

14/91

وزارة الجامعات
Ministère aux Universités

المدرسة الوطنية المتعددة التقنيات
المكتبة
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT GENIE ELECTRIQUE

PROJET DE FIN D'ETUDES

SUJET

LOGICIEL DE SIMULATION ET
D'OPTIMISATION D'UNE BOUCLE
D'ASSERVISSEMENT DIGITALE

Proposé par : R. ILOUL

Etudié par : H. SAARI

Dirigé par : R. ILOUL

PROMOTION JUIN 1991

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEDICACES

A mes parents

A mes frères et soeurs

A toute ma famille

A tous mes amis

A mes enseignants

Je dédie ce modeste travail.

hammad

REMERCIEMENTS

J'adresse mes vifs remerciements à monsieur Rachid ILOUL pour son aide, ses suggestions et ses précieux conseils.

Je remercie vivement mon ami Y. FENNI pour l'aide qu'il m'a apporté.

J'ajoute ma gratitude à tous les enseignants qui ont contribué à ma formation.

Je remercie également K. RIDOUH, T. LAHMER, M. DJEMAI,

L. BENCHEIKH, K. HAMITI et N. KACHEBI pour les meilleurs années que nous avons passé ensemble.

hammoud.

SOMMAIRE

- INTRODUCTION	3
- 1 ^{ère} PARTIE : DESCRIPTION DU LOGICIEL	4
1 - Introduction	5
2 - Structure générale du logiciel	6
3 - Menu principal	8
4 - commande du menu principal	8
- 2 ^{ème} PARTIE : ASPECT AUTOMATIQUE DU LOGICIEL	12
- Chapitre I - SIMULATION	13
1-1 - Introduction	13
1-2 - Résolution par la méthode des équations d'état	14
1-3 - Intégration numérique	20
- Chapitre II - TRAITEMENT DES REGLAGES DIGITAUX COMME SYSTEMES ECHANTILLONNES	22
2-1 - Introduction	22
2-2 - Le réglage digital	22
2-2-1 -Introduction	22
2-2-2 -Principe de simulation de réglage digital ..	22
2-2-3 - Algorithme de réglage	25
2-2-4 - Régulateur PID avec limitation incorporée .	30
2-2-5 - Choix de la période d'échantillonnage	32
2-2-6 - Effet des non-linéarités	33
2-2-7 - Effet de la quantification	37
- Chapitre III - OPTIMISATION DES PARAMETRES DU REGULATEUR	44
3-1 - Introduction	44
3-2 - Discrétisation de l'équation différentielle d'un régulateur PID	44
3-3 - Critère d'optimisation	46
3-4 - Méthodes d'optimisation	48
3-5- Méthode de Hooke et Jeeves	48
3-6 - Calcul de la fonction objectif	49
3-7 - Organigramme de la méthode de Hooke-Jeeves	50

- 3 ^{ème}	PARTIE : EXEMPLES	51
1 -	Introduction	52
2 -	Quantification	52
3 -	Optimisation des paramètres du régulateur	52
-	CONCLUSION	55
-	ANNEXE	57
-	BIBLIOGRAPHIE	66

INTRODUCTION

Les systèmes d'asservissement digitaux prennent une place, de plus en plus, importante dans l'industrie.

Dans ce projet, nous nous proposons de réaliser un logiciel interactif pour permettre l'insertion d'un compensateur dans la boucle d'asservissement.

Ce logiciel permettra en plus de la simulation du comportement du système seul, la simulation du comportement de la boucle d'asservissement. On simulera notamment le comportement du compensateur digital (quantification des données et des coefficients).

La possibilité de reprendre la simulation est laissée à l'utilisateur s'il estime le comportement du système non satisfaisant.

De plus, à partir du logiciel de simulation, on insère un logiciel permettant l'optimisation des paramètres du compensateur pour certaines configurations.

Nous avons divisé le travail en trois parties, dont la première est consacrée à la description du logiciel, où nous donnons les fonctions des commandes qui constituent son menu.

La seconde partie présente l'aspect automatique du logiciel, où nous présenterons tous les algorithmes nécessaires à la simulation et à l'optimisation.

La dernière partie apportera quelques exemples d'application. Les figures sont présentées en annexe.

Premiere Partie

Description du Logiciel

1 - INTRODUCTION:

Dans ce chapitre, nous allons présenter l'architecture interne du logiciel. Nous rappelons que ce logiciel a été réalisé dans le but d'assister l'utilisateur automatique ou non à la simulation et l'optimisation d'une boucle d'asservissement digitale.

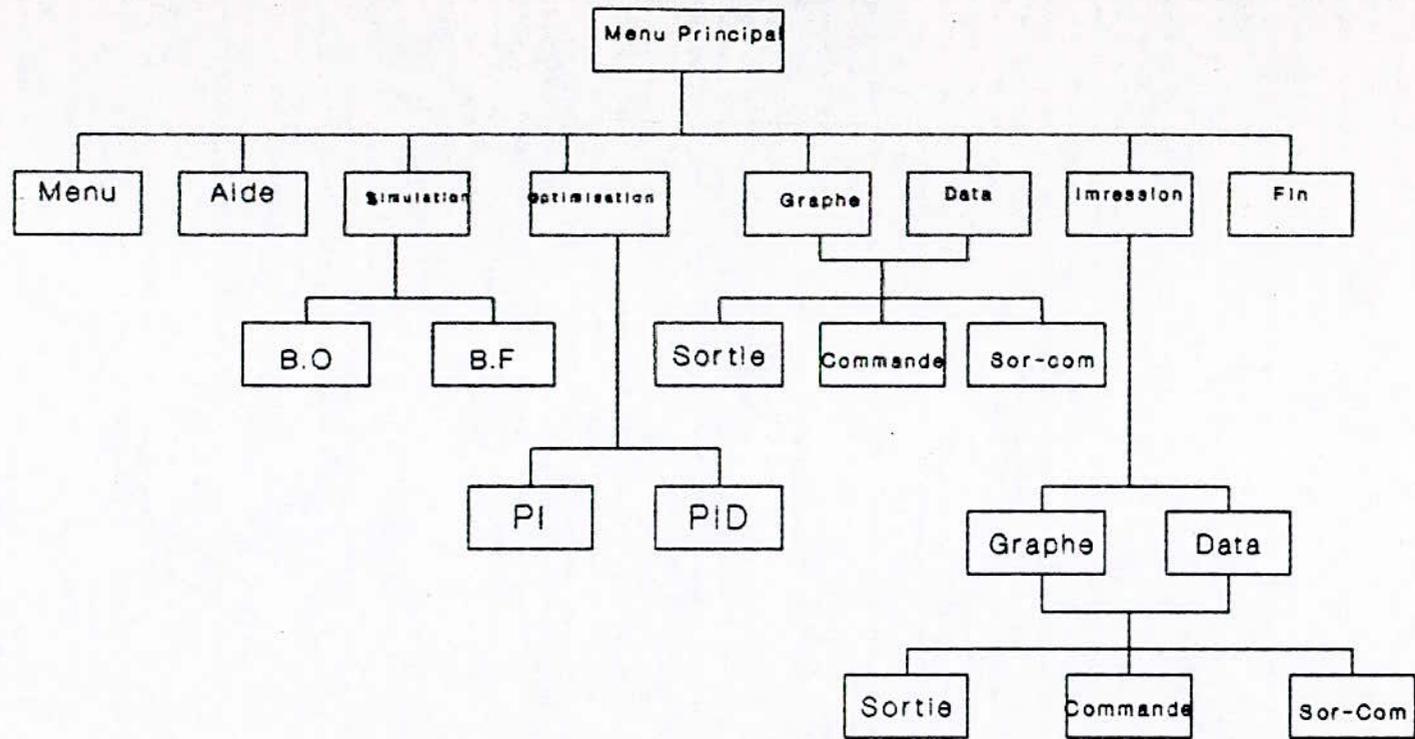
Le logiciel est développé sous l'environnement de programmation Turbo Pascal version 5.5 utilisant le système d'exploitation MsDos.

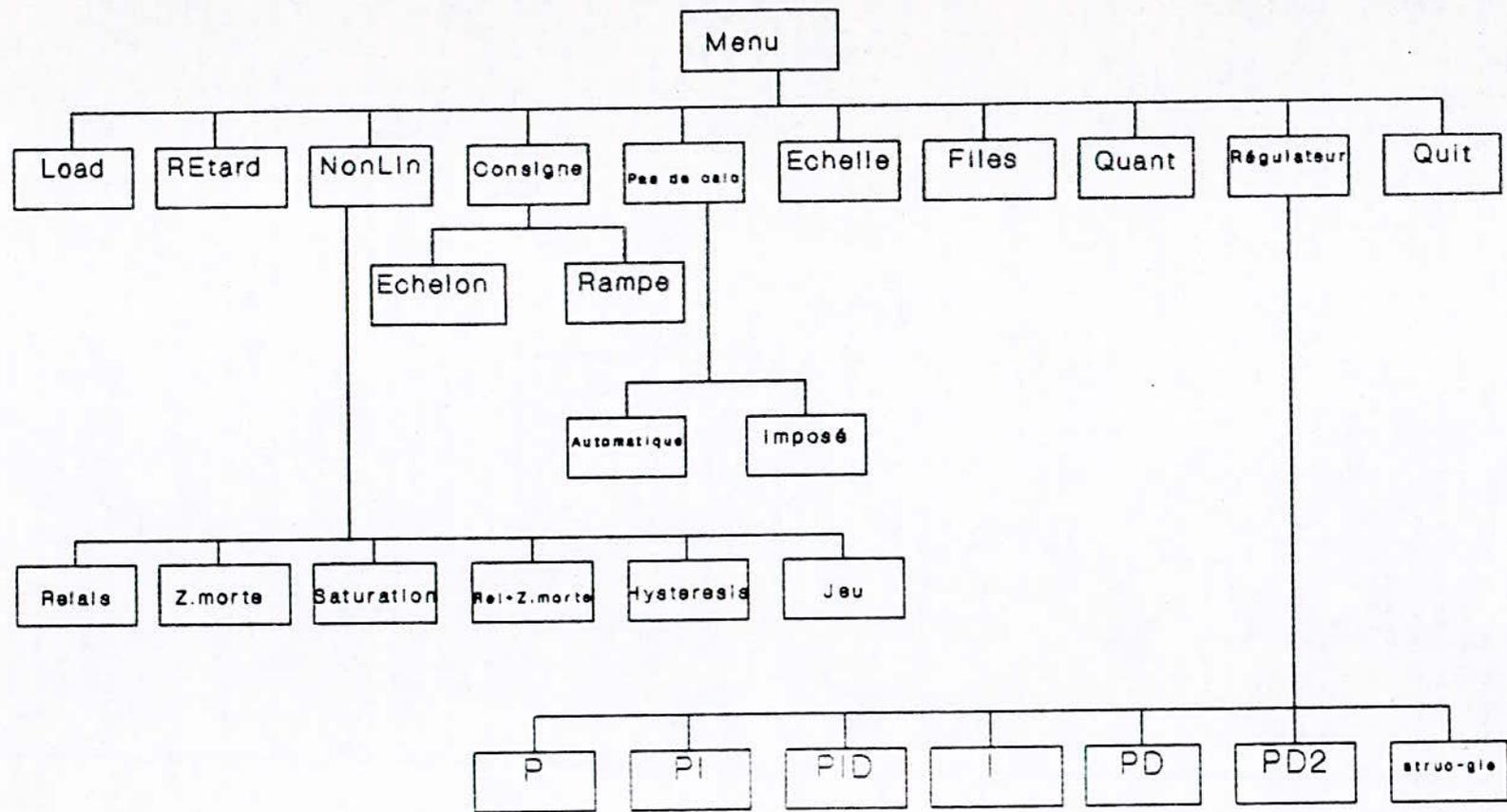
Le logiciel est standardisé de telle sorte qu'il puisse fonctionner sur tout micro-ordinateur répondant à la norme IBM PC et compatible. Le Logiciel présente une interactivité homme-machine; l'utilisateur est tout le temps guidé par un "Menu", ce qui rend son utilisation très souple.

Le logiciel est très simple à utiliser et ne nécessite pas de connaissance en Informatique.

Dans ce qui suit, nous mettons à l'évidence l'organigramme qui représente la structure globale du logiciel, en deuxième étape, nous décrirons les fonctions de chacune de ses options.

2 - Structure générale du logiciel:





3 - Menu Principal:

Le menu principal présente (8) options :

- Option Menu
- Option Aide
- Option Simulation
- Option Optimisation
- Option Graphe
- Option Data
- Option Impression
- Option Fin

Le choix de chaque option se fait en tapant sur la première lettre du nom de chaque option.

4 - Commande du Menu Principal:

4-1 - Choix de la commande "Menu" :

Cette Commande permet à l'utilisateur d'introduire toutes les données nécessaires pour faire de la simulation et l'optimisation.

En activant cette commande, il apparait un sous menu qui comporte 11 fonctions:

a - Fonction Load:

Permet d'introduire la fonction de transfert du système monovariante à étudier.

b - Fonction Retard:

Permet d'introduire la valeur du retard s'il a lieu.

c - Fonction NonLin:

Permet de tester si le système comporte une non-linéarité statique ou pas.

Si la nonlinéarité est validée, il apparait un autre sous menu où on a à choisir une des non-linéarités suivantes:

- Relais;
- Zone morte;
- Saturation;
- Relais avec Zone morte;
- Hystérésis;
- Jeu.

On introduit alors les paramètres correspondants.

d - Fonction Consigne:

Permet d'introduire la consigne. On a à choisir entre:

- un Echelon;
- une Rampe.

Pour l'échelon on introduit l'amplitude, et pour la rampe on introduit la pente.

e - Fonction Pas de Calcul:

Permet soit le calcul automatique du pas de calcul, soit d'introduire sa valeur.

f - Fonction Echelle:

Permet d'introduire :

- Le temps maximal de simulation;
- La valeur maximale de la sortie;
- la valeur maximale de la commande.

Ces deux dernières sont nécessaires pour la mise à l'échelle lors de la représentation graphique.

g - Fonction Files:

Permet de voir :

- La fonction de transfert;
- La consigne;
- La valeur du retard;
- La valeur du pas de calcul;
- Nonlinéarité (validée ou pas);
- Quantification (validée ou pas) (voir h).

h - Fonction Quantification:

Permet de tester si l'on tient compte de la quantification des paramètres, lors de la simulation ou non.

Si la quantification est validée, on introduit alors les paramètres nécessaires.

i - Fonction Période Echant:

Permet d'introduire la valeur de la période d'échantionnage.

j - Fonction Regulateur:

A l'aide de cette fonction, il apparait un sous menu où on a à choisir entre les régulateurs suivants:

- Rég P;
- Rég PI;
- Rég PID;
- Rég I;
- Rég PD;
- Rég PD2;
- Rég de structure générale.

k - Fonction Quit:

Permet de quitter le sous menu " Menu ".

4-2 - Choix de la commande " Aide " :

Cette commande permet à l'utilisateur d'avoir toutes les informations nécessaire lui permettant d'utiliser ce logiciel.

4-3 - Choix de la commande " Simulation " :

Permet de déclencher la simulation. Celle-ci se fait soit en boucle ouverte, soit en boucle fermée.

4-4 - Choix de la commande " Optimisation " :

Permet d'optimiser les paramètres des régulateurs:

- PI;
- PID.

Pour celà il faut introduire:

- Les valeurs initiales des paramètres ;
- Le pas de calcul de chaque paramètre;
- Le nombre maximum d'itération;
- L'erreur à admettre pour l'arrêt des calculs;
- Le coefficient de pondération du critère à minimiser.

4-5 - Choix de la commande " Graphe " :

L'édition graphique des résultats, permet de les apprécier sous forme de tracés.

On a à choisir parmi les trois types de graphes suivants:

- Sortie;
- Commande;
- Sortie - Commande.

4-6 - Choix de la commande "Data":

Les résultats sont mis dans un fichier et envoyés à l'écran. Comme pour la commande "Graphe", on a à choisir parmi les trois types de tableaux suivants:

- Sortie;
- Commande;
- Sortie - Commande.

4-7 - Choix de la commande "Impression":

A l'aide de cette commande, l'utilisateur peut faire l'impression de ses résultats numériques ou graphiques.

Il est noté que, si on veut faire l'impression des résultats graphiques, il faut introduire, avant d'appeler le logiciel, le fichier " Graphics.com ", en tapant " graphics ".

