

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERY, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

Master académique en automatique
Option commande des systèmes

Thème

*Conception et réalisation d'un
correcteur numérique en utilisant le
PIC 16F876*

Proposé et Dirigé par : M. KARA

Présenté par : AIT MOHAMMED Ferhat

Soutenu le : 20 / 09 /2012

IMAZATENE Hacene

Promotion 2012

Ce travail a été préparé à l'université de Mouloud MAMMERY de Tizi-Ouzou



Remerciements

En premier lieu, nous tenons à remercier notre DIEU, notre créateur pour nous avoir donné la force pour accomplir ce travail.

*Nos sincères remerciements à notre promoteur M. **KARA** de nous avoir guidé et encouragé durant ce travail.*

Nous tenons aussi à remercier tous les membres du jury d'avoir accepté de corriger ce modeste travail.

*Nous présentons nos chaleureux remerciements aux enseignants du département Automatique en particulier M. **MAIDI** pour leurs aides et orientations durant notre formation.*

Un très grand merci pour nos parents

Nos derniers remerciements et ne sont pas les moindres, vont à tous ceux qui ont contribué de près ou de loin pour l'aboutissement de ce travail.

Ferhat et Hacene





Dédicaces

Je dédie ce modeste mémoire :

A ceux qui m'ont donné le sens de la vie

Mon Père et ma Mère, mes chères sœurs Nadia, Zahra, Dahbia, Chafia et Chabha.

Et à tous mes amis(es) et toutes les personnes qui m'ont aidé et soutenu d'une façon ou d'une autre.

Et toute la promo surtout mon meilleur ami et binôme Hacene.

Ferhat





Dédicaces

Je dédie ce modeste mémoire :

A ceux qui m'ont donné le sens de la vie

Mon Père et ma Mère, mes chères sœurs et frères

Mon grand frère, sa femme et ses enfants Thiziri et Mazighi

Et à tous mes amis(es) et toutes les personnes qui m'ont aidé et soutenu d'une façon ou d'une autre.

Et toute la promo surtout mon meilleur ami et binôme Ferhat.

Hacène



Table des matières

Introduction générale..... 01

Chapitre I généralités sur les systèmes à temps discrets

I.1 Introduction.....02

I.2 Signaux échantillonnés et discrets.....02

 I.2.1 Signal échantillonné..... 02

 I.2.2 Signal discret..... 03

I.3 Modélisation des systèmes à temps discret..... 04

 I.3.1 Rappel sur les systèmes continus.....04

 I.3.2 Cas des systèmes à temps discret.....04

 I.3.2.1 Description par les équations récurrentes.....04

 I.3.2.2 Description par la transformée en Z.....05

I.4 Echantillonnage et reconstitution du signal.....05

 I.4.1 Les éléments échantillonneurs..... 05

 I.4.1.1 Echantillonneur idéal..... 05

 I.4.1.2 Echantillonneur à pulsation (ou échantillonneur)..... 06

 I.4.1.3 Quantification..... 06

 I.4.2 Reconstitution d'un signal échantillonné.....07

 I.4.2.1 Théorème de Shannon.....07

 I.4.2.2 Echantillonneur bloqueur.....08

I.5 Fonction de transfert discrète.....10

 I.5.1 Représentation symbolique d'un système à temps discret.....10

 I.5.2 Notion de fonction de transfert échantillonnée.....10

 I.5.2.1 Convolution discrète.....10

 I.5.2.2 Fonction de transfert échantillonnée.....11

Table des matières

I.6 Stabilité des systèmes à temps discret.....	12
I.6.1 Notion générale de la stabilité.....	12
I.6.1.1 Rappels.....	12
I.6.1.2 Cas des systèmes discret.....	13
❖ Critère de Nyquist.....	14
❖ Critère de jury.....	14
❖ Lieu des racines ou lieu d'EVANS.....	14
I.6.1.3 Marge de stabilité.....	15
I.7 Précision et rapidité des systèmes discrets.....	16
I.7.1 Définition.....	16
I.7.2 Précision statique.....	16
I.7.2.1 Constante d'erreur.....	16
I.7.3 La rapidité.....	17
I.7.3.1 Considération sur le régime transitoire.....	17
I.7.3.2 Rapidité des systèmes discrets.....	17
I.7.4 Systèmes contraints.....	17
I.7.4.1 Dilemme précision rapidité stabilité.....	17
I.7.4.2 Système minimal.....	18
I.7.4.3 Système à réponse pile.....	18
I.8 Conclusion.....	19

Chapitre II **Notion de la régulation et synthèse des régulateurs
numérique**

II.1	Introduction.....	21
II.2	Principe générale de la régulation.....	21
II.2.1	Comportement en régulation.....	23
II.2.2	Comportement en asservissement.....	23
II.3	Numérisation des correcteurs analogique.....	23
II.4	Faisabilité des correcteurs.....	25
II.5	Critère du choix de correcteur.....	27
II.5.1	Condition générales et critère temporels.....	27
II.5.2	Méthode de calcul.....	28
II.7	Les régulateurs discrets standards.....	31
II.7.1	Généralités.....	31
II.7.2	Les différentes actions.....	32
II.7.2.1	Action proportionnelle P.....	33
II.7.2.2	Action intégrale I.....	33
II.7.2.3	Action dérivée D.....	34
II.7.3	Les régulateurs standards.....	34
II.7.3.1	Le régulateur proportionnel-intégral PI.....	34
II.7.3.2	Correcteur à retard de phase.....	35
II.7.3.3	Le régulateur proportionnel-dérivé PD.....	37
II.7.3.4	Correcteur à avance de phase.....	37
II.7.3.5	Le régulateur universel PID.....	39
II.7.4	Choix et dimensionnement des régulateurs PID numériques.....	41
II.7.4.1	La boucle de régulation.....	41

Table des matières

II.7.4.2 Réglage des coefficients.....	42
II.7.4.3 Approche par modélisation.....	43
II.7.4.4 Approche par expérimentation.....	44
II.7.4.4.1 Méthode d'oscillation limite en BF(Ziegler-Nichols).....	44
II.7.4.2.2 Méthode de la réponse indicielle (Ziegler-Nichols en BO).....	45
II.7.5 Limites et amélioration du régulateur.....	46
II.7.5.1 Contraintes électrique et matérielles.....	47
II.7.5.2 Instabilité de la dérivation.....	47
II.8 Conclusion.....	48

Chapitre III

conception et réalisation pratique

III.1 Introduction.....	50
III.2 Introduction.....	50
III.3 Principe de fonctionnement.....	50
III.4 Guide d'utilisation.....	51
III.5 Système à microprocesseur.....	52
III.5.1 Définition d'un microcontrôleur.....	52
III.5.1.1 L'unité de traitement (CPU : Central Processing Unit).....	53
III.5.1.2 Les mémoires.....	53
III.5.1.3 L'horloge.....	54
III.5.1.4 Les ports d'entrée / sortie.....	54
III.5.2 Description du 16F876.....	54
III.5.2.1 Caractéristiques essentiel du 16F876.....	55
III.5.2.2 Structure interne de microcontrôleurs 16F876.....	56
III.5.2.3 Brochage du 16F876.....	57

