10 étapes afin d'obtenir un résultat d'une précision comparable à celle obtenue au paragraphe II. 2. Ce résultat est le suivant :

u_1 A et u_2 A sont les valeurs des fonctions de commande sur l'intervalle $[t_{i-1}, t_i]$ obtenues par la résolution de l'équation fonctionnelle ; u_1 R et u_2 R sont les valeurs exactes pour $t = t_m$.

t_{i-1}	t_i	u_1 A	u_2 A
1	1.03	0.168	0.0659
1.03	1.09	0.166	0.0652
1.09	1.15	0.162	0.0646
1.15	1.19	0.159	0.0662
1.19	1.24	0.156	0.0653
1.24	1.28	0.154	0.0651
1.28	1.31	0.153	0.0661
1.31	1.36	0.150	0.0654
1.36	1.41	0.148	0.0648
1.41	1.50	0.145	0.0634

t_m	u_1 R	u_2 R
1.05	0.1685	0.0628
1.15	0.1622	0.0630
1.25	0.1566	0.0632
1.35	0.1514	0.0634
1.45	0.1466	0.0636

le coût total est alors de 10^{-5}

III - 4 Comparaison avec la première méthode

Avantages

Comme nous l'avons signalé au chapitre I, les fonctions de coût optimal sont des fonctions d'une variable quel que soit l'ordre du système, ceci constitue le principal avantage de cette méthode par rapport à la précédente.

D'autre part nous n'avons plus ici le problème du choix des valeurs à donner aux variables d'état puisqu'elles apparaissent comme un résultat de chaque étape.

Inconvénient

Cette méthode semble donner de moins bons résultats que la méthode précédente pour un même nombre d'étapes, ceci peut s'expliquer par le fait que les recherches successives des différents vecteurs u_k sont indépendantes les unes des autres alors que, lorsque le partage est imposé, ces recherches sont liées.

En conclusion cette méthode avec optimisation du partage de l'intervalle d'étude apparaît préférable à la première lorsque l'ordre du système différentiel est élevé et lorsqu'on est limité au point de vue capacité du matériel utilisé ; en particulier sur CAE 510, elle seule permet l'ajustement sur un système d'ordre trois.

Cependant cette méthode présentera vraiment un intérêt par rapport à la précédente lorsqu'au lieu d'approcher les fonctions de commande par des fonctions en escalier, nous les approcherons par une suite de fonctions d'un type donné : linéaires ou combinaisons linéaires de polynômes. C'est dans cette direction qu'il faudra développer cette technique lorsque nous aurons à notre disposition un matériel plus puissant.

CONCLUSION

L'ajustement de paramètres variables ou de fonctions sur des systèmes différentiels non linéaires est un problème complexe pour lequel nous ne possédions, jusqu'à présent, aucune méthode de résolution numérique générale.

Nous avons établi tout d'abord un parallèle entre l'ajustement et la commande optimale et nous avons cherché une méthode de résolution fondée sur la programmation dynamique. L'étude théorique nous a montré que cette technique conduit effectivement à la solution du problème posé. Au niveau de l'application pratique, nous avons été obligé d'utiliser uniquement la forme discrète de la programmation dynamique et nous avons distingué deux méthodes selon que la discrétisation était imposée ou optimisée.

Ces méthodes sont apparues assez lourdes et pratiquement inutilisables sur un petit ordinateur. Les résultats présentés sont donc très partiels mais assez encourageants cependant pour permettre d'espérer que ces techniques seront efficaces dans l'avenir.

Ce travail n'est donc qu'une première étape qui permettra de résoudre par la suite des problèmes concrets, dans le domaine de la biologie humaine en particulier.

BIBLIOGRAPHIE

[1] BELLMAN R. E. - DREYFUS S. E.

La programmation dynamique et ses applications - Dunod

[2] BERTRAN J. P.

Méthode d'ajustement des paramètres d'un système différentiel linéaire ou non linéaire - Thèse de 3ième Cycle - Institut universitaire de calcul automatique - Nancy 1969.

[3] BOUDAREL R. - DELMAS J. - GUICHET P.

Commande optimale des processus - Dunod.

[4] CHARLES F.

Optimisation d'une fonction d'une variable - Contribution à la méthode du minimax. Thèse de 3ième Cycle - Institut universitaire de calcul automatique - Nancy 1968.

[5] KURT J. L.

Ajustement des paramètres des équations aux dérivées partielles. Application à un problème de chaleur. Thèse de 3ième Cycle - Institut universitaire de calcul automatique - Nancy 1968.

[6] LEGRAS J.

Précis d'analyse numérique - Dunod (1963).

[7] MONOT C.

Ajustement numérique d'un système différentiel linéaire à l'aide de la transformation de Laplace. Thèse de 3ième Cycle Institut Universitaire de Calcul Automatique - Nancy (1967).

[8] NASLIN P.

Introduction à la commande optimale - Dunod.

[9] YU Z. K.

Usage des techniques d'approximation en programmation dynamique. Thèse de 3ième Cycle - Institut Universitaire de Calcul Automatique - Nancy (1969).

NOM DE L'ETUDIANT : Mme Odile FOUCAUT

Nature de la thèse : Spécialité

"Ajustement de paramètres variables sur des systèmes différentiels non linéaires par la programmation dynamique".

Vu, Approuvé

et Permis d'imprimer

NANCY, le 6 Novembre 1969

Le Doyen,

J. ROBERT

UNIVERSITÉ DE NANCY * FACULTÉ DES SCIENCES *