4-8 - Choix de la commande "Fin":

Cette commande permet de sortir du logiciel. Le même travail est effectué par la touche "Esc".

Deuxieme Partie

Aspect Automatique du Logiciel

CHAPITRE I

SIMULATION

1-1 - Introduction:

La simulation est une technique puissante pour vérifier la validité de la conception d'un système automatique en général, et de la configuration et du dimensionnement des systèmes de réglages en particulier.

Dans ce contexte, on peut faire la distinction entre fig 1.1:

- Simulation numérique;
- Simulation en temps réel.

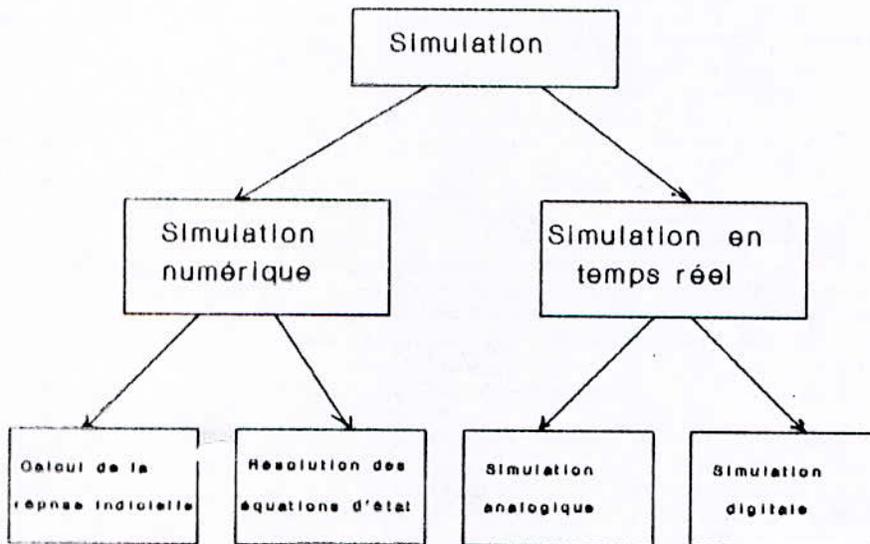


fig 1.1 classification de la simulation des systèmes automatiques

Avec la simulation numérique, on calcule l'allure temporelle de phénomènes transitoires sur ordinateur. On peut alors analyser d'une part, le comportement du système à régler, et d'autre part, on peut traiter le système complet avec les régulateurs. Les résultats dépendent de la rapidité de calcul de l'ordinateur.

On peut faire alors la subdivision en:

- Calcul de la réponse indicielle; [1]
- Résolution des équations d'état. [1]

Avec la simulation en temps réel, on représente en général uniquement le comportement du système à régler.

On distingue alors:

- La simulation analogique; [1]
- La simulation digitale. [1]

Dans ce chapitre, nous décrirons le procédé de résolution des équations d'état, et nous nous limitons au traitement de la simulation des systèmes analogiques, avec les régulateurs correspondant.

1-2 - Résolution par la méthode des équations d'état:

1-2-1 - Introduction:

Cette approche se prête particulièrement bien à la simulation numérique, vu qu'elle permet de traiter les non-linéarités, les limitations, et même les changements de structure intervenant lors de l'évolution du phénomène transitoire.

Cette méthode consiste en l'intégration des équations d'état du système par une des méthodes numériques.

1-2-2 - Fonction de transfert:

Dans tout ce qui suit, on suppose que la fonction de transfert est donnée sous la forme générale:

$$G(s) = \frac{C_{n-1}s^{n-1} + \dots + C_1s + C_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} \quad (\text{I.1})$$

où les coefficients c_1 et a_1 sont connus numériquement. On remarque que le degré du dénominateur est inférieur à celui du numérateur.

Cette fonction de transfert donne la relation entre la grandeur de sortie d'un système monovariante et sa grandeur d'entrée, qui peut être selon le cas; la grandeur de commande, la grandeur de consigne, la grandeur de perturbation, ou autre.

1-2-3 - Equation d'état:

En établissant les équations différentielles du système, il est aisé d'obtenir ses équations d'état.

Pour un système non-linéaire, les équations d'états sont:

$$\begin{cases} \dot{x} = f(x, u) \\ y = g(x) \end{cases} \quad (I.2)$$

Si le système est linéaire, les équations d'état prennent la forme:

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \quad (I.3)$$

On peut passer de la fonction de transfert $G(s)$ (I.1) à une des formes canonique (forme contrôlable) de représentation d'état (I.4) [4], telle que:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & 1 \\ -a_0 & -a_1 & \cdot & \cdot & -a_{n-1} \end{bmatrix}; B = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}; C = [c_1 \dots c_{n-1}] \quad (I.4)$$

1-2-4 - Intégration numérique:

L'équation d'état doit être résolue matriciellement en fonction du temps.

Aux instants discrets, les valeurs du vecteur d'état x , se

calculent par:

$$x_{k+1} = x_k + \Delta x_k \quad (I.5)$$

Dans le plus simple des cas (méthode d'intégration d'Euler), on a :

$$\dot{x}_k = f(x_k, u_k) \cdot \Delta t \quad (I.6)$$

Le pas de calcul (ou d'intégration) doit être considéré suivant la fig 1.2 (paragraphe 1-2-5).

1-2-5 - Pas de calcul:

Généralement, on adopte un pas de calcul Δt constant, de sorte que les valeurs discrètes du temps sont données par:

$$t_{k+1} = t_k + \Delta t \quad (I.7)$$

Le pas de calcul Δt doit être choisi correctement. Si Δt est trop grand, on aura des incertitudes sur le calcul. Si par contre, il est très petit, alors le temps de calcul sera excessivement élevé.

Le choix du pas de calcul dépend des pôles de la fonction de transfert. On obtient de bons résultats [1], pour des pôles réels

($p_1 = -\alpha$), ou des pôles complexes conjugués ($p_1 = -\alpha \pm j\omega$) selon:

$$\begin{cases} \Delta t \approx 1/4 \alpha \\ \Delta t \approx \pi/12 \omega \end{cases} \quad (I.8)$$

il faut choisir la valeur la plus petite.

L'algorithme correspondant est représenté à la fig 1.2 sous forme d'organigramme.

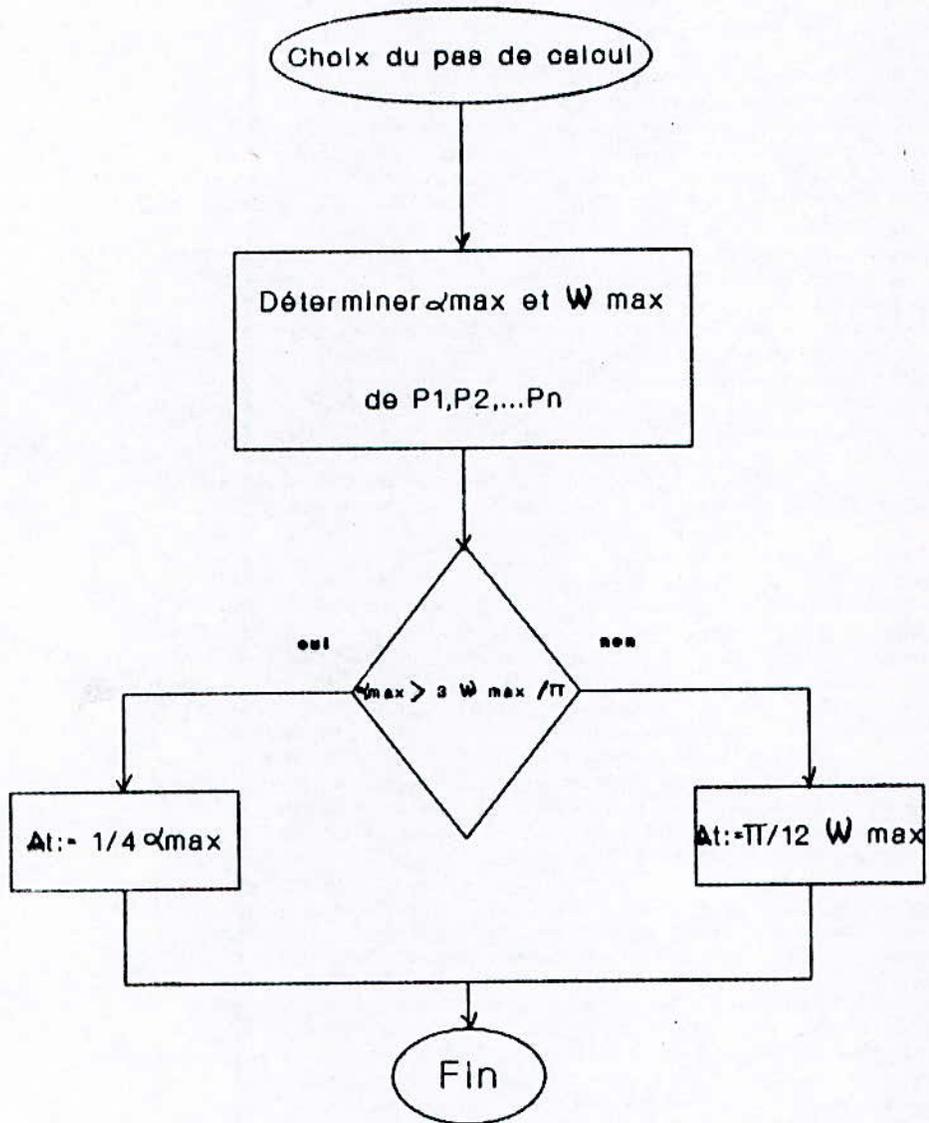
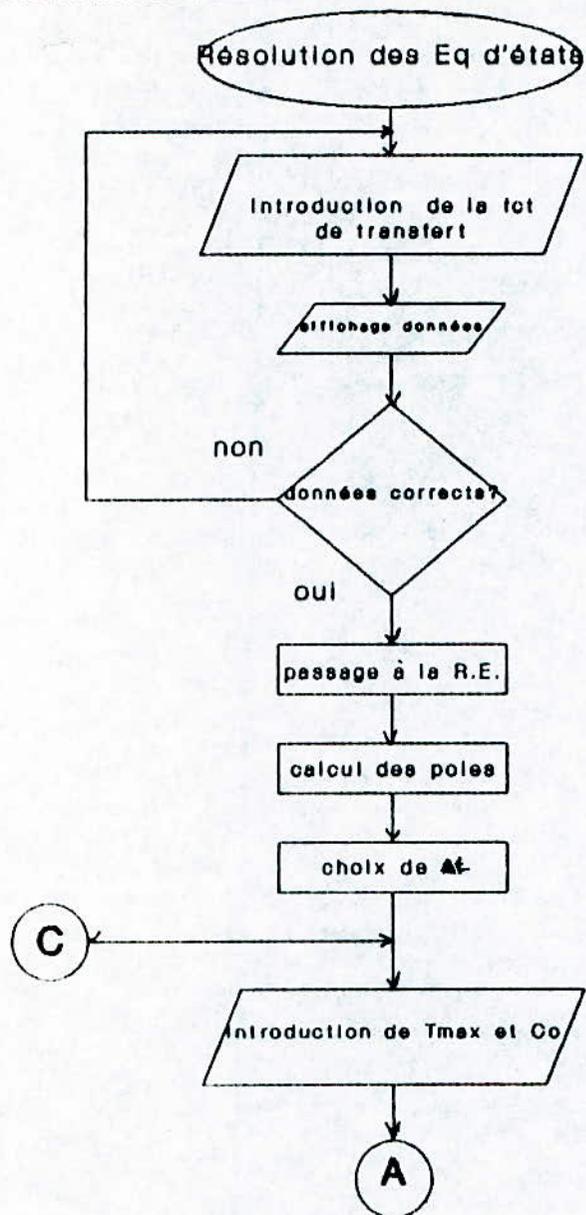


fig 1.2 Organigramme pour la détermination du pas de calcul.

1-2-6 - Organigramme:



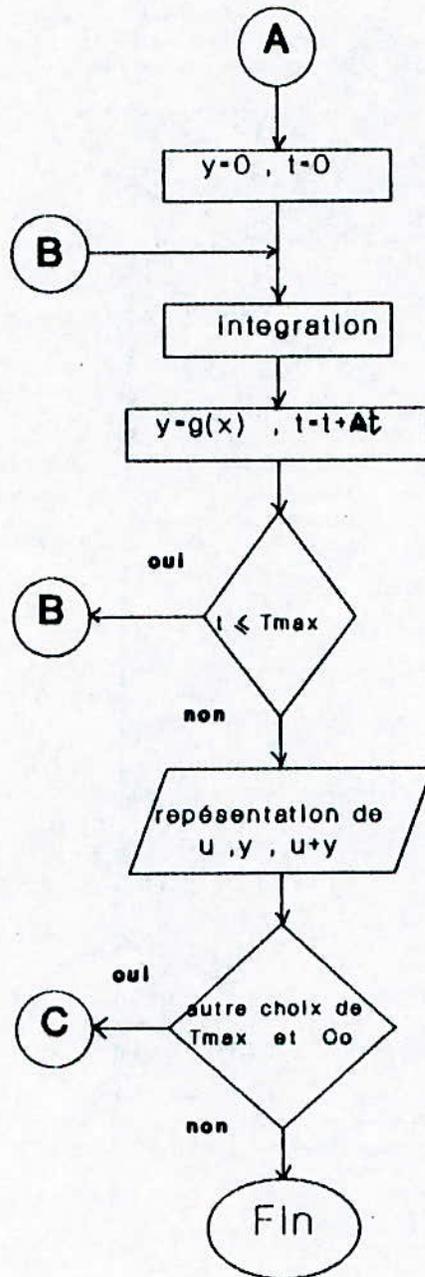


fig 1.3 Organigramme pour la simulation numérique par la résolution des équations d'état.

1-3 - Intégration Numérique:

1-3-1 - Introduction:

Il existe un nombre considérable de méthodes pour l'intégration numérique d'équations différentielles. Elles se différencient par leur précision, rapidité de calcul, et stabilité numérique, telle que;

Intégration rectangulaire, intégration polynomiale [5], méthode d'Euler, méthode de Runge-Kutta d'ordre quatre, méthode de la matrice de transition d'état [1], etc...

Nous allons étudier seulement les deux méthodes les plus utilisées; c'est les méthodes d'Euler et de Runge-Kutta d'ordre quatre.

1-3-2 - Méthode d'Euler:

C'est la méthode la plus simple. Le principe de base a été déjà donné au paragraphe 1-2-4.

Dans le cas général, on a la relation récursive:

$$x_{k+1} = x_k + f(x_k, u_k) \cdot \Delta t \quad (I.9)$$

Dans le cas de l'équation d'état linéaire (I.3), on obtient:

$$x_{k+1} = \Phi_E x_k + \Psi_E (B \cdot U_k) \quad (I.10)$$

avec :

$$\begin{cases} \Phi_E = I + A \cdot \Delta t \\ \Psi_E = I \cdot \Delta t \end{cases} \quad (I.11)$$

où I est la matrice identité.

Le pas de calcul dans ce cas doit être :

$$\Delta t \leq 1/10 \alpha \quad (I.12)$$

pour des pôles réels ($p_1 = -\alpha_1$)

et

$$\Delta t \leq \pi/30 \omega \quad (I.13)$$

pour des pôles complexes conjugués ($p_{1,i+1} = -\alpha_1 \pm j\omega_1$) selon la précision de calcul désirée.

1-3-3 - Méthode de Runge-Kutta d'ordre quatre:

C'est la méthode la plus utilisée pour l'intégration numérique. Dans le cas général, on a l'algorithme :

$$\left[\begin{array}{l}
 k_1 - f(x_k, u_k) \\
 x_{k+1/2}^1 - x_k + k_1 \Delta t / 2 \\
 k_2 - f(x_{k+1/2}^1, u_{k+1/2}) \\
 x_{k+1/2}^2 - x_k + k_2 \Delta t / 2 \\
 k_3 - f(x_{k+1/2}^2, u_{k+1/2}) \\
 x_{k+1}^1 - x_k + k_3 \Delta t \\
 k_4 - f(x_{k+1}^1, u_{k+1})
 \end{array} \right. \quad (I.14)$$

Avec ces valeurs particulières, on obtient finalement

$$x_{k+1} - x_k + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \cdot \Delta t \quad (I.15)$$

Dans le cas linéaire on aura :

$$x_{k+1} - \Phi_R x_k + \Psi_R (BU_k) \quad (I.16)$$

avec:

$$\left[\begin{array}{l}
 \Phi_R - I + A \Delta t + A^2 \frac{\Delta t^2}{2} + A^3 \frac{\Delta t^3}{6} + A^4 \frac{\Delta t^4}{24} \\
 \Psi_R - (I + A \frac{\Delta t}{2} + A^2 \frac{\Delta t^2}{6} + A^3 \frac{\Delta t^3}{24}) \cdot \Delta t
 \end{array} \right. \quad (I.17)$$

Le choix du pas d'intégration est moins critique.