III.5.2.4	Le convertisseur analogique numérique du 16F876.....	57
III.5.2.4.1	Résolution d'un convertisseur analogique numérique.....	57
III.5.2.4.2	Détail de fonctionnement du CAN contenu dans 16F876.....	58
III.5.2.4.3	Structure interne du convertisseur du 16F876.....	59
III.5.2.4.4	La fréquence d'échantillonnage.....	59
III.5.3	Convertisseur numérique analogique (DAC 0800 / DAC 0802).....	59
III.5.3.1	Caractéristiques principales du DAC 0800 DAC0802.....	60
III.5.3.2	Brochage de DAC 0802.....	60
III.5.3.3	Structure interne de DAC 0802.....	61
III.6	Conception matériel.....	61
III.6.1	Choix des composants.....	61
III.6.2	Schéma synoptique.....	62
III.6.3	Etage d'adaptation du signal d'entrée.....	63
III.6.4	Etage de conversion.....	64
III.6.4.1	Temps de conversion.....	64
III.6.4.2	Temps d'acquisition.....	65
III.6.5	Etage d'adaptation du signal de sortie.....	66
III.6.6	Le clavier.....	68
III.6.7	L'afficheur LCD.....	69
III.6.7.1	Rôle d'un afficheur LCD.....	69
III.6.7.2	La mise en œuvre d'un afficheur LCD.....	69
III.6.7.3	Schéma fonctionnel de l'afficheur LCD.....	69
III.6.7.4	Rôle des différentes broches de l'afficheur LCD.....	70
III.6.8	Le reset et l'oscillateur.....	70
III.7	Conception logiciel.....	71
III.7.1	Organigramme.....	71

Table des matières

III.7.2 Langage de programmation.....	71
III.7.3 Présentation générale de Logiciel de simulation utilisé PROTEUS.....	72
III.8 Tracé du circuit imprimé sous ARES	73
III.9 Circuit complet de la réalisation sous ISIS.....	74
III.10 Procédures de Test.....	75
III.10.1 Tests souhaités.....	75
III.10.2 Tests obtenu	76
III.11 Conclusion.....	77
Conclusion générale.....	78

INTRODUCTION
GENERALE

De nos jours, les systèmes physiques commandés par des régulateurs numériques sont très répons dans l'industrie. En effet, la majorité des procédés industriels nécessitent le contrôle d'un certain nombre de grandeurs physiques tel que : la température, la pression, le débit...etc. afin que la grandeur mesurée d'un processus puisse suivre une trajectoire désirée, en présence de perturbations éventuelles. L'un des objectifs premiers d'un automaticien est l'étude, la conception et la réalisation du régulateur approprié à chaque système.

L'objectif de ce travail est, la conception et la réalisation d'un régulateur à base d'un microcontrôleur, PIC 16F876. Ce régulateur, effectue des fonctions de corrections numérique, PID, avance et retard de phase. Selon le correcteur choisi, les paramètres K_p , K_i et K_d sont modifiable par l'utilisateur à tout instant. Ceci confère au régulateur plus de performance et plus de flexibilité.

Ce mémoire est organisé comme suit :

Au premier chapitre, nous avons fait quelques rappels essentiels par rapport aux systèmes discrets. La modélisation des systèmes discrets, notion de de fonction transfert et de la stabilité de tels systèmes abordés.

Le deuxième chapitre, traite les régulateurs numériques et leurs objectifs. Nous commençant ainsi par introduire les actions proportionnelles, intégrales et dérivée, et les différents régulateurs PI, PD, PID, et avance retard de phase.

Le troisième chapitre est consacré à la partie réalisation pratique. Dans un premier lieu, on a abordé quelque généralité sur les systèmes à microprocesseurs. Par la suite, on explique le rôle de chaque composant utilisé ainsi que les étapes de réalisation .on terminent ce chapitre par des tests et résultats.

Nous terminant notre mémoire par une conclusion générale et quelques perspectives.

CHAPITRE 1

1. Introduction

Dans le but d'optimiser le fonctionnement d'une commande d'un système automatique on a fait appel aux systèmes discrets.

Alors on se propose d'illustrer quelques notions qui nous seront très utile dans notre mémoire tel que l'échantillonnage, la reconstruction du signal et la modélisation.

2. Signaux échantillonnés et discrets

2.1 Signal échantillonné

Dans le contexte de l'informatisation d'une partie du processus et plus généralement du traitement de l'information, échantillonner un signal analogique signifie, le remplacer par une suite de ses valeurs prises à des instants bien définis. L'échantillonnage joue un rôle capital en réglage automatique du fait qu'un ordinateur traite des nombres plutôt que des grandeurs analogiques.

En analogique, on a l'habitude de traiter des signaux continus, présents à tout instant. Le signal $x(t)$ de ce type possède une énergie finie, correspondant à son intégrale sur un laps de temps fini :

$$w(t) = \int_0^t x^2(\tau) d\tau \quad (1.1)$$

Le signal échantillonné $x_k(t)$, associé au signal continu $x(t)$, est composé d'une série d'impulsions de Dirac apparaissant uniquement aux instants d'échantillonnage. Par la suite, on supposera que l'échantillonnage est périodique, de période T.

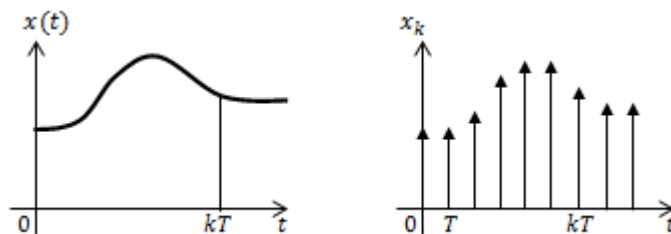


Figure (1.1): Echantillonnage d'un signal continu

On rappelle que l'impulsion de Dirac (impulsion-unité) est une distribution. Cette impulsion de Dirac s'obtient par un passage à la limite d'une impulsion rectangulaire qui tendra vers une impulsion infiniment étroite et d'amplitude infinie tout en gardant une surface constante, correspondant à l'énergie véhiculée par le signal à l'instant d'échantillonnage. L'intégrale d'une impulsion-unité est égale à l'unité.

2.2 Signal discret

Un signal discret consiste en une séquence de valeurs qui ne sont définies qu'aux instants d'échantillonnage. En dehors des instants d'échantillonnage, le signal discret n'existe pas. Un tel signal sera représenté par une ligne pointillée et un point à une ordonnée correspondant à la valeur numérique du signal $x(t)$ continu associé, pris à l'instant d'échantillonnage considéré.

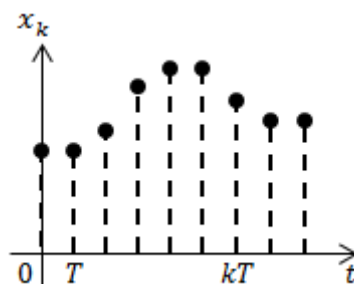


Figure (1.2): Signal discret

Le signal discret $x_k(t)$, associé au signal continu $x(t)$, a pour valeur à l'instant kT :

$$x_k = x(kT) \quad (1.2)$$

L'énergie d'un signal discret est nulle. Avec un signal discret, on peut faire toutes les opérations arithmétiques (addition, soustraction, multiplication et division). Par contre, on ne peut pas exciter avec un tel signal un système continu (parce que l'énergie du signal est nulle).

3. Modélisation des systèmes à temps discret

3.1 Rappel sur les systèmes continus

Les signaux qui transitent à travers les différents éléments d'un système de commande subissent des transformations à tous les stades, que celles-ci soient volontaires (par exemple, traitement par filtrage, ou analyse spectrale, pour en extraire une information) ou non. Pour un système S quelconque, ceci est pris en compte par la relation entrée-sortie qui décrit le système.

Rappelons qu'un système continu est modélisé par sa fonction de transfert $H(p)$ ou sa réponse impulsionnelle $h(t)$, obtenues par partie de sa description par des équations différentielles.

$$Y(p) = H(p).U(p) \quad (1.3)$$

$$y(t) = h(t) * u(t) \quad (1.4)$$

3.2 Cas des systèmes à temps discret

3.2.1 Description par les équations récurrentes

Dans le cas de signaux n'apparaissant qu'à des instants précis du temps (et nuls dans les intervalles), on a affaire à des systèmes à temps discret (qualifiés souvent par contraction de systèmes discrets) dont la relation entrée-sortie peut s'écrire :

$$y(k) = H(z) \quad (1.5)$$

Le système à temps discret qui en résulte est alors représenté par un modèle mathématique liant les valeurs numériques du signal d'entrée à celles du signal de sortie.