CHAPITRE II

TRAITEMENT DES REGLAGES DIGITAUX COMME SYSTEME ECHANTILLONNE

2-1 - Introduction:

Jusqu'à présent, nous avons considéré la simulation numérique d'un système en boucle ouverte. Dans ce chapitre nous verrons le procédé à suivre pour simuler une boucle d'asservissement, et comment tenir compte des non-linéarités statiques lors de la simulation. Finalement, nous traiterons le problème de la quantification.

2-2 - Le réglage digital:

2-2-1 - Introduction:

Nous allons montrer d'abord, le principe de la simulation du réglage digital, toute les étapes pourront être alors résumées sur un organigramme. Nous décrirons ensuite les différents algorithmes de réglages dans le cas des régulateurs discrets, et nous traiterons le problème de limitation dans le cas d'un régulateur PID. On terminera par le choix de la période d'échantillonnage.

2-2-2 - Principe de simulation du réglage digital [3]:

Lorsqu'il n'y a pas de non-linéarité, il est possible de transformer les équations d'état différentielles du système à régler, en équations d'état aux différences. Ces dernières peuvent être alors réunies avec les équations d'état aux différences du régulateur. On obtient alors une solution numérique pour les équations aux différences du système global et ceci par récurrence.

Par contre, si on doit tenir compte des non-linéarités au niveau du système à régler, ou des limitations et de la quantification au niveau du régulateur, il serait alors

préférable de simuler séparément le système à régler et le régulateur.

Le schémas de principe d'un réglage digital (boucle d'asservissement digital) est donné par la fig 2.1.

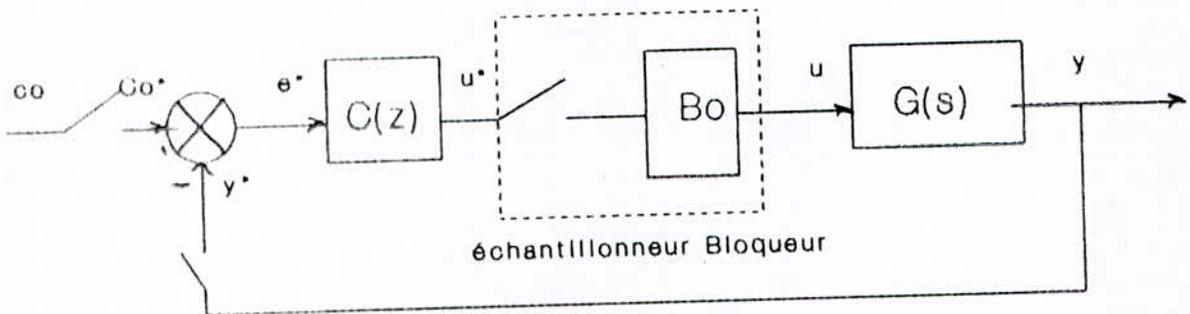


fig 2.1 schémas de principe d'un réglage digital

Dans ce schéma on a tenu compte de l'échantillonnage de la consigne " Co " et de la sortie " y " en " Co^* " et " y^* " d'une part. Ensuite le signal de commande discret u^* est transformé en un signal de commande continu, qui doit rester constant pendant une période d'échantillonnage. Ceci est obtenu en insérant un élément de maintien.

Pour la simulation, on élabore à chaque instant d'échantillonnage KTe l'algorithme de réglage (2-2-3). Avec cette valeur, on résoud, ensuite pendant une période d'échantillonnage Te , l'équation d'état du système à régler.

Le pas d'intégration Δt doit être une fraction entière de la période d'échantillonnage Te .

Ce procédé est résumé par l'organigramme suivant.

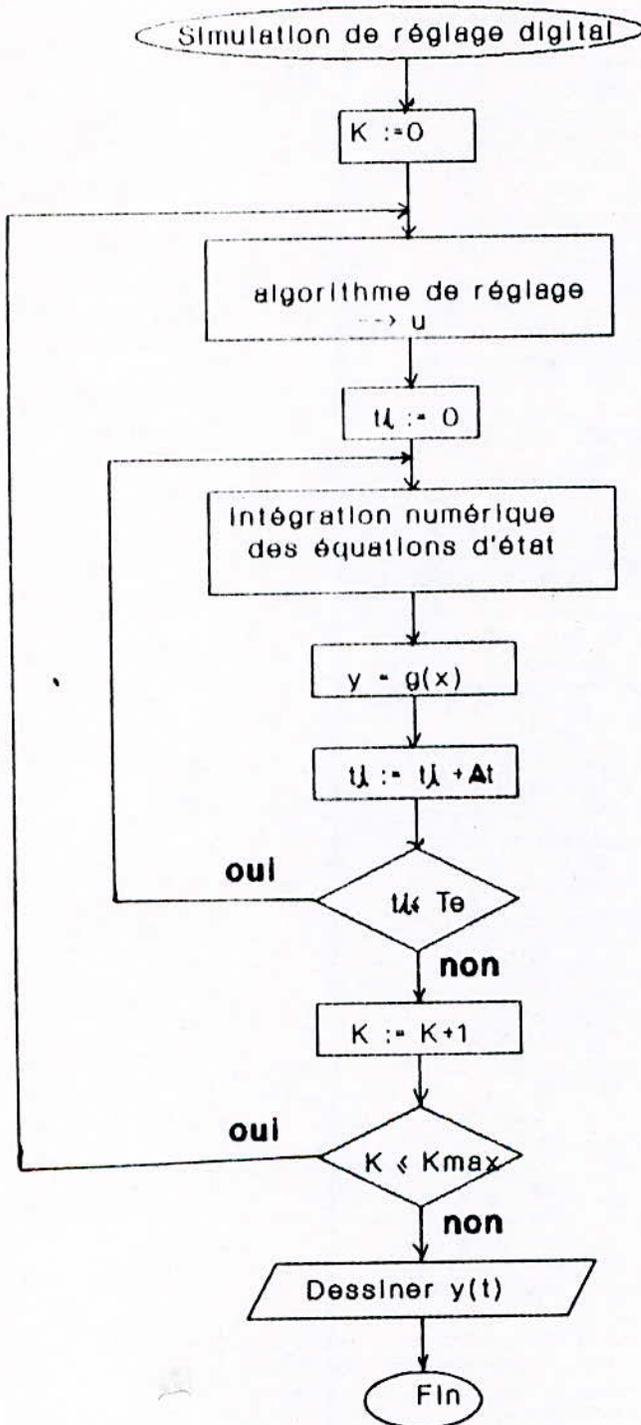


fig 2.2 Organigramme simplifié pour la simulation d'un réglage digital.

2-2-3 - Algorithme de réglage:

1 - Introduction:

Un régulateur discret élabore la grandeur de commande discrète $u^*[k]$ en fonction de l'écart de réglage discret $e^*[k]$.

Comme dans le cas des régulateurs continus, on distingue des régulateurs discrets standard, il s'agit des régulateurs P, I, PI, PID, PD, et PD2. Cependant dans le domaine des réglages digitaux, on fait souvent appel à des régulateurs de structure générale [2].

Nous allons étudier seulement les régulateurs P, PI, PID et le régulateur de structure générale.

2 - Régulateur Proportionnel P:

- Relation de base:

$$u^*[k] = K_p e^*[k] \quad (II.1)$$

- Fonction de transfert discrète:

$$C(z) = \frac{u^*(z)}{E^*(z)} = k_p \quad (II.2)$$

- Algorithme de réglage:

A partir de (II.1), on obtient

$$\left[\begin{array}{l} u := K_p e \\ \text{sortir } u \end{array} \right. \quad (II.3)$$

3 - Régulateur Proportionnel-Intégral PI:

- Relation de base:

$$u^*[k] = k_p e^*[k] + k_i \sum_{i=0}^{k-1} e^*[i] \quad (II.4)$$

Le 1^{er} terme correspond au comportement proportionnel du régulateur et le deuxième terme à son comportement intégrateur.

-Fonction de transfert discrète:

Lorsqu'on soumet l'équation (II.4) à la transformation en z , on obtient:

$$U^*(z) = (k_p + k_i \frac{z}{z-1}) E^*(z) \quad (II.5)$$

la fonction de transfert du régulateur PI devient alors

$$C(z) = k_p + k_i \frac{z}{z-1} \quad (II.6)$$

ou sous la forme suivante:

$$C(z) = \frac{b_1 z + b_0}{z-1} \quad (II.7)$$

avec:

$$b_1 = k_p + k_i \quad ; \quad b_0 = -k_p \quad (II.8)$$

- Transformation de la relation de base:

Pour la programmation de l'algorithme de réglage, il est judicieux d'introduire une grandeur auxiliaire définie par:

$$x^*[k-1] = k_i \sum_{i=0}^{k-1} e^*[i] \quad (II.9)$$

la relation (II.4) prend la forme

$$u^*[k] = x^*[k-1] + (k_p + k_i) e^*[k] \quad (II.10)$$

- Algorithme de réglage:

L'algorithme de réglage, devient avec $K_{pi} = k_p + k_i$

$$\left[\begin{array}{l} u := x + k_{pi} e \\ \text{sortir } u \\ x := x + k_i e \end{array} \right. \quad (II.11)$$

x doit être initialisé à 0.

4 - Régulateur Proportionnel- Intégral- Dérivée PID:

- Relation de base:

$$U^*[k] = k_p e^*[k] + k_i \sum_{i=0}^k e^*[i] + k_d (e^*[k] - e^*[k-1]) \quad (\text{II.12})$$

k_d est appelé coefficient de dérivée.

- Fonction de transfert discrète:

De la relation (II.12) et à l'aide de la transformation en z on obtient:

$$U^*(z) = \left(k_p + k_i \frac{z}{z-1} + k_d \frac{z-1}{z} \right) E^*(z) \quad (\text{II.13})$$

d'où:

$$C(z) = k_p + k_i \frac{z}{z-1} + k_d \frac{z-1}{z} \quad (\text{II.14})$$

Lorsqu'on met l'expression pour la fonction transfert discrète sous dénominateur commun, on obtient:

$$C(z) = \frac{b_2 z^2 + b_1 z + b_0}{z(z-1)} \quad (\text{II.15})$$

avec:

$$b_2 = k_i + k_p + k_d \quad ; \quad b_1 = -(k_p + 2k_d) \quad ; \quad b_0 = -k_d \quad (\text{II.16})$$

- Transformation de la relation de base:

Pour la programmation de l'algorithme de réglage d'un régulateur PID discret, on doit également introduire la grandeur auxiliaire définie par la relation (II.9), on obtient:

$$u^*[k] = x^*[k-1] + (k_p + k_i + k_d) e^*[k] - k_d e^*[k-1] \quad (\text{II.17})$$

- Algorithme de réglage:

On peut alors écrire l'algorithme de réglage, avec

$$K_{pid} = K_p + K_i + k_d$$

$$\left[\begin{array}{l} u := x + K_{pid} e - K_d e_{-1} \\ \text{sortir } u \\ x := x + K_1 e \\ e_{-1} := e \end{array} \right. \quad (II.18)$$

x, e_{-1} doivent être initialisés à 0.

5 - Régulateur de structure générale:

- Introduction:

Les régulateurs standards étudiés précédemment, possèdent une fonction de transfert du 1^{er} ou 2^{ème} ordre. En plus, les coefficients du dénominateur sont imposés à priori selon le type du régulateur.

Cependant, avec des régulateurs discrets d'ordre supérieur à deux, ou avec une structure permettant d'influencer les coefficients du dénominateur, on obtient souvent une qualité de réglage meilleure que celle réalisable avec un régulateur standard.

- Relation de base:

La relation de base d'un régulateur discret d'ordre n peut être décrite par l'équation aux différences suivante:

$$\begin{aligned} a_n U^*[k] + a_{n-1} U^*[k-1] + \dots + a_0 U^*[k-n] - \\ b_n e^*[k] + b_{n-1} e^*[k-1] + \dots + b_0 e^*[k-n] \end{aligned} \quad (II.19)$$

où a_n, a_{n-1}, \dots, a_0 et b_n, b_{n-1}, \dots, b_0 sont les coefficients qui déterminent le comportement du régulateur discret.

- Fonction de transfert discrète:

De la relation (II.19) et à l'aide de la transformation en z , on obtient :

$$C(z) = \frac{b_n z^n + b_{n-1} z^{n-1} + \dots + b_0}{a_n z^n + a_{n-1} z^{n-1} + \dots + a_0} = \frac{U^*(z)}{E^*(z)} \quad (II.20)$$

pour des raisons de réalisabilité, il faut que a_n soit différent de zéro.

- Transformation de la relation de base:

Pour trouver une forme adéquate pour l'algorithme de réglage, il est judicieux de transformer d'abord la relation de base. Dans ce but, on va regrouper les termes dans (II.20) de la manière suivante, avec $a_n = 1$:

$$U^*(z) - b_n E^*(z) + [b_{n-1} E^*(z) - a_{n-1} U^*(z)] z^{-1} + \dots \quad (\text{II.21})$$

$$\dots + [b_0 E^*(z) - a_0 U^*(z)] z^{-n}$$

On introduit maintenant des grandeurs auxiliaires qui permettront de faire la sommation de la partie droite d'une manière récursive. Ces grandeurs auxiliaires sont définies par

$$\left\{ \begin{array}{l} X_0^*(z) - b_0 E^*(z) - a_0 U^*(z) \\ X_1^*(z) - b_1 E^*(z) - a_1 U^*(z) + X_0^*(z) z^{-1} \\ \vdots \\ X_{n-1}^*(z) - b_{n-1} E^*(z) - a_{n-1} U^*(z) + X_{n-2}^*(z) z^{-1} \end{array} \right. \quad (\text{II.22})$$

La transformée en z de la grandeur de commande est alors donnée par :

$$U^*(z) - b_n E^*(z) + X_{n-1}^*(z) z^{-1} \quad (\text{II.23})$$

par la transformée en z inverse, on obtient :

$$u^*[k] - b_n e^*[k] + x_{n-1}^*[k-1] \quad (\text{II.24})$$

pour les grandeurs auxiliaires, et en inversant également la séquence des expressions, il existe les relations :

$$\left[\begin{array}{l} x_{n-1}^* [k] - b_{n-1} e^* [k] - a_{n-1} u^* [k] + x_{n-2}^* [k-1] \\ \vdots \\ x_1^* - b_1 e^* [k] - a_1 u^* [k] + x_0^* [k-1] \\ x_0^* [k] - b_0 e^* [k] - a_0 u^* [k] \end{array} \right. \quad (II.25)$$

- Algorithme de réglage:

A partir des relations (II.24) et (II.25) on tire l'algorithme de réglage suivant:

$$\left[\begin{array}{l} u := b_n e + x_{n-1} \\ \text{sortir } u \\ \text{pour } i := n-1 \dots 1 \\ \quad x_i := b_i e - a_i u + x_{i-1} \\ x_0 := b_0 e - a_0 u \end{array} \right. \quad (II.26)$$

on initialise les valeurs auxiliaires x_{n-1}, \dots, x_0 par 0.

2-2-4 - Régulateur PID avec limitation incorporée:

1 - Introduction:

Lorsque le signal de sortie u du régulateur doit être limité à u_{\max} ou u_{\min} , l'algorithme de réglage doit être alors modifié. En plus de la limitation de la sortie, pour éviter des phénomènes transitoires gênants lors de l'intervention des limitations, on doit également corriger la composante intégrale.

On montrera dans ce qui suit les mesures à prendre pour cette correction dans le cas du régulateur PID. Les mêmes mesures doivent aussi être appliquées aux autres régulateurs standards, en particulier aux régulateurs PI et I.

2 - Mesures à prendre:

Lorsque la grandeur de sortie $u[k]$ est en limitation, le signal de sortie est u_{\lim} , égale à u_{\max} ou u_{\min} . La composante intégrale x n'est plus

incrémentée selon (II.18) mais selon:

$$x[k+1] = x[k] + K_i e_{lim}[k] \quad (II.27)$$

où $e_{lim}[k]$ est un écart de réglage fictif

$$e_{lim}[k] = (u_{lim}[k] - x[k] + K_d e[k-1]) / K_{pid} \quad (II.28)$$

Pour établir l'algorithme de réglage, il est judicieux de déterminer l'écart de réglage fictif à partir de la différence entre la grandeur de sortie non limitée et limitée.

En désignant par $u'[k]$ la grandeur de sortie non limitée, on obtient de (II.18)

$$u'[k] - u_{lim}[k] = K_{pid}(e[k] - e_{lim}[k]) \quad (II.29)$$

d'où l'on tire

$$e_{lim}[k] = e[k] - (u'[k] - u_{lim}[k]) / K_{pid} \quad (II.30)$$

3 - Algorithme de réglage:

L'algorithme de réglage (II.18) doit être modifié lorsqu'il est nécessaire d'incorporer une limitation de la grandeur de sortie du régulateur.

Comme on l'a montré au paragraphe précédent, il est alors indispensable de corriger aussi la composante intégrale.

$$\begin{array}{l}
 u' := x + K_{pid} e - K_d e_{-1} \\
 \text{si } u' > u_{max} \text{ alors } u := u_{max} \\
 \quad \text{sinon si } u' < u_{min} \text{ alors } u := u_{min} \\
 \quad \quad \text{sinon } u := u' \\
 \text{sortir } u \\
 e_{lim} := e - (u' - u) / K_{pid} \\
 x := x + K_i e_{lim} \\
 e_{-1} := e
 \end{array} \quad (II.31)$$

2-2-5 - Choix de la période d'échantillonnage:

1 - Introduction:

Lors de la conception d'un système de réglage échantillonné, la période d'échantillonnage n'est pas connue à priori et constitue un des paramètres à déterminer.

Nous mentionnerons dans la suite quelques remarques qui nous aideront à trouver une valeur adéquate pour cette période d'échantillonnage.

2 - comportement dynamique du système à régler:

Le comportement dynamique du système continu à régler peut être caractérisé par les pôles p_1 de la fonction de transfert $G(s)$.

Une étude approfondie a été faite dans [2]. Pour des pôles complexes conjugués la partie imaginaire doit satisfaire

$$T_e \leq \pi / (4 \omega_1) \quad (II.32)$$

Pour des pôles réels $-\alpha_1$, on doit satisfaire

$$T_e \leq 1 / (2 \alpha_1) = T_1 / 2 \quad (II.33)$$

3 - Comportement dynamique du circuit de réglage échantillonné:

Lors de la conception d'un système de réglage échantillonné, on doit choisir la période d'échantillonnage selon les règles mentionnées au paragraphe précédent. Pour cette période d'échantillonnage, on évalue la qualité de réglage, on répète alors le même calcul pour des valeurs de T_e plus petite (par exemple la moitié de la valeur précédente). On constate en général que, à partir d'une certaine valeur de la période d'échantillonnage, la qualité de réglage ne varie plus de manière importante. Ainsi on obtient une valeur "optimale" pour la période d'échantillonnage.

2-2-6 - Effet des non-linéarités:

1 - introduction:

Dans cette section, nous prendrons en considération quelques non-linéarités statiques. Pour chacune d'entre elles, nous présenterons l'algorithme à suivre pour réaliser sa simulation.

Dans cette étude nous avons considéré le cas particulier où la non-linéarité est placée à la sortie du système comme le montre la fig 2.3.

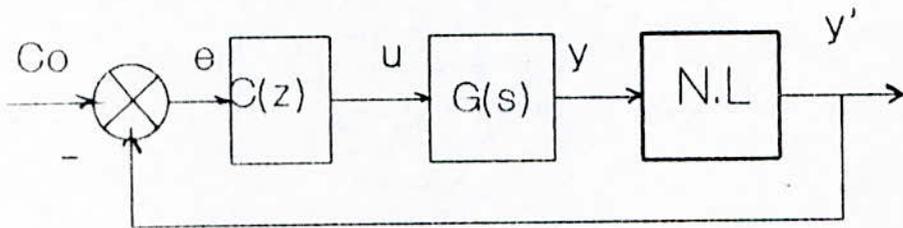


fig 2.3 Schémas de réglage avec non-linéarité

Notons qu'il existe une autre possibilité de placement de la N.L., i.e à la sortie du régulateur $C(z)$, on a vu ce cas pour la limitation de l'action de la commande(2-2-4), ainsi que pour la quantification(2-2-6).

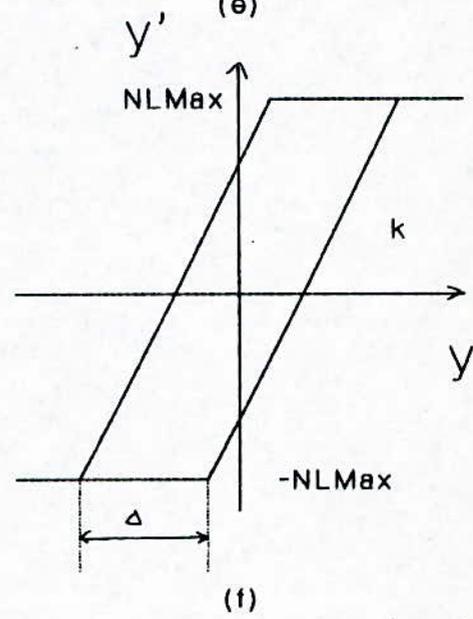
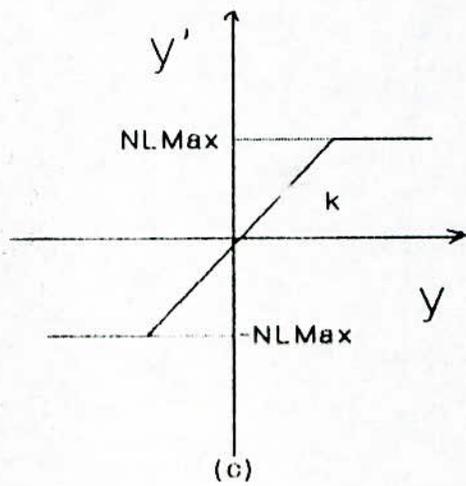
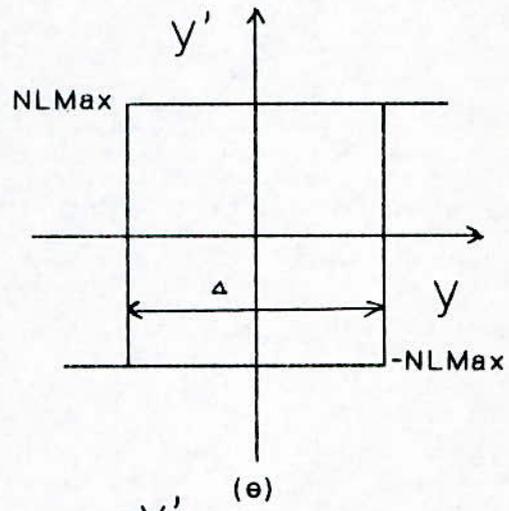
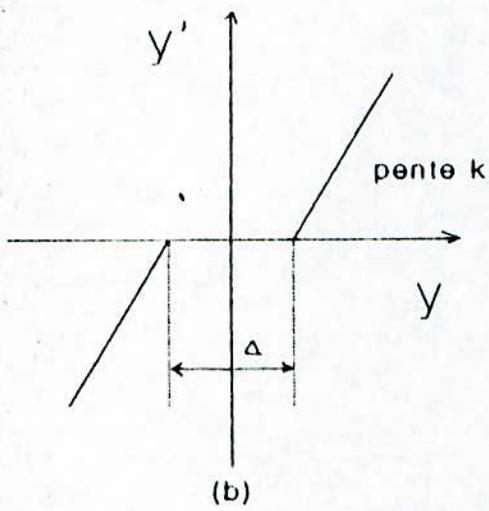
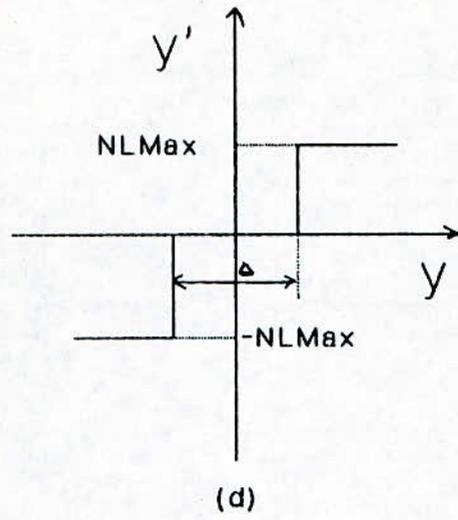
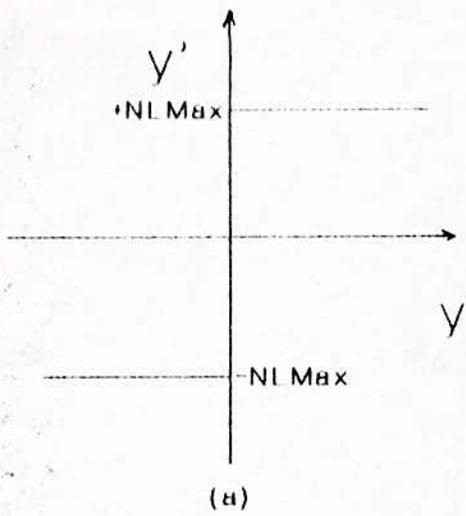


fig 2.4 Non-linéarités usuelles.

2 - Non-linéarités usuelles [6]:

Nous étudions ici l'influence de quelques non-linéarités typiques, que l'on rencontre souvent dans de nombreux systèmes.

Chacune d'elle est représentée par une courbe caractéristique, et à partir de celle-ci, on tire les équations à prendre en considération lors de la simulation.

a) - Élément non-linéaire à relais:

On rencontre cette caractéristique dans la plupart des entraînements réglés, comme machines, robots, laminoirs, ...etc.

Cette caractéristique peut être exprimée analytiquement par:

$$\left[\begin{array}{l} \text{si } y < 0 \text{ alors } y' := -NLMax \\ \text{si } y \geq 0 \text{ alors } y' := NLMax \end{array} \right. \quad (II.34)$$

b) - Élément non-linéaire à zone morte:

On le rencontre par exemple lors de la présence d'un jeu de transmission dans un entraînement.

Cette caractéristique peut être exprimée analytiquement par:

$$\left[\begin{array}{l} \text{si } |y| \leq \Delta/2 \text{ alors } y' := 0 \\ \text{sinon} \\ \quad \text{si } y > 0 \text{ alors } y' := k(y - \Delta/2) \\ \quad \text{sinon } y' := k(y + \Delta/2) \end{array} \right. \quad (II.34)$$

c) - Élément non-linéaire à saturation:

Ce phénomène existe dans tous les systèmes réels. Cette caractéristique peut être exprimée analytiquement par:

$$\left[\begin{array}{l} \text{si } |y| \leq NLMax/k \text{ alors } y' := k.y \\ \text{sinon} \\ \quad \text{si } y \geq NLMax \text{ alors } y' := NLMax \\ \quad \text{sinon} \\ \quad \quad \text{si } y \leq -NLMax \text{ alors } y' := -NLMax \end{array} \right. \quad (II.35)$$

d) - Élément non-linéaire à relais avec zone morte:

$$\left[\begin{array}{l} \text{si } |y| < \Delta/2 \text{ alors } y' := 0 \\ \text{sinon} \\ \quad \text{si } y \geq \Delta/2 \text{ alors } y' := \text{NLMax} \\ \quad \text{sinon} \\ \quad \quad \text{si } y \leq -\Delta/2 \text{ alors } y' := -\text{NLMax} \end{array} \right. \quad (\text{II.36})$$

e) - Hystérésis:

$$\left[\begin{array}{l} 1 - \text{si } \dot{y} \geq 0 \text{ alors} \\ \quad \text{si } y \geq \Delta/2 \text{ alors } y' := \text{NLMax} \\ \quad \quad \text{sinon } y' := -\text{NLMax} \\ 2 - \text{si } \dot{y} < 0 \text{ alors} \\ \quad \text{si } y > -\Delta/2 \text{ alors } y' := \text{NLMax} \\ \quad \quad \text{sinon } y' := -\text{NLMax} \end{array} \right. \quad (\text{II.37})$$

Remarque : la dérivée y est approximé par $\Delta y/\Delta t$.

f) - Jeu:

$$\left[\begin{array}{l} 1 - \text{si } \dot{y} \geq 0 \text{ alors} \\ \quad \text{si } y < -\text{NLMax}/k + \Delta/2 \text{ alors } y' := -\text{NLMax} \\ \quad \text{si } -\text{NLMax}/k + \Delta/2 < y < \text{NLMax}/k + \Delta/2 \text{ alors} \\ \quad \quad y' := k(y - \Delta/2) \\ \quad \text{sinon} \\ \quad \quad y' := \text{NLMax} \\ 2 - \text{si } \dot{y} < 0 \text{ alors} \\ \quad \text{si } y > \text{NLMax}/k - \Delta/2 \text{ alors } y' := \text{NLMax} \\ \quad \text{si } -\text{NLMax}/k - \Delta/2 \leq y \leq \text{NLMax}/k - \Delta/2 \text{ alors} \\ \quad \quad y' := k(y + \Delta/2) \\ \quad \text{sinon} \\ \quad \quad y' := -\text{NLMax} \end{array} \right. \quad (\text{II.38})$$

2-2-7 - Effet de la Quantification:

1 - Introduction:

Dans cette section, nous introduisons la notion de quantification et son effet dans la boucle d'asservissement digitale.

Nous commencerons tout d'abord par définir une grandeur digitale. Nous parlerons ensuite de la conversion d'une grandeur analogique en une grandeur digitale, qui sera suivie de la définition de la quantification. Après la discussion du codage et du format attribué à une grandeur digitale, on montrera les principes à considérer lors de la simulation.

On terminera par un schéma bloc qui résumera les points où l'on doit tenir compte de la quantification dans une boucle d'asservissement.

2 - Définition d'une grandeur digitale:

Une grandeur digitale est une grandeur ne pouvant prendre qu'un nombre limité d'états discrets. Elle représente normalement une grandeur analytique, avec une précision donnée par le nombre des chiffres, codé par une série de grandeurs binaires.

3 - Données de processus analogique et leur conversion:

Un grand nombre de données d'un processus sont des grandeurs physiques (grandeurs analogiques) représentées, le plus souvent par une tension continue.

Ces grandeurs analogiques peuvent varier continuellement entre les limites maximales et minimales. Pour le traitement dans le calculateur de processus, ces grandeurs doivent être converties et représentées sous forme digitale (on parle souvent aussi de forme numérique).

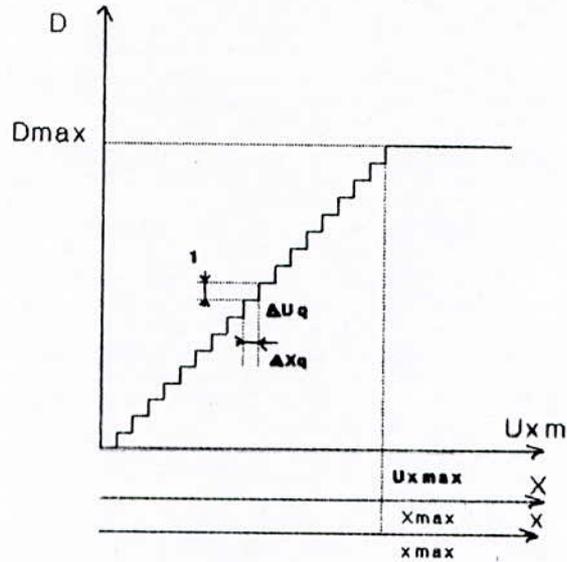


fig 2.5 Relation: grandeur digitale- grandeur analogique

La conversion d'une grandeur physique en grandeur digital se fait par:

- Convertisseurs A/D;
- Organes de mesures digitaux.