Cette relation peut être décrite par une équation récurrente qui joue le même rôle que les équations différentielles pour les systèmes continus. L'équation récurrente, qui se prête bien à la programmation, définit l'algorithme de génération de sa solution.

3.2.2 Description par la transformée en z

L'écriture et la résolution des équations récurrentes sont facilitées par l'emploi de l'opérateur z , d'avance d'une période d'échantillonnage T , défini à partir de la variable complexe p de Laplace par la relation :

$$z = e^{Tp} \quad (1.6)$$

Nous verrons qu'à toute équation récurrente linéaire, stationnaire, peut être associée une transformée en z , dont l'originale est solution de l'équation récurrente ; réciproquement, à toute transformée en z , on pourra associer une équation récurrente.

Le système à temps discret peut alors être modélisé par une fonction de transfert en z , qui lie les signaux d'entrée et de sortie :

$$Y(z) = H(z).U(z) \quad (1.7)$$

4. Echantillonnage et reconstitution du signal

4.1 Les éléments échantillonneurs

4.1.1 Echantillonneur idéal

Un échantillonneur idéal fait correspondre à un signal continu $x(t)$ une suite de valeurs discrètes, distribuées sur toutes les périodes (ou pas) T .

La fermeture du contact est supposée avoir lieu sur un laps de temps infiniment court par rapport à la période T .

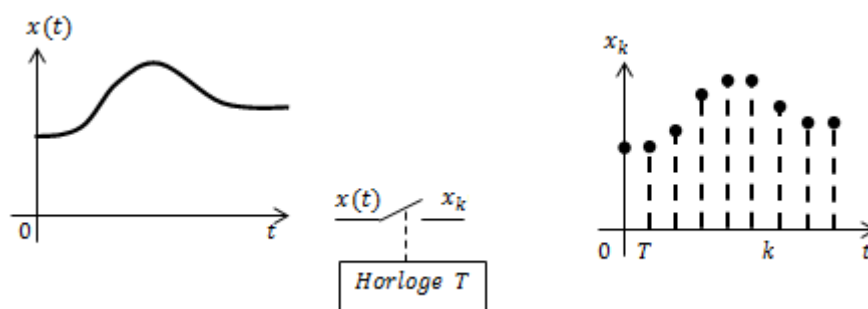


Figure (1.3) : Echantillonneur idéal

4.1.2 Echantillonneur à pulsations (ou échantillonneur)

En associant aux échantillons précédents des impulsions de Dirac (régulièrement réparties dans le temps, selon la période T).

On forme à partir du signal continu $x(t)$ le signal échantillonné x_k .

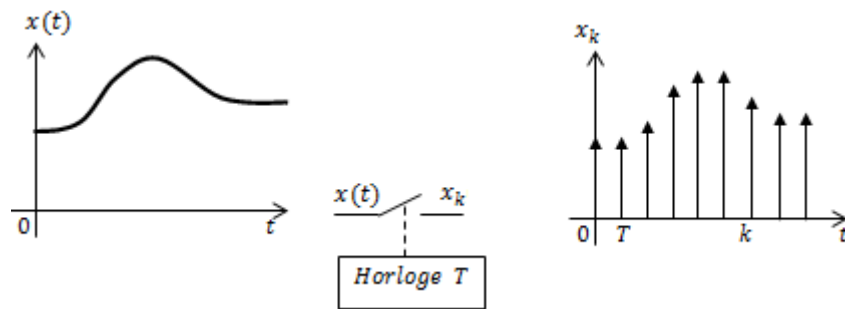


Figure (1.4) : Echantillonneur

On peut exprimer ce signal par :

$$x_k = \sum_{k=0}^{+\infty} x(k) \cdot \delta(t - k) \quad (1.8)$$

On peut remarquer que ce signal résulte en fait de la modulation d'amplitude d'un signal-porteur $\delta_T(t)$, formé d'un train périodique d'impulsions-unité, par le signal utile (continu) $x(t)$.

4.1.3 Quantification

Les calculateurs de processus travaillent de manière digitale. Un signal digital, convenablement codifié, ne peut pas varier de manière continue, mais seulement par gradins, à cause du nombre fini des chiffres représentant une grandeur digitale. Ainsi, après échantillonnage, le signal (toujours sous forme analogique) doit être quantifié, c'est-à-dire numérisé et transformé en différents niveaux qui sont codés en binaire. C'est le rôle du Convertisseur Analogique-Numérique (CAN) que de quantifier ce signal ; le CAN assure également et préalablement la fonction Echantillonnage.

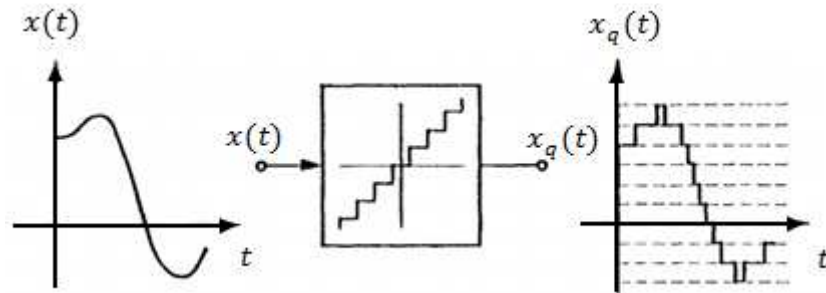


Figure (1.5) : Quantification

Si la quantification est effectuée dans de bonnes conditions, elle n'introduit aucune distorsion (ou une distorsion très faible) sur l'information contenue dans le signal, contrairement à la fonction-Echantillonnage, dont on verra qu'elle peut entraîner une perte d'information importante. Compte-tenu du principe de fonctionnement d'un CAN, le nombre de niveaux de conversion est toujours une puissance de 2 ; par exemple 2^8 , soit 256 niveaux. On dit alors que l'on dispose d'un convertisseur 8 bits. L'information analogique est alors codée sur un octet.

L'erreur absolue commise dans cette opération de quantification est proportionnelle aux nombres de niveaux de conversion; si l'on utilise un CAN 8 bits pour convertir une tension comprise entre 0 et 10V, l'erreur absolue maximum commise est égale à :

$$\frac{10}{2^8} = 0.04V \quad \text{Soit : } 40 \text{ mV environ}$$

4.2 Reconstitution d'un signal échantillonné

4.2.1 Théorème de Shannon

On remarque que l'on perd peu d'information sur un signal continu échantillonné à haute cadence Figure (1.6) (fréquence d'échantillonnage élevée par rapport au spectre de fréquences du signal analogique) ; par contre, si la fréquence d'échantillonnage est mal choisie, toute l'information peut être amoindrie Figure (1.7), voire irrémédiablement perdue Figure (1.8).

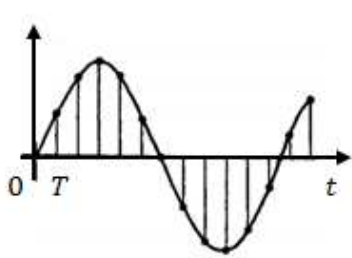


Figure (1.6)

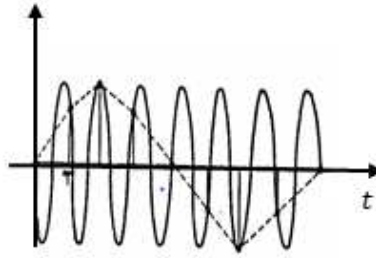


Figure (1.7)

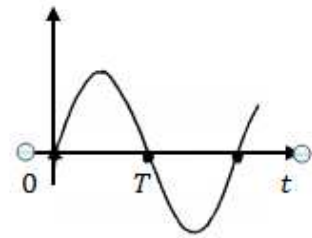


Figure (1.8)

Rappelons que le théorème de l'échantillonnage ou théorème de SHANNON précise les conditions dans lesquelles un signal analogique peut être reconstruit de façon unique à partir de sa version échantillonnée.

Théorème : Pour pouvoir reconstituer un signal continu à partir d'un train d'échantillons de période T , il faut que la pulsation d'échantillonnage $\Omega = \frac{2\pi}{T}$, soit au moins deux fois plus grande que la plus grande des pulsations contenues dans le signal continu que lui a donné naissance. [1]

4.2.2 Echantillonneur bloqueur

En supposant que le théorème de SHANNON ait été respecté, il n'en reste pas moins que l'échantillonnage amène une dégradation des propriétés dynamiques du signal et un certain retard dans la transmission des informations ; de même, la quantification amène une certaine dégradation de l'état du signal en régime permanent. Il est donc impératif de filtrer ce signal avant toute utilisation.

En Automatique, le filtre reconstituant le plus utilisé est le bloqueur d'ordre zéro. Cet élément de maintien est bien souvent associé à l'échantillonneur classique, car la plupart du temps :

- le signal discret (échantillon) est mis en mémoire jusqu'à l'apparition de l'échantillon suivant.
- le signal échantillonné (impulsion de Dirac) est transformé en échelon de position, entre deux prises d'informations, afin d'attaquer le système physique commandé.

On obtient le signal mémorisé $X_m(t)$ qui se présente comme une suite de paliers, changeant tous les instants kT .

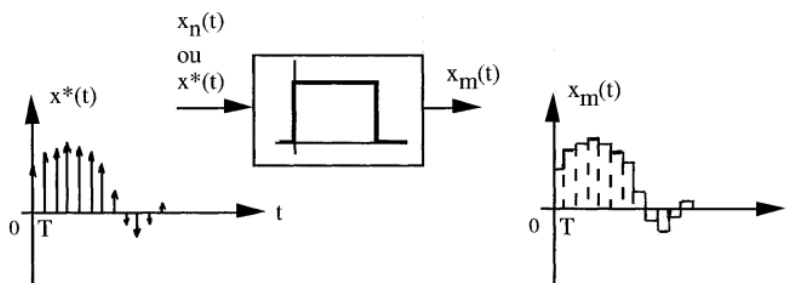


Figure (1.9) : échantillonneur bloqueur

On peut considérer que l'élément de maintien d'ordre zéro résulte de la différence entre deux échelons-unitaires de position, décalés de T :

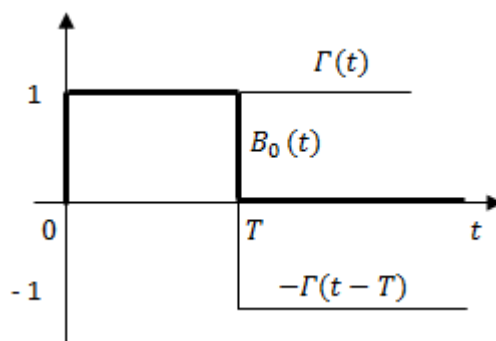


Figure (1.10) : Bloqueur d'ordre zéro a base d'échelon

$$B_0(t) = \Gamma(t) - \Gamma(t - T) \tag{1.9}$$

La forme générale d'un bloqueur d'ordre k est :

$$B_k(p) = kT \left(\frac{1 - e^{-Tp}}{Tp} \right)^2 + \frac{1 - e^{-Tp}}{p} (1 - ke^{-Tp}) \tag{1.10}$$

Remarque : on peut concevoir d'autres types de filtres-bloqueurs, tels que le bloqueur d'ordre k , avec correction partielle de vitesse :

Si $k = 0$ on retrouve le bloqueur d'ordre zéro

Si $k = 1$ on retrouve le bloqueur d'ordre un

5. Fonction de transfert discrète

5.1 Représentation symbolique d'un système à temps discret

Dans un système à temps discret, se trouvent rassemblés pour composer une chaîne de commande (en chaîne ouverte ou en boucle fermée) des éléments de type analogique (continu) et de type numérique (discret), associés à des convertisseurs CAN et CNA pour passer des uns aux autres. Cette chaîne peut être symbolisée par le schéma suivant :

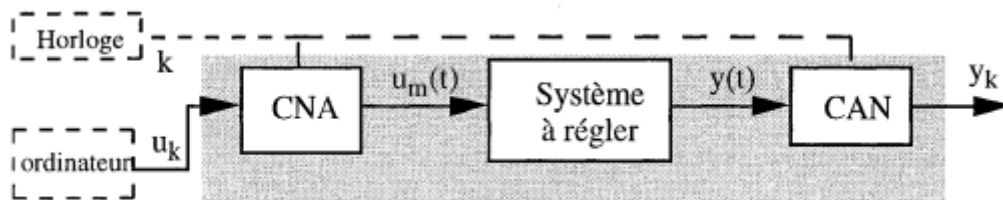


Figure (1.12) : Représentation d'un système à temps discret [2]

Puisqu'à une entrée discrète $u_k = u(k)$ correspond un signal discret $y_k = y(k)$ Bien que le processus physique à commander soit de type analogique, la vision qu'en a le calculateur (qui assure la commande) à travers les convertisseurs est de type discret ; en effet, les signaux analogiques $u(t)$ et $y(t)$ n'interviennent dans l'algorithme qu'aux instants d'échantillonnage délivrés par l'horloge de l'ordinateur. On peut donc définir une fonction de transfert échantillonnée, qui modélise le système et qui lie le signal discret d'entrée u_k au signal de sortie échantillonné y_k .

5.2 Notion de fonction de transfert échantillonnée

5.2.1 Convolution discrète

Comme la sortie d'un système linéaire continu résulte d'un produit de convolution :

$$y(t) = h(t) * u(t) \quad (1.11)$$

La relation entrée-sortie d'un système à temps discret s'appuie sur la notion de convolution discrète.

Soit le système linéaire à temps discret (ou système échantillonné) ci-dessous :

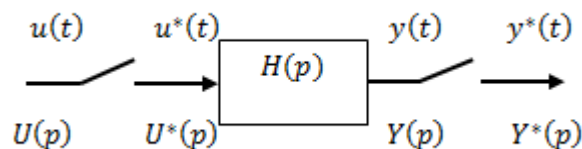


Figure (1.13) : Système échantillonné

On peut écrire :

$$y(t) = h(t) * u^*(t) \quad (1.12)$$

$$y(t) = h(t) * \sum_{k=0}^{+\infty} u(k) \delta(t - k) \quad (1.13)$$

$$y(t) = \sum_{k=0}^{+\infty} u(k) \cdot h(t) * \delta(t - k) \quad (1.14)$$

$$y(t) = \sum_{k=0}^{+\infty} u(k) \cdot h(t - k) \quad (1.15)$$

5.2.2 Fonction de transfert échantillonnée

On sait que [2]

$$Y^*(p) = \sum_{k=0}^{+\infty} y_k \cdot e^{-kp} \quad (1.16)$$

Alors

$$Y^*(p) = \sum_{k=0}^{+\infty} \left[\sum_{m=0}^k u(m) \cdot h[(k - m)] \right] \cdot e^{-kp} \quad (1.17)$$

Posons $k - m = n$ d'où $k = n + m$ Si $k = 0$, alors $n = -m$

$$Y^*(p) = \sum_{n=-m}^{+\infty} \sum_{m=0}^{+\infty} u(m) \cdot h(n) \cdot e^{-np} e^{-mp} \quad (1.18)$$

$$Y^*(p) = \sum_{n=-m}^{+\infty} h(n) e^{-np} \sum_{m=0}^{+\infty} u(m) \cdot e^{-mp} \quad (1.19)$$

Du fait de la causalité de la réponse impulsionnelle, on peut écrire :

$$Y^*(p) = \sum_{n=0}^{+\infty} h(n). e^{-np} . \sum_{m=0}^{+\infty} u(m). e^{-mp} \quad (1.20)$$