La fig 2.5 montre la relation entre une grandeur digitale D et une grandeur analogique. Cette dernière peut être représentée par:

- Tension de mesure U_{xm}
- Grandeur physique X
- Grandeur relative x

La grandeur digitale D est supposée dans ce contexte, comme un nombre entier.

La valeur D_{Max} doit correspondre à la valeur maximale de U_{xMax} (X_{Max} , x_{Max}), que peut prendre la grandeur physique dans le cas extrême du fonctionnement normal, en tenant compte du dépassement lors de phénomènes transitoires.

4 - quantification:

Une grandeur physique ne peut être représenté par une grandeur digitale qu'avec une précision limitée car D (comme grandeur entière), ne peut varier que par pas de 1 (fig 2.5), la variation par un pas est appelée 'quantification'. L'erreur de

quantification ΔU_q , sur la tension de mesure U_M ou sur la grandeur physique X , est donnée par:

$$\frac{\Delta U_q}{U_{XMax}} = \frac{\Delta X_q}{X_{Max}} = \frac{1}{D_{Max}} \quad (\text{II.39})$$

avec:

$$D_{Max} = \sum_{m=0}^{n_D-1} 2^m = 2^{n_D}-1 \quad (\text{II.40})$$

où n_D est le nombre de bits de la grandeur digitale.
d'où l'on tire:

$$\frac{\Delta U_q}{U_{XMax}} = \frac{1}{2^{n_D}-1} \quad (\text{II.41})$$

n_D	D_{Max}	$\Delta U_q/U_{XMax}$
8	255	0.003922
10	1023	0.000977
12	4095	0.000244
14	16383	0.000061
16	65535	0.0000153

tab.1 Erreurs de quantification

en grandeurs relatives, on a:

$$\Delta X_q = \frac{X_{Max}}{D_{Max}} = \frac{X_{Max}}{2^{n_D}-1} \quad (\text{II.43})$$

si $x_{\max} = 2.5$, $n_D = 8$ $\implies \Delta x_q = 0.0098 \approx 1\%$.

5 - Codage:

Les calculateurs de processus travaillent presque exclusivement avec le code binaire pur. A chaque bit de la grandeur digitale est alors attribuée une pondération selon une puissance de deux.

exemple: nbre décimale = $27,25 = 16 + 8 + 2 + 1 + 1/4$
 $= 1.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 + 0.2^{-1} + 1.2^{-2}$

1 1 0 1 1 . 0 1

Lors du codage, on ajoute à la grandeur digitale à gauche un bit de signe. Celui-ci vaut 0 pour une grandeur positive, et 1 pour une grandeur négative.

Ainsi le nombre de bits total que présente la grandeur digital est $n = n_D + 1$.

6 - Format:

Pour caractériser la position de la virgule, on doit attribuer à chaque grandeur digitale un format. Celui-ci s'exprime par:

$$(n_{av} + 1).n_{ap} \tag{II.44}$$

où:

n_{av} : nbre de bits (sans bit de signe) avant la virgule.

n_{ap} : nbre de bits apres la virgule.

on a donc:

$$n_{av} + 1 + n_{ap} = n \tag{II.45}$$

un format de 4.12 indique donc une grandeur digitale avec un nbre totale de 16 bits dont 4 (y compris le bit de signe) sont à considérer avant la virgule, et les 12 restants après la virgule.

7 - Simulation:

Lors de la simulation numérique de réglage digitaux, il est très important de simuler aussi l'effet de la quantification. Ceci permet en particulier de détecter l'existence d'éventuels

cycles limites. Le principe de simulation, pour les réglages digitaux, se prête particulièrement bien à la prise en considération de la quantification. En effet, il suffit, en principe, de traiter l'algorithme de réglage en virgule fixe, conformément aux formats pour les variables et les coefficients implantés dans le processeur destiné au réglage digital.

Au niveau du convertisseur A/D, pour une grandeur analogique x , exprimée en virgule flottante, on peut appliquer la relation:

$$D_x = \text{Trunc}[(2^{n_D}-1) \frac{x}{x_{\text{Max}}}] \quad (\text{II.46})$$

D_x est la grandeur digital exprimée comme nombre entier.

n_D est de bits (sans bit de signe) du convertisseur A/D.

x_{Max} est la valeur maximale de x correspondant à $D_{\text{max}} = 2^{n_D}-1$.

L'opération Trunc effectuée la transformation d'une grandeur en virgule flottante, en une grandeur en nombre entier, en tronquant cette dernière après la virgule.

A cette grandeur digitale, il faut attribuer une grandeur quantifiée en virgule fixe x^{**} d'un format $(n_{av} + 1).n_{ap}$, où n_{av} est le nombre de bits (sans bit de signe) avant la virgule, et n_{ap} le nombre de bits après la virgule.

Pour cela, on doit introduire un facteur d'adaptation A_e . [1], on obtient alors:

$$x^{**} = A_e \cdot D_x \quad (\text{II.47})$$

Afin de simplifier la programmation de l'algorithme de réglage pour la simulation, il est judicieux d'effectuer les calculs, non en nombre entier comme en réalité, mais en virgule flottante, tout en tenant compte de la quantification.

Le facteur d'adaptation se calcul alors par:

$$A_e = \frac{1}{2^{n_{ap}}} \text{Round}(2^{n_{ap}}/D_{x1}) \quad (\text{II.48})$$

Round(a) fournit la grandeur a comme grandeur entière arrondie.
 D_{x1} est la grandeur digitale entière pour $x=1$ selon:

$$D_{x1} = \text{Trunc}[(2^{n_p}-1)/x_{\text{Max}}] \quad (\text{II.49})$$

Dans la simulation des algorithmes de réglage, on doit également quantifier les coefficients.

Si un coefficient K est représenté par le format $(n_{sv} + 1).n_{sp}$, le coefficient quantifié devient:

$$K^{**} = \frac{1}{2^{n_{sp}}} \text{Round}(K.2^{n_{sp}}) \quad (\text{II.50})$$

Enfin, au niveau du convertisseur D/A qui sort par exemple le signal de commande u , il faut d'abord transformer la grandeur quantifiée u^{**} en une grandeur entière D_u selon:

$$D_u = \text{Trunc}(A_u u^{**}) \quad (\text{II.51})$$

Lorsque le convertisseur D/A est ajusté de sorte qu'à $u^{**} = 1$ corresponde la grandeur entière D_{u1} , on voit que $A_u = D_{u1}$. Ainsi le signal de commande u à utiliser pour la simulation du système continu devient:

$$u = \frac{1}{D_{u1}} \text{Trunc}(D_{u1}.u^{**}) \quad (\text{II.52})$$

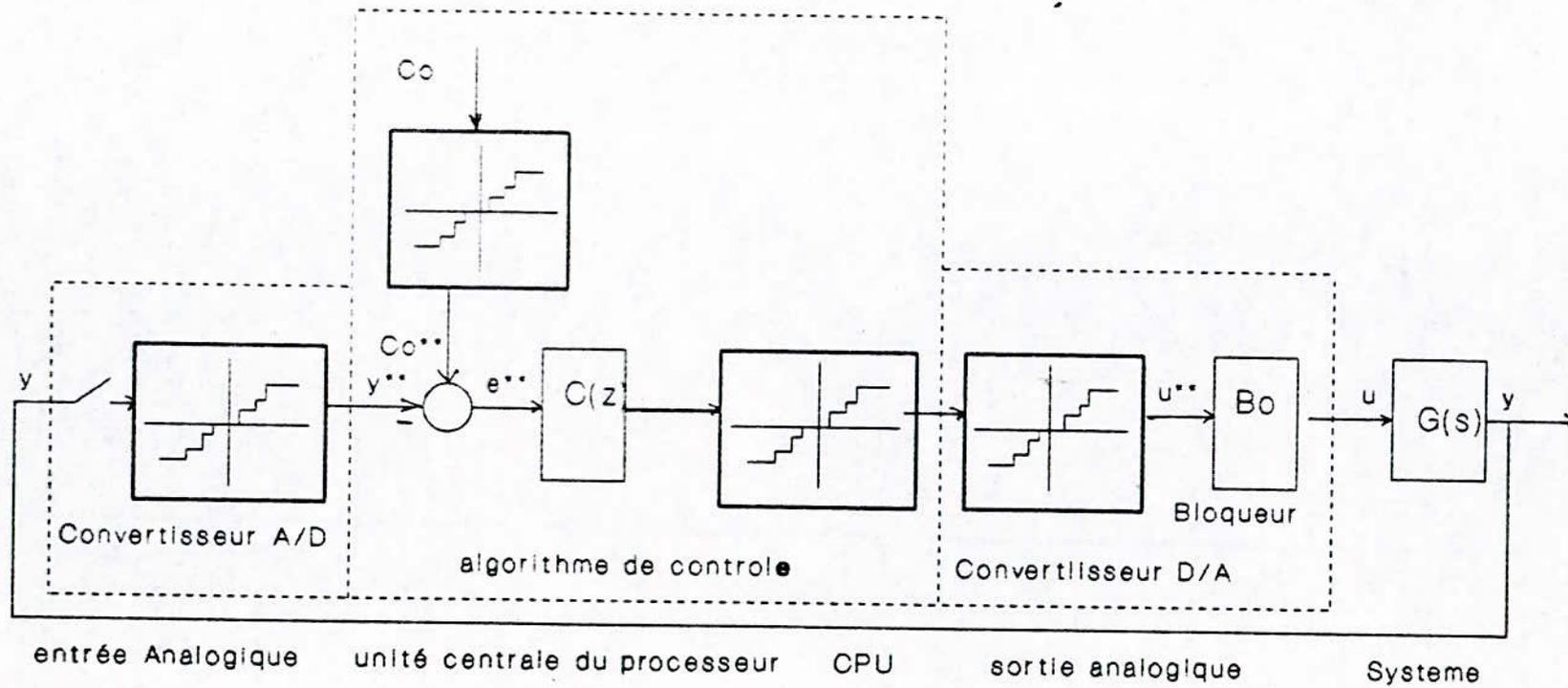


fig 2.6 schémas bloc simplifié montrant les non-linéarités
causées par la quantification, dans une boucle d'asservissement

CHAPITRE III

OPTIMISATION DES PARAMETRES DU REGULATEUR

3 - 1 - Introduction:

Dans ce chapitre, nous allons décrire la procédure de minimisation des paramètres du régulateur.

Après une description détaillée de la discrétisation de l'équation différentielle du régulateur PID continu, nous indiquerons le critère à minimiser afin d'optimiser les paramètres de ce régulateur.

Nous citerons en dernier lieu les différentes méthodes d'optimisation, et mentionerons que nous avons choisi la méthode de Hookes, et Jeeves, car c'est une méthode directe, et ne nécessitant pas le calcul du gradient.

3 -2 - Discrétisation de l'équation différentielle d'un régulateur PID.

Les régulateurs PID permettent d'engendrer à partir de l'écart entre consigne et sortie à régler, une grandeur proportionnelle d'une part à l'erreur, à sa dérivée et à son intégrale.

$$u(t) = K[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt}] \quad (\text{III.1})$$

avec pour paramètres:

K : gain;

T_i : constante d'intégration;

T_D : constante de dérivation.

Cette équation peut être transformée en équation aux différences, en remplaçant la dérivation par une somme.

Le temps d'échantionnage T_s est supposé suffisamment petit. si on utilise l'approximation réctangulaire pour l'intégrale on a:

$$\frac{de(t)}{dt} \approx \frac{\Delta e(k)}{T_e} = \frac{e(k) - e(k-1)}{T_e} \quad (\text{III.2})$$

$$\int_0^t e(\tau) d\tau \approx T_e \sum_{i=0}^k e(i) \quad (\text{III.3})$$

l'équation (III.1) devient donc:

$$u(k) = K \left[e(k) + \frac{T_e}{T_i} \sum_{i=0}^k e(i) + \frac{T_D}{T_e} (e(k) - e(k-1)) \right] \quad (\text{III.4})$$

en identifiant la relation (III.4) à celle donnée par (II.12), on peut écrire:

$$\begin{aligned} K_p &= K \\ K_i &= K T_e / T_i \\ K_d &= K T_D / T_e \end{aligned} \quad (\text{III.5})$$

en écrivant la relation (III.4) sous la forme récurrente, on obtient:

$$u[k] - u[k-1] = q_0 e[k] + q_1 e[k-1] + q_2 e[k-2] \quad (\text{III.6})$$

avec:

$$\begin{aligned} q_0 &= K_p + K_i + K_d \\ q_1 &= -(K_p + 2K_d) \\ q_2 &= K_d \end{aligned} \quad (\text{III.7})$$

On note que parfois, l'approximation de l'intégration est modifiée dans l'équation (III.3), on utilise $e[i-1]$ au lieu de $e[i]$. Dans ce cas les coefficients q_0 et q_1 changent de valeurs.

La fonction de transfert discrète de ce type de régulateur est donnée, comme on la obtenue dans (II.15) par:

$$C(z) = \frac{q_0 z^2 + q_1 z + q_2}{z(z-1)} \quad (\text{III.8})$$

Le régulateur PID discret est un système du deuxième ordre. Sa fonction de transfert possède un pôle $z_1 = 0$ et un pôle $z_2 = 1$.

En principe ce régulateur peut annuler l'écart de réglage en régime permanent.

On remarque que le dénominateur de la fonction de transfert d'un régulateur PID discret et donc les pôles sont indépendants des coefficients K_p , K_i et K_d du régulateur. Ces coefficients n'interviennent que sur le numérateur et par conséquent sur les zéros de la fonction de transfert.

On note que dans le cas où $q_2 = 0$ (ie $K_d = 0$), on obtient un régulateur PI.

3 - 3 - Critère d'optimisation:

Les coefficients q_0 , q_1 et q_2 doivent être ajustés sur le processus de façon à obtenir de bonnes performances de réglage.

Plusieurs méthodes sont ainsi disponibles:

a) On peut obtenir ces paramètres en minimisant un critère de performance selon le modèle du processus.

b) On peut utiliser des règles d'ajustage. Ce qui conduit approximativement à des paramètres de régulateurs optimaux, et ceci si on se base sur certains critères.

c) En commençant par de faibles valeurs (un gain faible), on les fait augmenter jusqu'à obtenir un amortissement acceptable en boucle fermée.

Pour des systèmes simples et des critères pas trop exigés, les méthodes b) et c) suffisent. Par contre, pour avoir des performances suffisantes, on utilise la méthode a), qui est aisément implémentable sur ordinateur.

La question d'évaluation des systèmes de réglage dépend des considérations pratiques. Elle reste souvent un aspect subjectif.

La synthèse des systèmes de réglage à paramètres optimisés, est basée sur l'évaluation d'un critère de performance.

Parmi les différents critères, le plus utilisé est le critère quadratique:

$$S_{eu}^2 = \sum_{k=0}^M (e^2[k] + r \cdot u_o^2[k]) \quad (\text{III.9})$$

$e[k] = C_o[k] - y[k]$ est l'erreur de réglage.

$u_o[k] = u[k] - \bar{u}$ est la variation de la valeur de commande.

avec $\bar{u} = u(\infty)$ valeur finale de la commande.

On peut considérer ainsi les critères quadratiques moyens:

- pour le signal d'erreur:

$$S_e^2 = \overline{e^2[k]} = \frac{1}{M+1} \sum_{k=0}^M e^2[k] \quad (\text{III.10})$$

- pour le signal de commande:

$$S_u^2 = \overline{u_o^2[k]} = \frac{1}{M+1} \sum_{k=0}^M u_o^2[k] \quad (\text{III.11})$$

Le coefficient de pondération r détermine lequel des deux termes est prédominant ou a le plus d'influence.

Si on choisit r petit, S_e^2 est minimisé en fournissant le plus d'énergie S_u^2 . Quand r augmente, l'erreur augmente aussi, et la variation de u est faible.

Dans les régulateurs à paramètres optimisés, les paramètres q_i sont choisis de façon que la valeur de S_{eu}^2 soit minimale.

i.e tels que:

$$\frac{dS_{eu}^2}{dq_i} = 0 \quad (\text{III.12})$$

3 - 4 - Méthodes d'optimisations [8]:

Pour la recherche des paramètres optimaux q_i du régulateur qui minimisent le critère(III.9), il existe plusieurs méthodes:

- Méthodes de recherches simples:

Recherche équidistantes, méthode de Hooke-jeeves, méthode de Rosenbrock,... etc.

- Méthodes utilisant le gradient:

Méthode de Newton-Raphson, méthode de Fletcher et Reeves.