On voit que l'expression:

$$\sum_{n=0}^{+\infty} h(n). e^{-np} \quad (1.21)$$

est en fait **la fonction de transfert échantillonnée** $H^*(p)$ du système considéré. On constate que la sortie échantillonnée a alors pour transformée de Laplace :

$$Y^*(p) = H^*(p). U^*(p) \quad (1.22)$$

Il est évident que la relation entrée-sortie peut aussi s'exprimer par les transformées en z :

$$Y(z) = H(z). U(z) \quad (1.23)$$

6. Stabilité des systèmes à temps discret

6.1 Notion générale de la stabilité

6.1.1 Rappels

Tout réglage automatique, qu'il soit continu ou qu'il soit discret, doit absolument satisfaire à des conditions strictes de Stabilité

Il existe de nombreuses définitions de la Stabilité d'un système. On peut, par exemple, envisager le cas où le système étudié est excité pendant un certain temps par un signal d'entrée d'une allure quelconque bornée et ensuite soumis à une valeur nulle. On pourra conclure quant à la stabilité du système, selon la valeur que prend la grandeur de sortie en régime établi (pour $t \rightarrow \infty$). Trois cas pourront se présenter

- la sortie tend vers zéro : le système est stable (courbe a)
- la sortie tend vers une valeur finie : le système est à la limite de la stabilité (courbe b)
- la sortie diverge : le système est instable (courbe c)

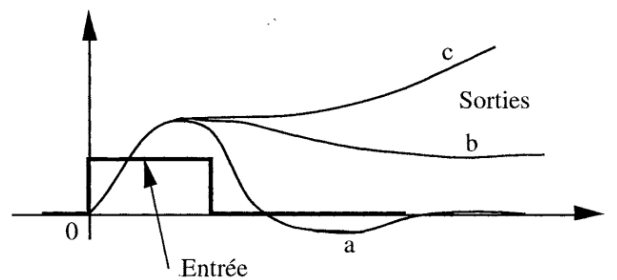


Figure (1.14) : Les trois cas de la stabilité

6.1.2 Cas des systèmes discret

Ces considérations générales sont transposables aux systèmes à temps discret.

La condition générale de stabilité s'écrit :

$$\lim_{k \rightarrow \infty} h(k) = 0 \quad (1.24)$$

Sachant que la réponse impulsionnelle $h^*(t)$ est une suite d'échantillons :

$$h^*(t) = \sum_{k=0}^{+\infty} h(k) \cdot \delta(t - k) \quad (1.25)$$

En supposant que le degré du dénominateur de la fonction de transfert du système soit d , le système possède d pôles α_i réels et/ou complexes conjugués, et le terme général de la réponse impulsionnelle peut se mettre sous la forme :

$$h(k) = \sum_{i=1}^d C_i \alpha_i^k \quad (1.26)$$

Cette expression tendra vers l'état d'équilibre-zéro si et seulement si tous les α_i ont leur module inférieur à l'unité :

$$|\alpha_i| < 1 \quad (1.27)$$

L'étude de la Stabilité d'un système à temps discret, qu'il soit en chaîne ouverte ou de type asservi, peut se faire soit par :

- l'approche fréquentielle dans le plan de NYQUIST comme dans le cas des systèmes continus où la fonction de transfert en boucle ouverte du système ne possède pas de pôles à partie réelle positive, le critère de NYQUIST s'énonce de la façon suivante :

Critère de NYQUIST : *Un système asservi est stable si la variation de l'argument de sa fonction de transfert en boucle ouverte, autour du point critique d'abscisse : -1, lorsque w varie de 0 à Ω , est nulle. Il est instable dans le cas contraire.*

- soit en s'appuyant sur la transformée en z qui se fait plus volontiers, soit par le critère de JURY, soit dans le plan des z par le lieu des racines (lieu d'EVANS).

Critère de JURY

Il s'agit d'une forme simplifiée du critère de SCHUR-COHN, qui est valable pour les polynômes à coefficients réels. Le critère de JURY donne des relations sous forme d'inégalités se basant sur les coefficients $d_0, d_1, d_2, .. d_n$ de l'équation caractéristique en z :

$$d(z) = \Sigma(z) = d_0 + d_1 z + d_2 \dots d_n z^n \quad (1.28)$$

Si et seulement si toutes les inégalités sont vraies, les racines de l'équation caractéristique se trouvent à l'intérieur du cercle-unité ; le système considéré est alors stable. [3]

Lieu des racines ou lieu d'EVANS

Méthode graphique qui s'applique souvent aux systèmes continus, elle peut être aussi utilisée dans le plan des z pour tester les conditions de stabilité des systèmes asservis à temps discret.

6.1.3 Marge de stabilité

Que ce soit par le critère de JURY ou par le tracé du lieu d'EVANS, l'étude de la stabilité des systèmes bouclés conduit à constater que sous certaines conditions (souvent imposées au gain statique en boucle ouverte) le système est stable ou instable.

Cependant, *il n'est guère possible de tirer des conclusions sur la qualité de cette stabilité, c'est-à-dire sur l'amortissement des phénomènes transitoires.*

On sait, à propos des systèmes asservis linéaires continus, qu'il est en particulier indispensable que le lieu de transfert en boucle ouverte (tracé dans le plan de NYQUIST) passe à une certaine distance du point critique, distance exprimée par la marge de phase ou la marge de gain.

Ces notions s'appliquent sans aucune difficulté aux systèmes échantillonnés. Prendre une marge de stabilité revient à limiter encore plus la région autorisée du plan des z où l'on représente les racines de l'équation caractéristique du système étudié.

La figure ci-dessous montre l'incidence de ces marges de stabilité dans le dans le plan des z .

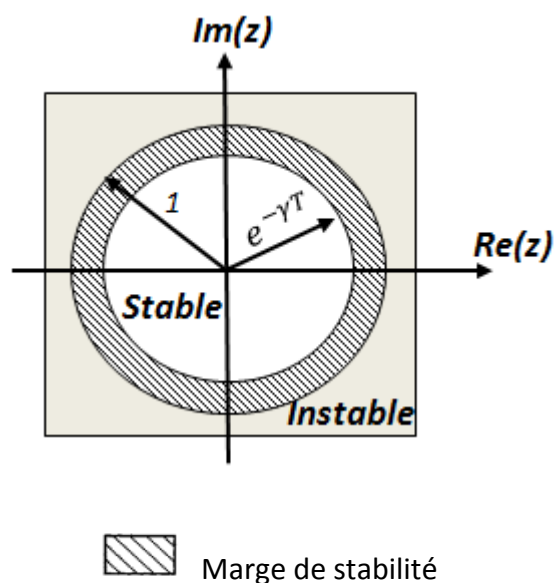


Figure (1.15) : Marge de stabilité dans le plan complexe

7 Précision et rapidité des systèmes discrets

7.1 Définition

La valeur permanente de l'écart qui existe entre la consigne, lorsqu'elle est unitaire, et la sortie d'un système discret, exprimée en pourcentage, constitue la précision du système

7.2 Précision statique

Nous nous proposons de donner dans ce paragraphe quelques notions sur la précision des systèmes à temps discret en régime établi aux instants d'échantillonnage ; ces notions ne sont que des extensions de celles concernant les systèmes continus.