- Combinaison de plusieurs méthodes:

Méthode de Fletcher-Powell.

Parmi ces différentes méthodes, nous allons appliquer la méthode de Hooke-Jeeves.

3 - 5 Méthode de Hooke et Jeeves:

Cette méthode est basée sur l'exploration locale suivant des directions parallèles aux axes de coordonnées.

Cette méthode est directe et ne nécessite pas de calcul des dérivées.

Le principe de la méthode est le suivant:

1 - On choisit un point initial $X^{(0)} \{x_i\}$, et des longueurs de pas δ_i .

2 - On évalue la fonction objectif F (fonction à minimiser).

3 - On établit un réseau de recherche autour de $X^{(k)}$

$\{X_i^{(k)} \pm \delta_i\}$, et on évalue la fonction objectif F .

- si la valeur de F s'améliore (diminue), le déplacement est qualifié de succès et la longueur du pas est augmentée d'un facteur α :

$$\delta_i := \alpha \cdot \delta_i \quad \text{avec} \quad \alpha \geq 1.$$

- si la valeur du F ne s'améliore pas (augmente), le déplacement est qualifié d'échec, la longueur du pas est alors réduite:

$$\delta_i := \beta \cdot \delta_i \quad \text{avec} \quad 0 < \beta < 1.$$

et on refait 3, jusqu'à satisfaire le test d'arrêt.

On recommande de prendre $\alpha=3$ et $\beta=0.5$.

3 -6 - Calcul de la fonction objectif:

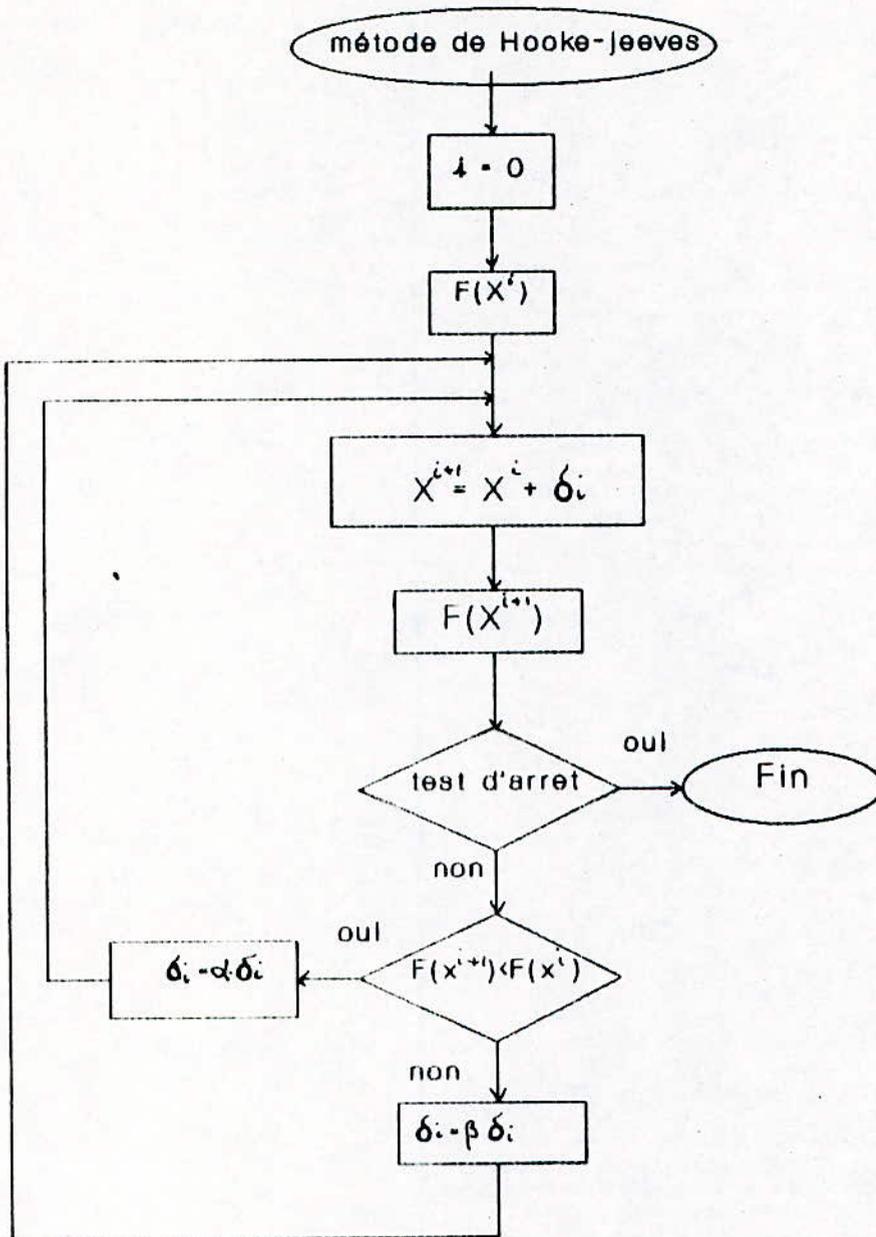
Le calcul de la fonction objectif (critère quadratique) passe par trois étapes:

- 1 - On calcul la réponse $y[k]$ suivant l'organigramme (fig 2.2).
- 2 - On enregistre $u[k]$ et $e[k]$.
- 3 - On calcul S^2 qui est la fonction à minimiser.

Remarque:

La méthode de Hooke et Jeeves est d'autant plus longue que l'ordre du système est élevé, il est conseillé alors d'utiliser un micro ordinateur équipé d'un coprocesseur arithmétique.

3 - 7 - Organigramme de la méthode de Hooke-Jeeves:



troisieme Partie

Exemples

1 - Introduction:

Dans cette partie, on va présenter quelques exemples afin d'illustrer l'intérêt du logiciel.

Tout d'abord, on présentera un exemple simple qui montrera l'effet de la quantification. Ensuite on essaiera d'optimiser les paramètres d'un régulateur PID pour un système à phase non minimale.

2 - Quantification:

Pour simuler l'effet de la quantification (2-2-7), nous avons pris un système du 1^{er} ordre:

$$G(s) = \frac{1}{1+s} \quad (1)$$

avec un régulateur proportionnel :

$$C(z) = K_p = 1 \quad (2)$$

Les convertisseurs A/N et N/A ont 8 bits et une valeur maximale de 10.

Pour l'algorithme de réglage, on utilise un calculateur de 8 bits; format (2.6).

La fig 1 montre la réponse temporelle en B.F sans tenir compte de la quantification.

La fig 2 montre la réponse temporelle en B.F avec quantification. On remarque, un cycle limite en régime permanent, avec en plus un biais important. Celà est dû à la quantification.

3 - Optimisation des paramètres du régulateur [7]:

Nous avons étudié un système intéressant, vu qu'il présente un undershoot; c'est un système à phase non minimale. Sa fonction de transfert est donnée par:

$$G(s) = \frac{1-4s}{(1+4s)(1+10s)} \quad (3)$$

la fig 3 montre la réponse indicielle de ce système. On utilise un régulateur discret PID de la forme:

$$C(z) = \frac{q_0 z^2 + q_1 z + q_2}{z(z-1)} \quad (4)$$

on essaye d'optimiser les paramètres q_0, q_1, q_2 de ce régulateur en minimisant le critère donné par (III.9)

$$S_{eu}^2 = \sum_{k=0}^M (e^2[k] + r \cdot u_e^2[k]) \quad (5)$$

et cela par la méthode de Hooke et Jeeves (3-5).

Nous avons étudié l'influence de la période d'échantillonnage et l'influence du facteur de pondération sur la qualité du réglage.

2-1 - Influence de la période d'échantillonnage:

T_e	1	4	8	16
q_0	5.94	2.30	1.79	1.53
q_1	-10.31	-3.04	-1.81	-0.86
q_2	4.48	1.10	0.62	0.25

tab 1

Les figures 4, 5, 6 et 7 montrent les valeurs discrètes de la commande et les réponses temporelles du système en B.F correspondant, pour des périodes d'échantillonnages $T_e = 1, 4, 8$ et 16 sec et pour $r = 0$.

Pour la plus petite période d'échantillonnage $T_e = 1$ sec, on obtient une approximation de la commande semblable à celle obtenue par un régulateur PID continu. Pour $T_e = 4$ sec, le signal continu de la commande peut être estimé bon. Par contre pour les valeurs de $T_e = 8$ sec et 16 sec, le signal de commande ne peut être considéré comme continu.

Le tableau 1 montre les paramètres optimaux q_0, q_1 et q_2 .

Pour $T_e = 1$ sec, on a un undershoot de -0.47 et un temps de réponse à 5 % de 18.5 sec.

Par contre pour $T_e = 4$ sec, on a un undershoot de -0.24 et un temps de réponse à 5 % de 44 sec.

On va prendre le cas où l'undershoot est le plus petit et on va essayer de diminuer le temps de réponse, et cela en variant la valeur du facteur de pondération r dans (5).

2 - 2 - Influence du facteur de pondération r pour $T_e=4$ sec:

r	0	0.1	0.25
q_0	2.30	2.15	1.63
q_1	-3.04	-2.75	-1.96
q_2	1.10	0.94	0.61
$tr_{5\%}$	44	31	24
undershoot	-0.24	-0.22	-0.17

tab 2

Les figures 5, 8 et 9 montrent les réponses temporelles en B.F pour différents valeurs de r et pour $T_e = 4$ sec.

Le tableau 2 montre les paramètres q_0, q_1 et q_2 optimaux pour $T_e = 4$ sec ainsi que les temps de réponses à 5 % et les undershoots.

Pour $r = 0.1$, on a un temps de réponse à 5 % qui vaut 31 sec et un undershoot de -0.22 . Par contre pour $r = 0.25$, on a un temps de réponse à 5 % inférieur au précédent et qui vaut 24 sec, mais un undershoot moins important que celui pour $r = 0.1$.

On a donc ainsi optimisé et amélioré les performances de notre système et cela en prenant $T_e = 4$ sec et $r = 0.25$ et en utilisant un régulateur PID :

$$C(z) = \frac{1.63z^2 - 1.96z + 0.69}{z(z-1)}$$

CONCLUSION

Le travail réalisé est un exemple parmi d'autres pour l'automatique.

Ce logiciel permet:

- La simulation en boucle ouverte.
- La simulation de la boucle d'asservissement.
- La simulation du compensateur digital.
- L'illustration d'un phénomène important qui est la quantification des paramètres du compensateur.
- Optimiser les paramètres du compensateur digital pour certaines structures, et cela en minimisant un critère quadratique.

Ce logiciel présente l'avantage de pouvoir, sans changer à chaque fois, la fonction de transfert du système, de faire l'étude (simulation et optimisation) pour différentes périodes d'échantillonnage et pour différents types de compensateur. Comme il permet également de montrer à l'utilisateur, le comportement de la boucle d'asservissement, et cela entre les instants d'échantillonnage.

Tout au long de ce travail, nous avons cherché à utiliser les méthodes et les algorithmes les plus performants, afin d'obtenir de bons résultats en un temps minimal.

Nous avons consacré notre étude sur la conception d'un programme interactif avec l'utilisation d'un menu. Celui-ci permet une communication interactive entre l'utilisateur et le logiciel.

Nous avons pu également réaliser une représentation graphique, ce qui permet à l'utilisateur l'interprétation aisée des courbes.

La disponibilité des résultats numériques, et des différentes courbes sur l'écran et sur l'imprimante, et l'une des caractéristiques de ce logiciel.

Les résultats des exemples traités donnent une idée sur l'efficacité des algorithmes utilisés.

Nous avons réalisé ce logiciel tout en ayant la conviction qu'il trouvera une large application et utilisation au laboratoire d'automatique de notre école.

Enfin, notons que ce logiciel est appelé à être amélioré, et ce en considérant les diverses remarques qu'auront faites les utilisateurs. De plus, il peut être partie intégrante d'un système interactif qui traiterait divers types de systèmes avec d'autres représentations (représentation d'état et transformée en z), ainsi que diverses méthodes d'optimisation (méthodes utilisant le gradient...etc).

Annexe

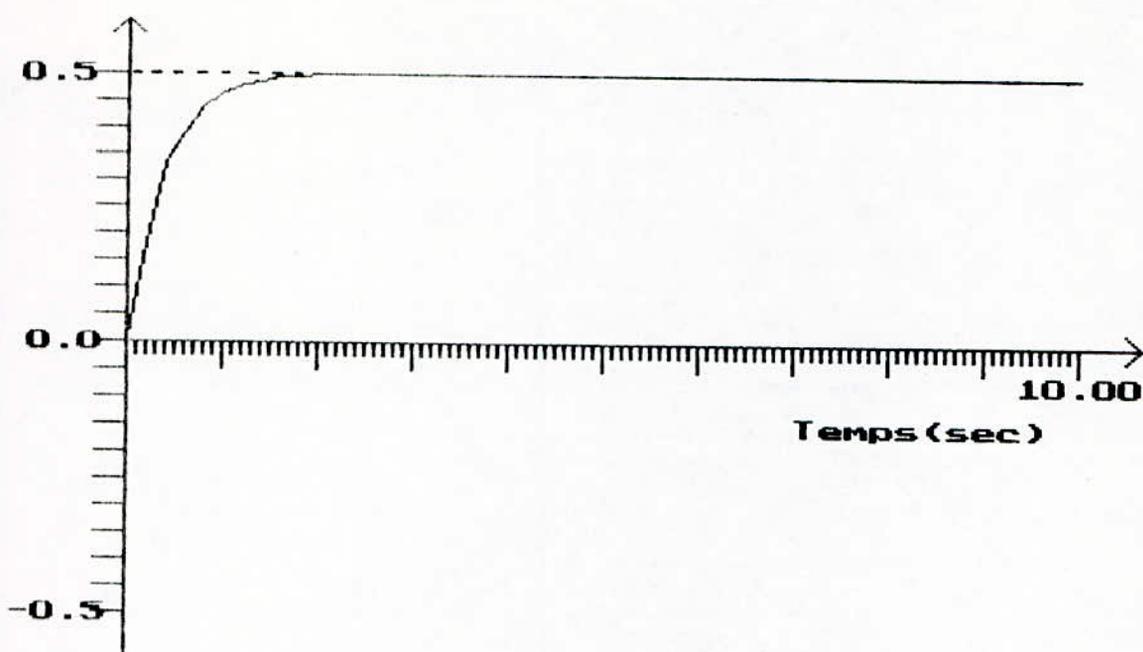


fig 1 Réponse Temporelle en B.F

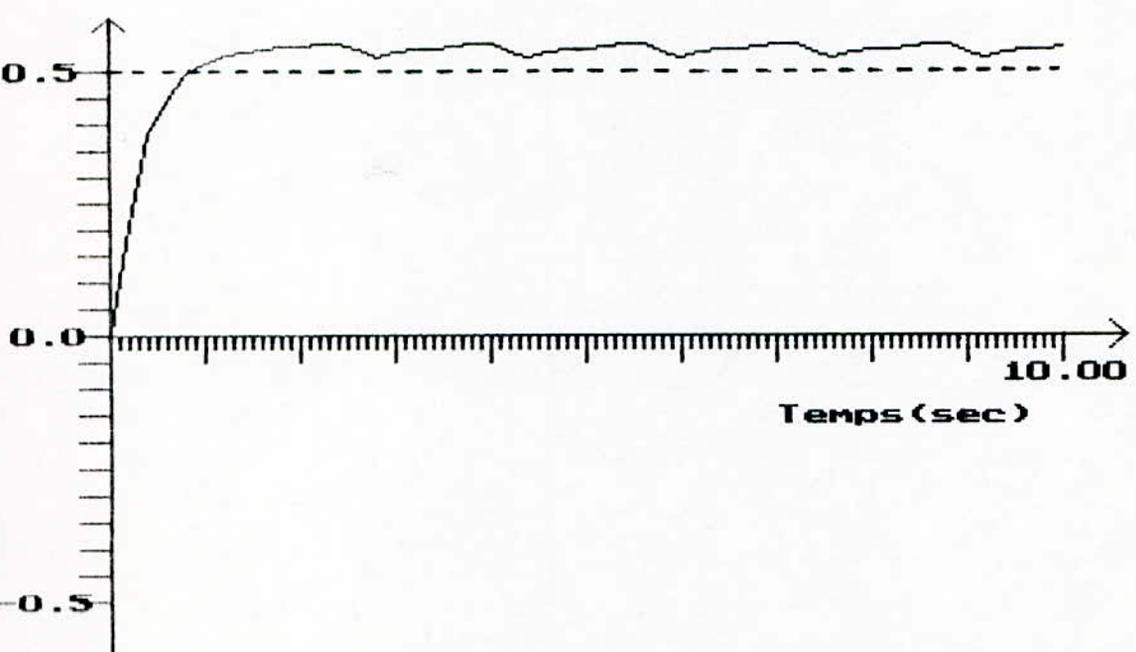


fig 2 Réponse Temporelle en B.F

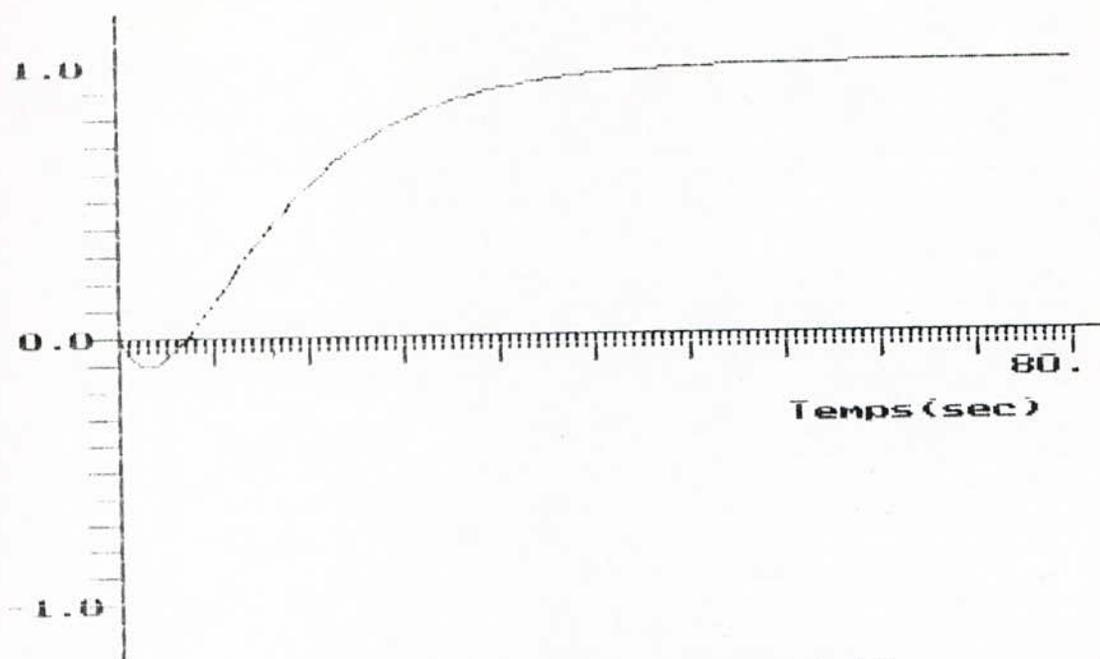
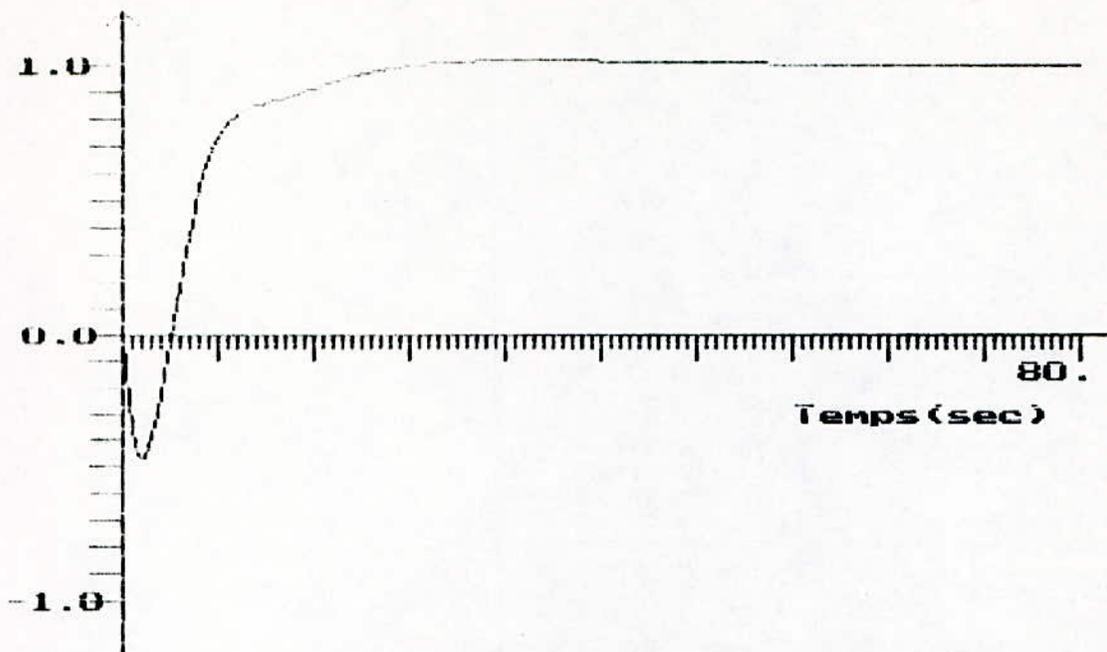
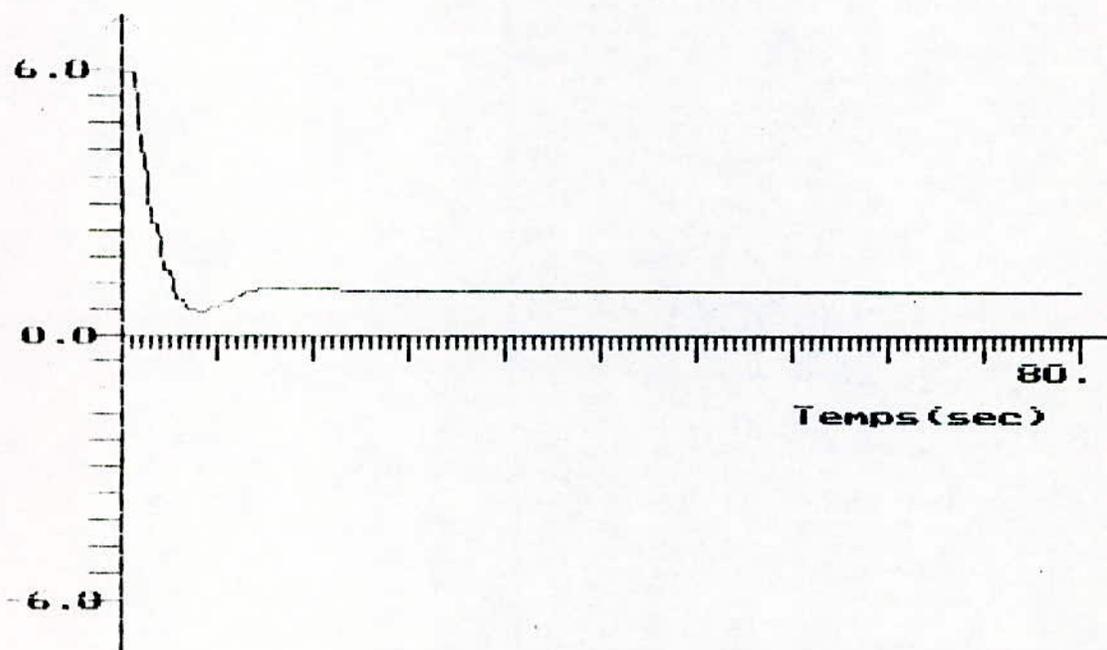