7.2.1 Constante d'erreur

Soit un asservissement à temps discret, représenté en z

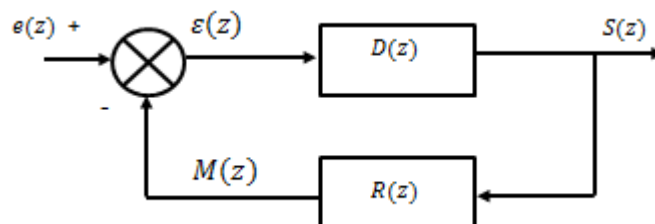


Figure (1.16) : Système discret en fonction de la variable z

$$h_{bo(z)} = D(z) \cdot R(z) \quad (1.29)$$

$$\varepsilon(z) = \frac{e(z)}{1 + H_{bo(z)}} \quad (1.30)$$

Sollicité par une commande, il présentera après disparition du régime transitoire une erreur statique permanente, obtenue à partir de :

$$\varepsilon(\infty) = \lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{k \rightarrow \infty} \varepsilon(k) = \lim_{z \rightarrow 1} (z - 1) \frac{e(z)}{1 + H_{bo(z)}} \quad (1.31)$$

L'erreur permanente dépend de la forme de l'entrée et évidemment de celle de la fonction de transfert du système échantillonné étudié. Pour calculer cette erreur, il est intéressant d'introduire la notion de constante d'erreur, liée à l'ordre de l'entrée appliquée.

7.3 La rapidité

7.3.1 Considération sur le régime transitoire

On peut se faire une bonne idée de la nature du phénomène transitoire dans un système échantillonné ou discret en observant la relation existant entre la réponse impulsionnelle et les pôles et les zéros de la fonction de transfert correspondante.

7.3.2 Rapidité des systèmes discrets

Pour tester la rapidité d'un système échantillonné, on procède tout à fait comme pour les systèmes continus :

- signal-test : échelon-unité ou échelon de position

$$\Gamma^*(t) = \sum_{n=0}^{+\infty} \Gamma(n) = \delta_T(t) \quad (1.32)$$

- temps de réponse à $\pm 5\%$ de la réponse permanente,
- la fonction de transfert à prendre en considération est $H(z)$ si on a affaire à un système isolé ou en chaîne ouverte, $H_{bf}(z)$ si on étudie le cas d'un système bouclé.

7.4 Systèmes contraints

7.4.1 Dilemme précision rapidité stabilité

La précision dynamique d'un système en réponse à une entrée donnée, ainsi que sa rapidité sont fortement liées, puisque l'une et l'autre de ces caractéristiques dépendent de l'évolution du régime transitoire qui affecte le système. Du fait de la souplesse d'emploi des machines numériques et des possibilités qu'elles offrent pour engendrer certaines fonctions uniquement par programmation, les systèmes à temps discret se prêtent mieux à transformation ; on peut donc imposer certaines contraintes au système en boucle fermée,

permettant de biaiser le dilemme précision-rapidité, classique dans le réglage des systèmes asservis continus.

On trouvera dans la suite de ce paragraphe, deux types de systèmes contraints, qui sont intéressants du fait qu'ils présentent une rapidité contrôlée et une erreur statique nulle.

7.4.2 Système minimal

Un système est dit minimal ou à temps d'établissement fini lorsque, pour une entrée spécifiée, le régime définitif est atteint en un nombre fini des périodes ; l'erreur correspondant à cette entrée s'annule.

On peut noter que cette appellation correspond à une extension de la notion de temps de réponse, qui en toute rigueur correspond à la seule entrée indicielle $F(kT)$. Si de plus, le système est soumis à des contraintes qui minimisent son régime transitoire, il est dit à temps de réponse fini minimal absolu, ou système minimal absolu. Cette définition autorise des dépassements (oscillations) de la réponse obtenue, durant le régime transitoire.

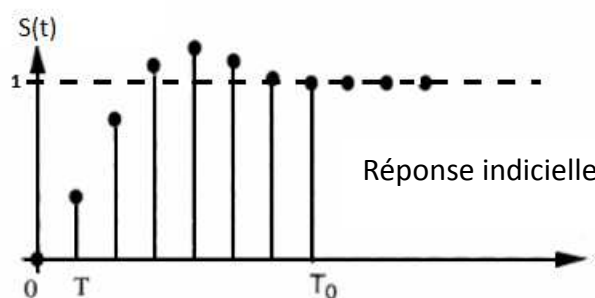


Figure (1.17) : Représentation d'une réponse pour un système minimal

7.4.3 Système à réponse pile

Le réglage de ce système rajoute une spécification supplémentaire, puisqu'un tel système est dit à réponse-pile lorsque la sortie atteint son régime définitif, pour une entrée-type donnée, en un nombre fini d'échantillons, sans dépassement.

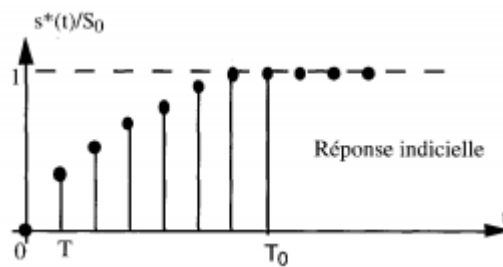


Figure (1.18) : Représentation d'une réponse pile pour une entrée échelon

Après j périodes, sans dépassement, le système a atteint définitivement son régime permanent.

Là aussi, l'entrée appliquée n'est pas forcément un échelon-unité. La figure suivante montre un système à réponse-pile pour une commande en rampe de vitesse.

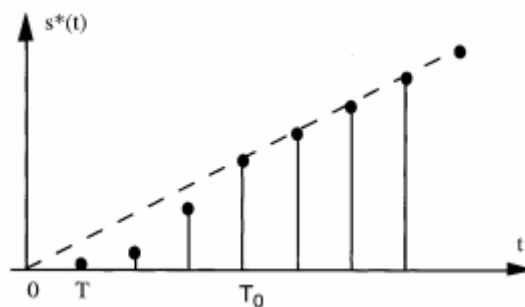


Figure (1.19) : Représentation d'une réponse pile pour une entrée rampe

Pour être à réponse-pile, le système doit satisfaire à un certain nombre de conditions pas toujours réalisables, ce qui limite considérablement la portée de cette approche. C'est cependant un point de vue qui peut être intéressant lors de la détermination des circuits correcteurs, qui permettent de compenser les systèmes asservis échantillonnés.

8 Conclusion

Dans ce premier chapitre nous avons fait un rappel général sur les systèmes discrets tel que : l'échantillonnage, la modélisation et les performances des systèmes discrets ou on a pu constater la simplicité et l'efficacité du discret et on a pu conclure que ce dernier est plus avantageux que le continu et c'est la raison de notre choix (correcteur numérique).

Le tableau suivant montre la comparaison entre la correction analogique et la correction numérique.

	Numérique	Analogique
Avantages	<ul style="list-style-type: none"> - capacité de mémoire sans limite, - grande possibilité de synthèse de la récurrence (algorithme), - algorithme adaptable, - mise au point souple, - convivialité (dialogue) et sécurité (séquentiel). 	<ul style="list-style-type: none"> - le système est observé en continu (adapté aux systèmes rapides), - précis.
Inconvénients	<ul style="list-style-type: none"> - la boucle est ouverte en dehors des instants d'échantillonnage (oscillations non observées), - discrétisation de la mesure (précis à $q = \frac{PE}{2^N}$), non linéaire : risque de cycle limite, - compromis $\frac{fe}{N}$ (nombre de bits), pas encore adapté aux systèmes rapides (Shannon). 	<ul style="list-style-type: none"> - structure rigide ; PID(ou approchée), - possibilité de synthèse réduite, - mise au point et adaptation difficile, - d'un point de vue technologique (valeurs des composants), non adapté aux systèmes lents.

Tableau (1.1) : comparaison entre l'analogique et le numérique

CHAPITRE 2

1. Introduction

L'analyse d'un système asservi, qu'il soit continu ou discret, conduit inmanquablement à se poser le problème de l'adjonction au système d'un circuit correcteur, afin de conférer à l'ensemble les meilleures performances possibles.

Ainsi, étant donnée une installation de fonction de transfert connue, réaliser sa synthèse consiste à rechercher un réseau correcteur d'expression $C(z)$, conçu de telle façon que l'ensemble corrigé satisfasse un certain nombre de spécifications qui peuvent se traduire soit dans le domaine temporel, soit dans le domaine fréquentiel aux :

- **exigences temporelles** (réponse temporelle imposée, temps de réponse minimal, dépassement maximal contrôlé, précision statique imposée,...etc)
- **exigences fréquentielles** (marge de phase imposée, bande passante à respecter, résonance spécifiée,...etc.)

L'utilisation des machines numériques (microprocesseur et autres, plus ou moins importantes) pour commander les systèmes automatiques facilite grandement cette façon de procéder. En effet, il est alors relativement aisé de programmer également des éléments correcteurs qui permettent d'obtenir des effets très intéressants sur les signaux de commande du processus, difficilement réalisables par éléments câblés.

2. Principe générale de la régulation

Dans la plupart des appareils et installations industrielles, tertiaires et mêmes domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations externes ou internes influant sur ces grandeurs. Le niveau d'un réservoir d'eau, la température d'une étuve, le débit d'une conduite de gaz, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré. Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage (dit en boucle ouverte) permet d'obtenir et de maintenir la valeur demandée (par exemple : action sur un robinet d'eau). Dans la majorité des cas, ce type de réglage n'est pas suffisant pour garantir la stabilité des systèmes. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglante. On a, dans

ce cas, constitué une boucle de régulation et plus généralement une boucle d'asservissement (Figure 2.1). Cette boucle nécessite la mise en œuvre d'un ensemble de moyens de mesure, de traitement de signal ou de calcul, d'amplification et de commande d'actionneur, constituant une chaîne d'éléments associés : la chaîne de régulation (ou d'asservissement). Toute chaîne de régulation (ou d'asservissement) comprend trois maillons indispensables : l'organe de mesure, l'organe de régulation et l'organe de contrôle. Il faut donc commencer par mesurer les principales grandeurs servant à contrôler le processus. L'organe de régulation récupère ces mesures et les compare aux valeurs souhaitées, plus communément appelées valeurs de consigne. En cas de non-concordance des valeurs de mesure et des valeurs de consigne, l'organe de régulation envoie un signal de commande à l'organe de contrôle (vanne, moteur, etc.), afin que celui-ci agisse sur le processus. Les paramètres qui régissent le processus sont ainsi stabilisés en permanence à des niveaux souhaités.

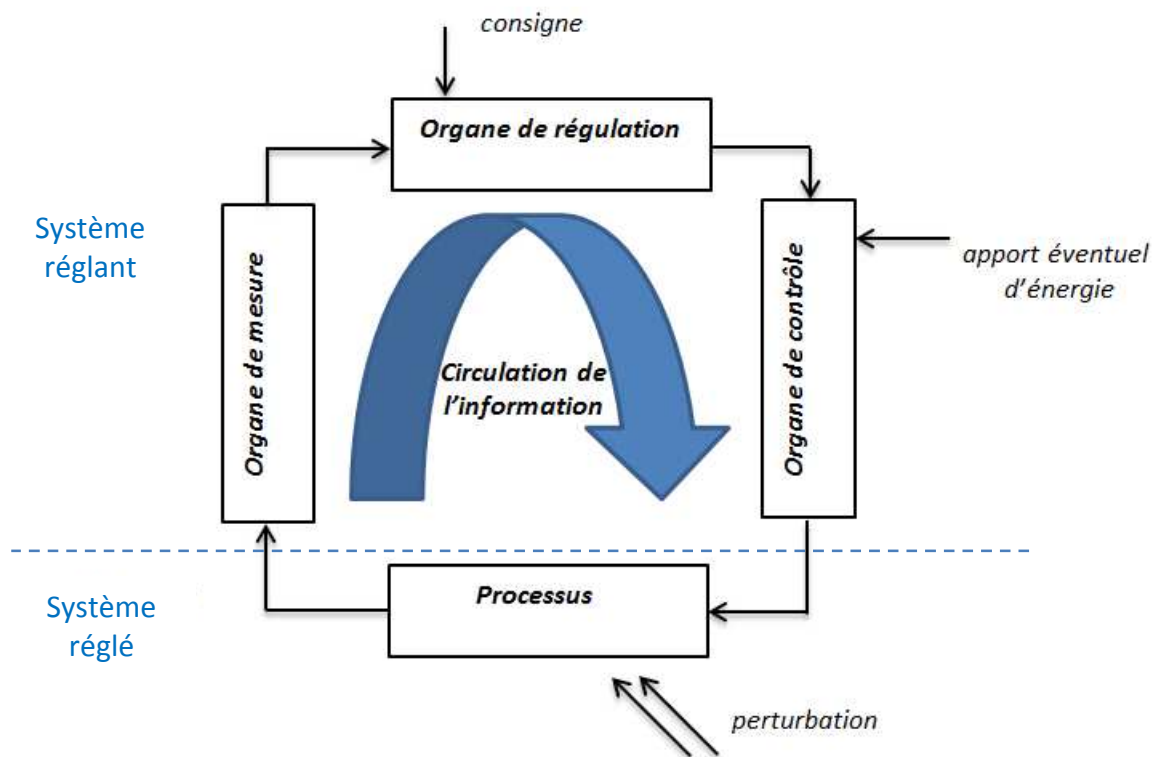


Figure (2.1) : Schéma de principe d'une chaîne de régulation

Le choix des éléments de la chaîne de régulation est dicté par les caractéristiques du processus à contrôler, ce qui nécessite de bien connaître le processus en question et son comportement.

2.1 Comportement en régulation

La consigne est maintenue constante et il se produit sur le procédé une modification (ou une variation) d'une des entrées perturbatrices. L'aspect régulation est considéré comme le plus important dans le milieu industriel, car les valeurs des consignes sont souvent fixes. Néanmoins, pour tester les performances et la qualité d'une boucle de régulation, on s'intéresse à l'aspect asservissement.

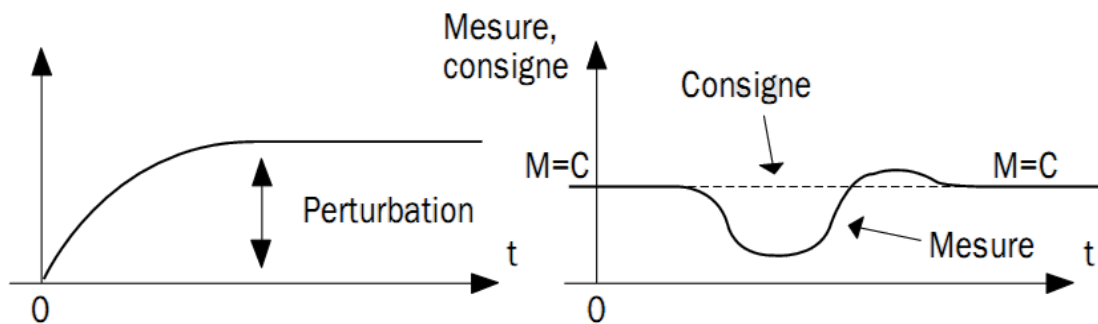


Figure (2.2) : comportement en régulation

2.2 Comportement en asservissement

L'objectif de tout asservissement est de maintenir la mesure M la plus proche possible de la consigne C.

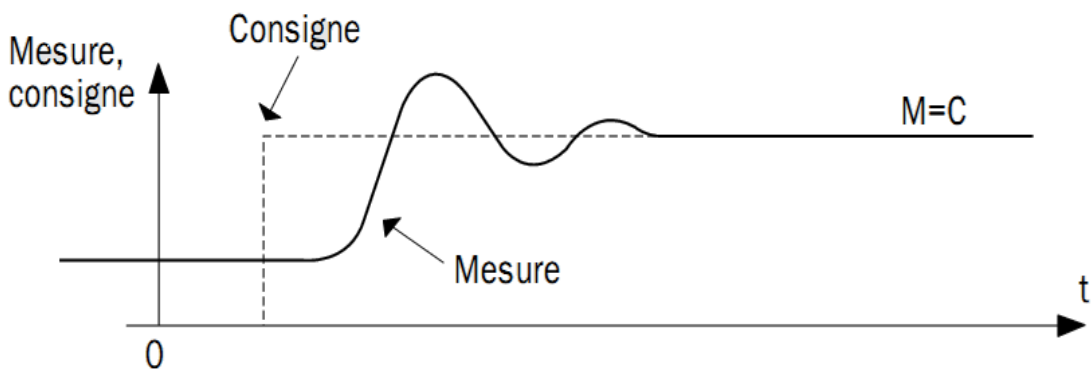


Figure (2.3) : comportement en asservissement

3. Numérisation des correcteurs analogique [3]

Il existe grosso modo six filières de synthèse d'un régulateur numérique, comme cela est schématisé dans la figure suivante.