fig3 Réponse indicielle

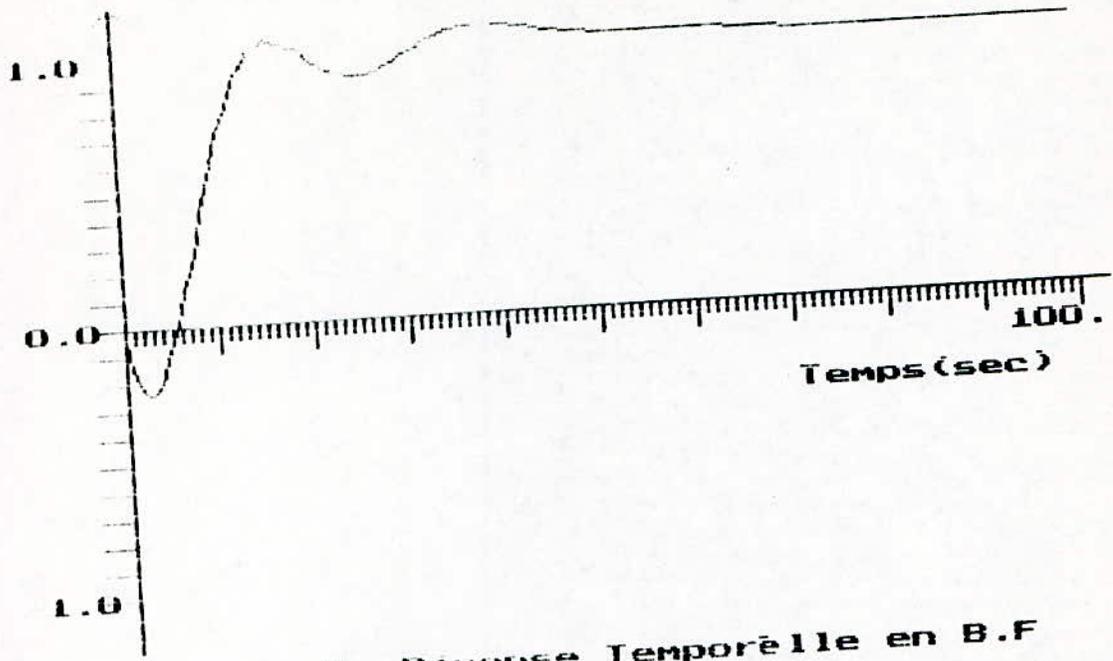


a. Réponse Temporelle en B.F

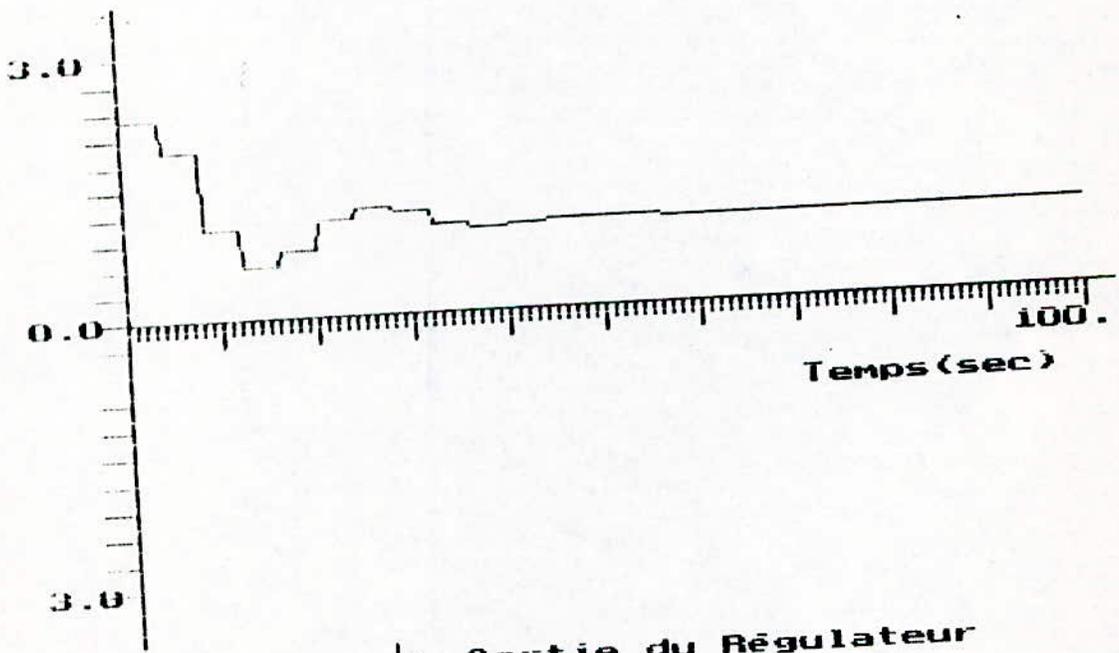


b. Sortie du Régulateur

fig 4

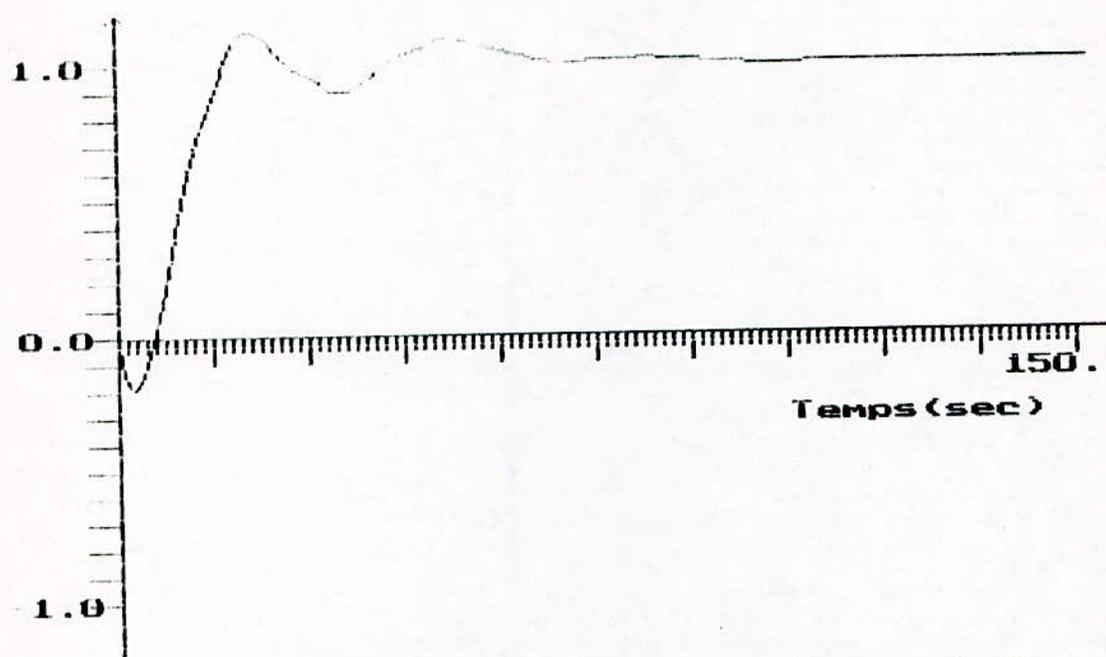


a. Réponse Temporelle en B.F

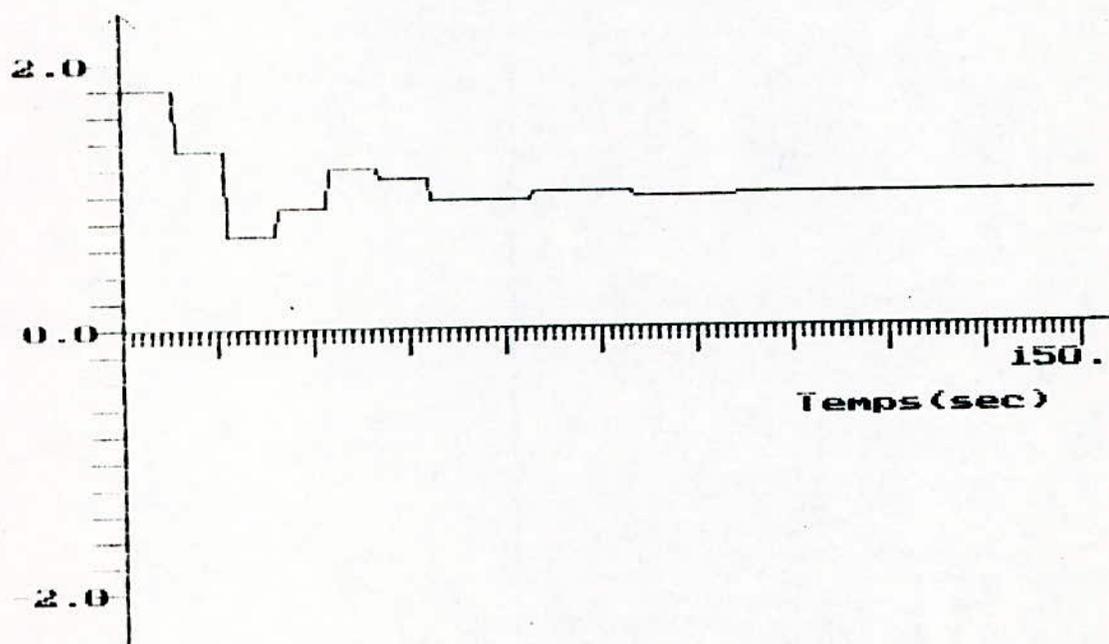


b. Sortie du Régulateur

fig 5

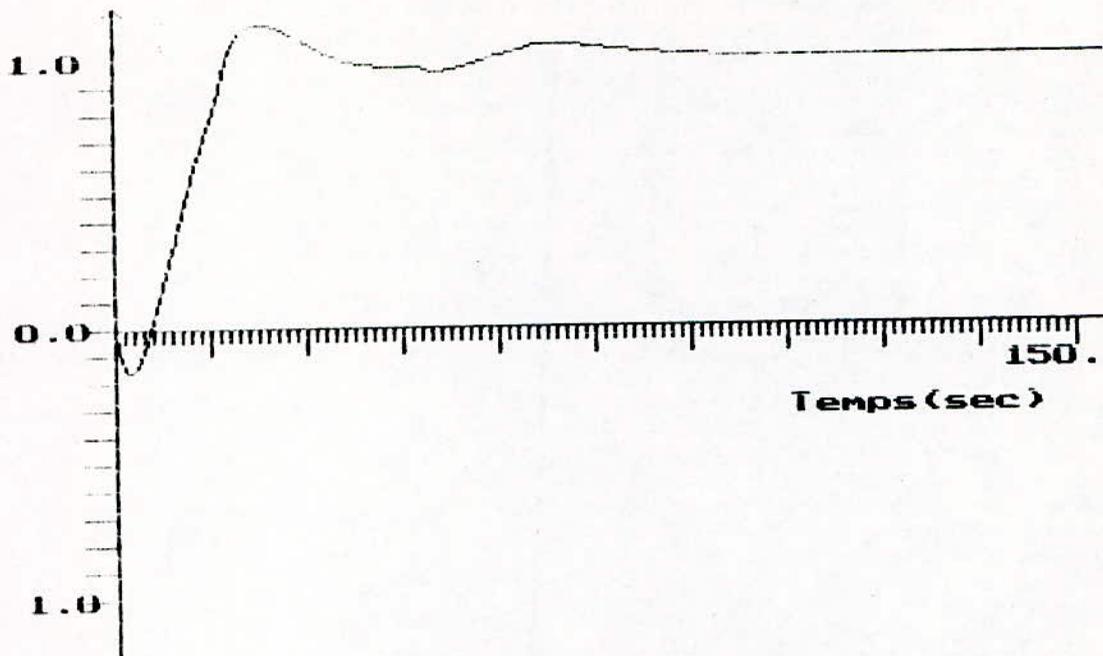


a. Réponse Temporelle en B.F

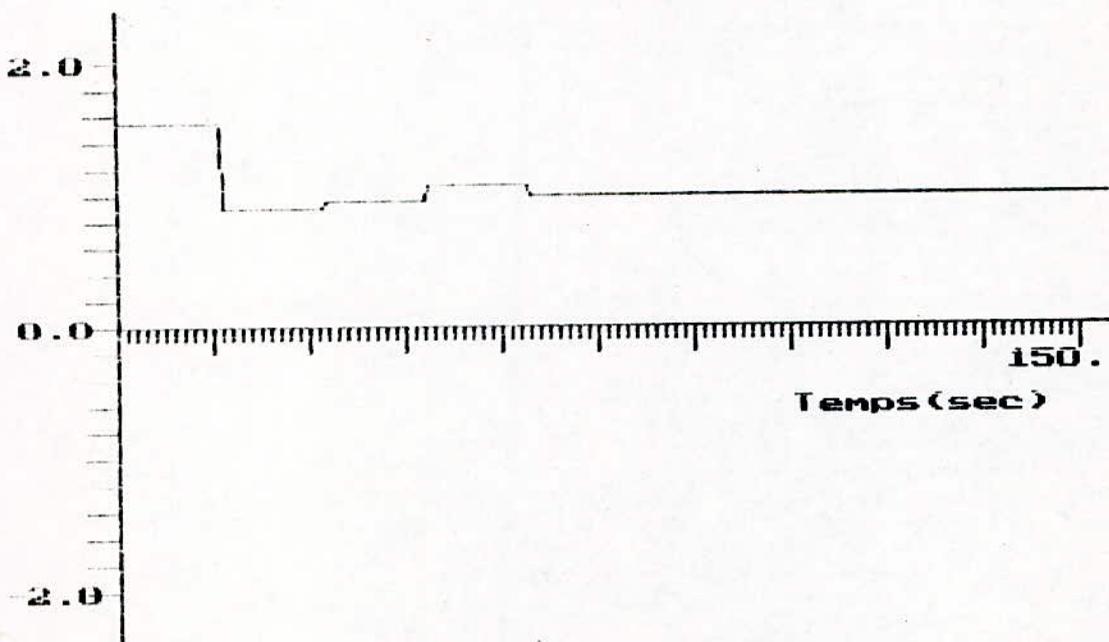


b. Sortie du Régulateur

fig 6

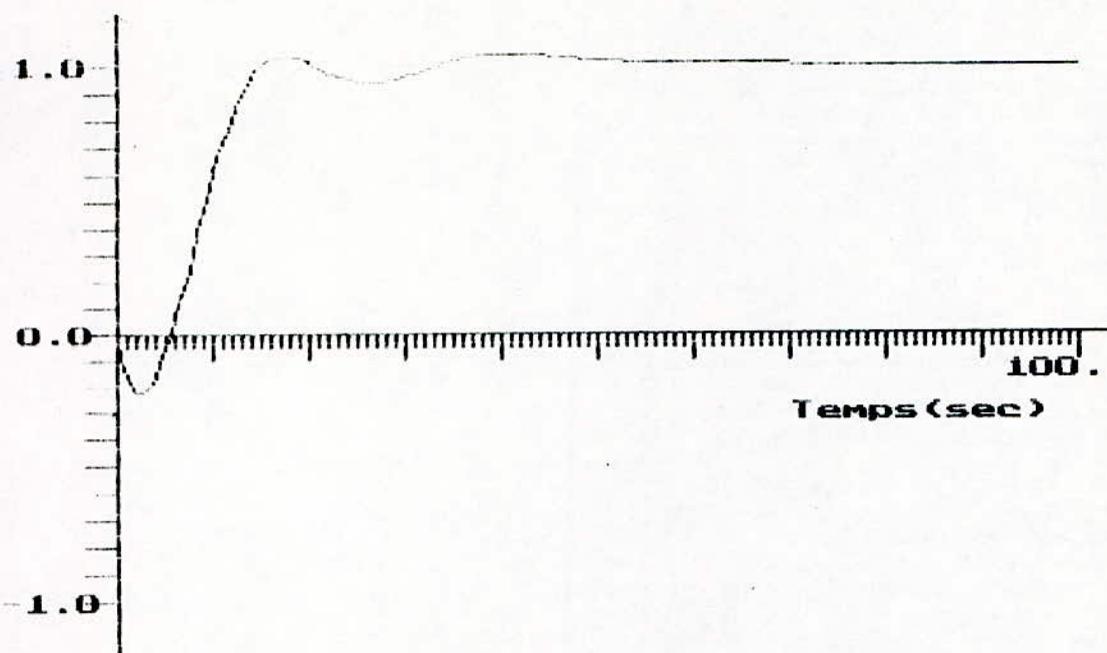


a. Réponse Temporelle en B.F

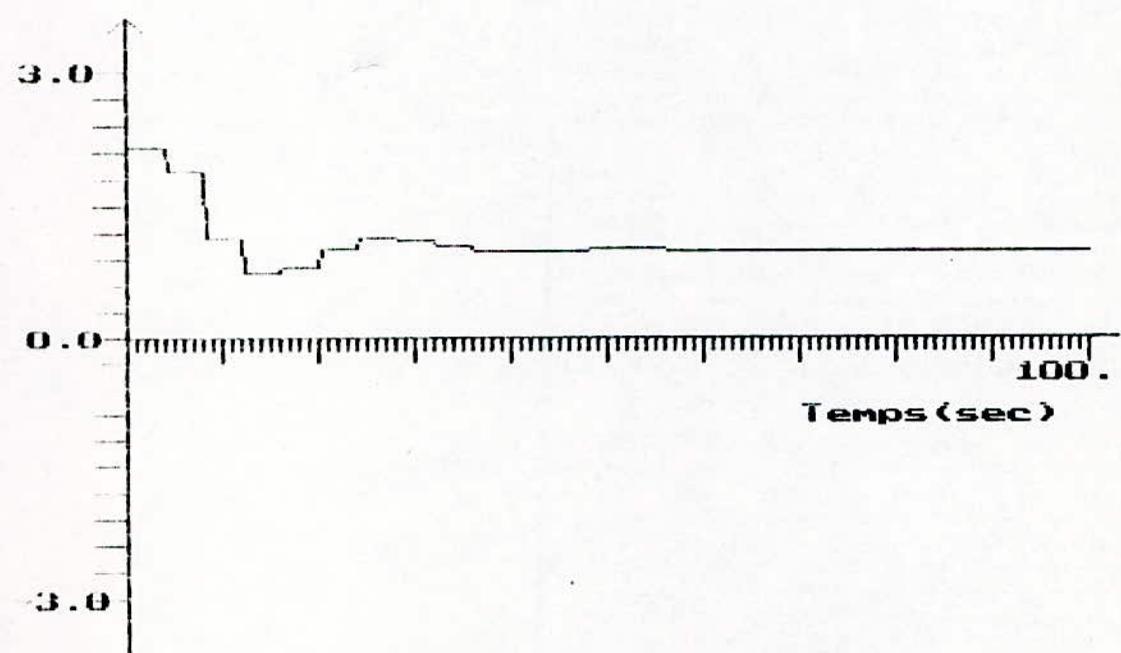


b. Sortie du Régulateur

fig 7

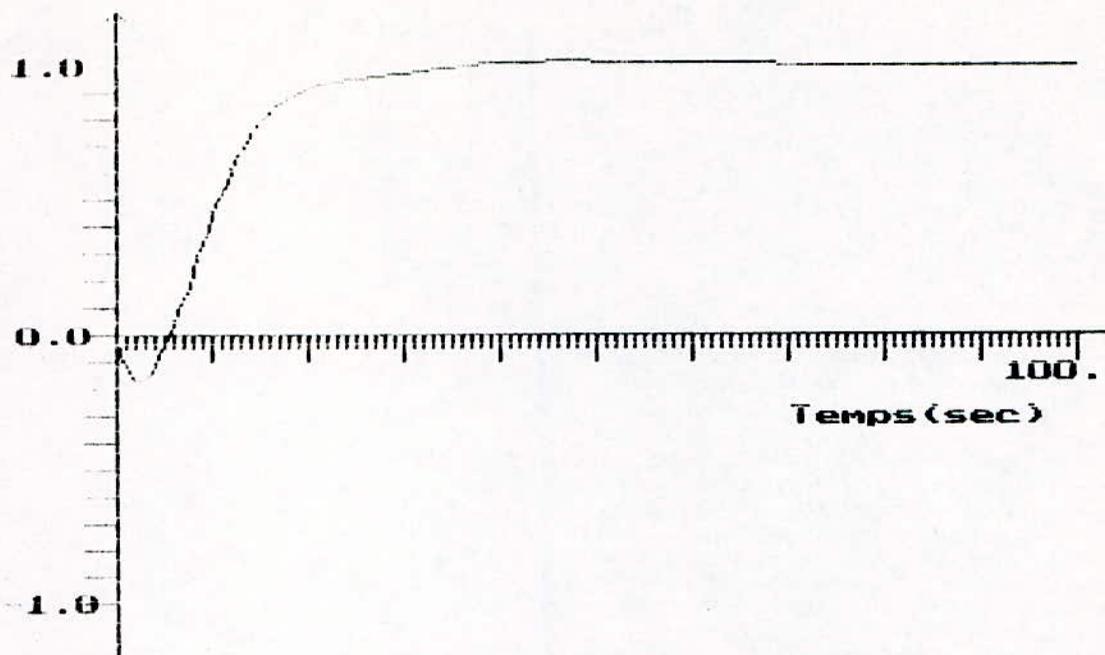


a. Réponse Temporelle en B.F

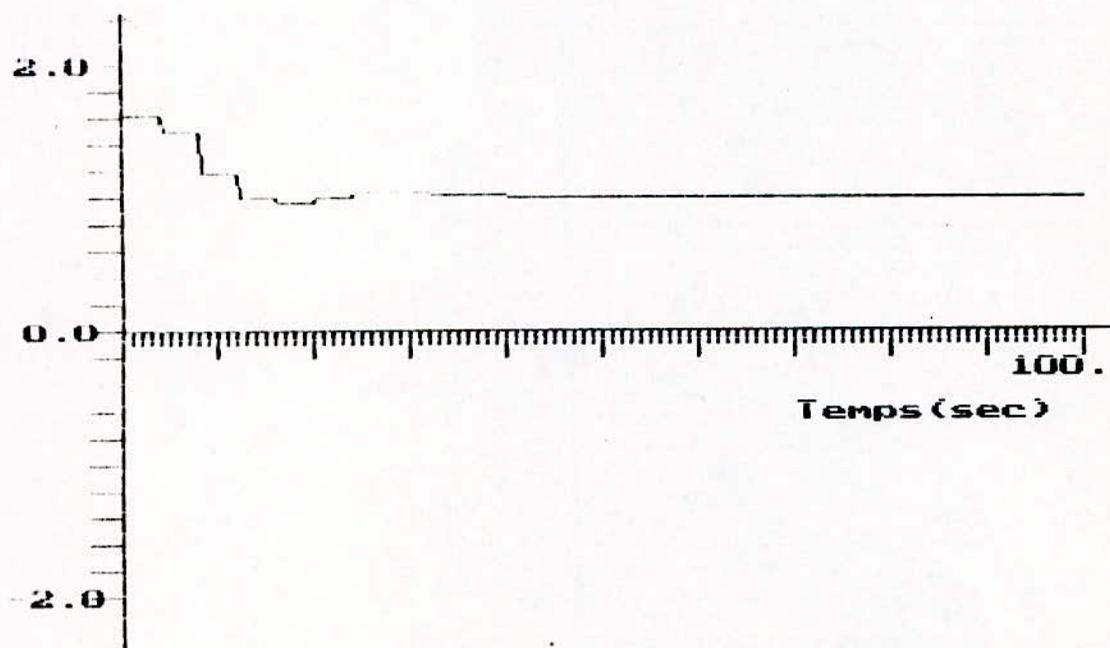


b. Sortie du Régulateur

Fig 8



a. Réponse Temporelle en B.F



b. Sortie du Régulateur

fig 9

BIBLIOGRAPHIE

- [1] H. Bühler Conception de Systèmes Automatiques
ed. Presse Polytechnique Romande 1988
- [2] H. Bühler Réglages Echantillonnés vol 1 : Traitement
par la transformation en Z ed. Presse
Polytechnique Romande 1986
- [3] H. Bühler Réglages Echantillonnés vol 2 : Traitement dans
l'espace d'état ed. Presse Polytechnique
Romande 1983
- [4] C. Nowakowski Méthde de Calcul Numérique Tome 1 ed. PSI
1984
- [5] B.C. Kuo Digital Control Systems ed. M.E.Van
Valkenburg 1980
- [6] L. Maret Régulation Automatique ed. Presse
Polytechnique Romande 1987
- [7] R. Iserman Digital Control Systems ed. Spring-Verlag-
Heidelberg New-York 1981
- [8] S.G. Beveridge
R.S. Schechter Optimisation : Theory and Practice ed.
Mc Graw - Hill 1970
- [9] D. Maille Le langage Pascal ed. Berti 1990
- [10] H.L. Planchat Turbo Pascal Version 5.0 et 5.5 ed. Eyrolles
1990