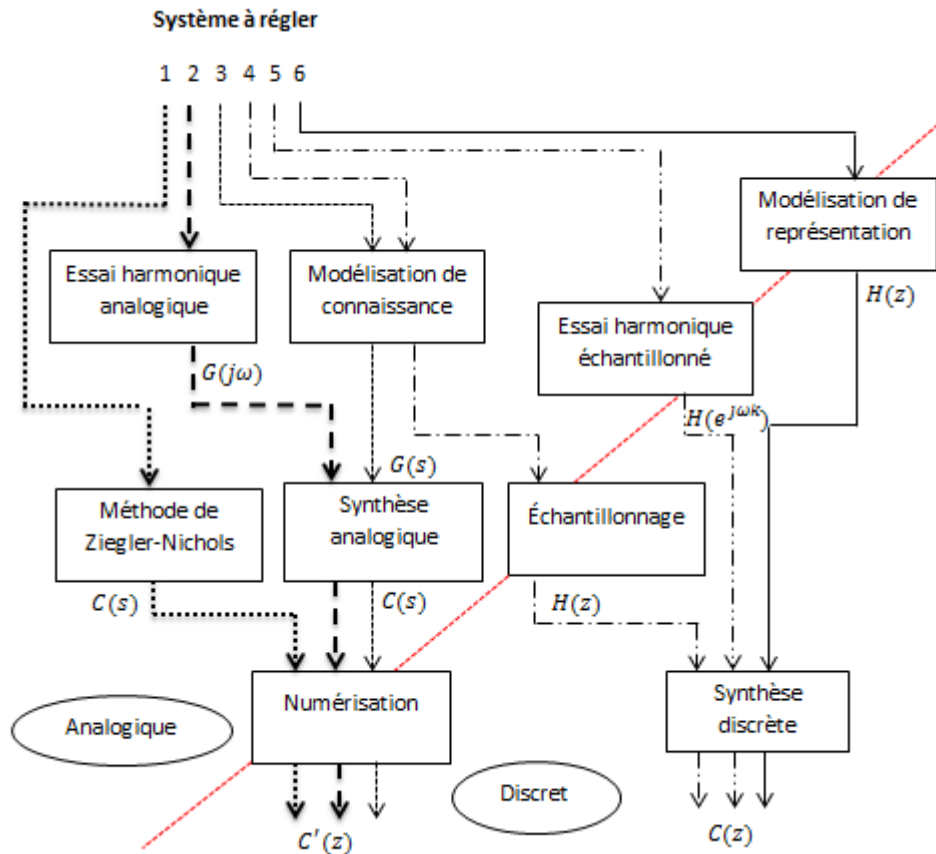


Figure (2.4) : Synthèse d'un régulateur numérique

La première approche consiste à exploiter la méthode de Ziegler-Nichols, se fondant sur un essai expérimental en boucle ouverte ou fermée, fournissant le régulateur analogique $C(s)$. Ce dernier est traduit, le plus fidèlement possible, sous la forme d'un algorithme représenté par la fonction de transfert discrète $C'(z)$. La phase de traduction est rencontrée sous le nom de **numérisation**.

La deuxième filière repose sur un essai harmonique effectué sur le système à régler analogique, conduisant à la fonction de transfert harmonique $G(j\omega)$. Une synthèse intégralement analogique, n'incorporant pas les convertisseurs analogique-digital et digital-analogique, donne le régulateur analogique $C(s)$, lequel est finalement numérisé.

La troisième approche tire profit des lois physiques gouvernant le processus à régler, telles que les équations de la mécanique, de l'électricité, de la thermodynamique...etc. Afin de construire un **modèle de connaissance**. Ces lois étant différentielles, le modèle obtenu, si

nécessaire après linéarisation autour d'un point de fonctionnement et introduction de variable d'écart, est une fonction de transfert analogique $G(s)$. Une synthèse analogique conduit alors au régulateur $C(s)$, qui est ensuite numérisé.

La quatrième méthode se fonde toujours sur le **modèle de connaissance** $G(s)$, lequel est échantillonné. La fonction de transfert discrète $H(z)$ qui en résulte rend possible une synthèse tenant compte de tous les phénomènes discrets mis en jeu.

La cinquième filière consiste en un essai harmonique discret sur le système à régler équipé des convertisseurs digital-analogique et analogique-discret. Cette phase fournit la fonction de transfert harmonique discrète $H(e^{j\omega k})$, qui autorise une synthèse discrète.

La sixième et dernière approche, quant à elle, ne se base pas explicitement sur les lois physiques auxquelles obéit le processus à régler, mais sur des résultats d'essais réels effectués *in situ* sur ce dernier. Une procédure d'identification élabore un modèle de représentation se comportant de manière aussi proche que possible de l'installation à régler. Le résultat est directement une fonction de transfert discrète $H(z)$ régissant le système à régler échantillonné ; c'est le point de départ d'une synthèse discrète d'un régulateur numérique.

On peut affirmer que dans les trois premières méthodes, passant par une numérisation, la frontière séparant le monde analogique du monde discret est repoussée le plus tard possible. Dans la quatrième technique, l'échantillonnage du processus à régler permet de franchir cette frontière immédiatement après la modélisation de connaissance. Enfin, les cinquième et sixième techniques font intégralement partie du domaine numérique.

4. **Faisabilité des correcteurs** [2]

Afin de déterminer le réseau correcteur qui convient le mieux à un système asservi commandé numériquement, on ramènera, dans la suite, tout système à un diagramme fonctionnel à retour unitaire :

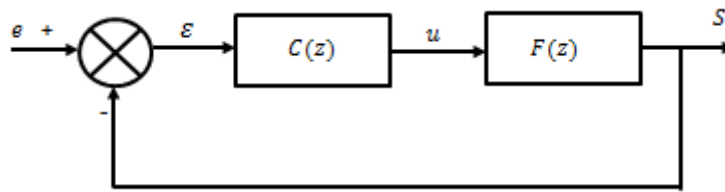


Figure (2.5) : diagramme fonctionnel à retour unitaire pour système à corriger

Dans ce cas, la fonction de transfert en boucle fermée s'exprime par :

$$H_{bf}(z) = \frac{C(z).F(z)}{1+C(z).F(z)} \quad (2.1)$$

Le cahier des charges imposant certaines contraintes (de rapidité, de précision,...) au système bouclé, on connaît a priori la forme à donner à $H_{bf}(z)$. Le processus de fonction de transfert $F(z)$ étant lui-même connu, on en déduit que le correcteur doit répondre à :

$$C(z) = \frac{H_{bf}(z)}{F(z)[1-H_{bf}(z)]} \quad (2.2)$$

Généralement, le correcteur se présentera sous la forme d'un rapport de polynômes en z :

$$C(z) = \frac{b_0+b_1z+b_2z^2+\dots+b_nz^n}{a_0+a_1z+a_2z^2+\dots+a_dz^d} \quad (2.3)$$

La réalisation pratique du correcteur, déterminé par le calcul, implique que le développement en série de sa fonction de transfert $C(z)$ ne comporte pas de puissances positives de z (principe de causalité à respecter impérativement). Cette condition se traduit par :

$$\lim_{z \rightarrow \infty} [z^{-1}C(z)] = 0 \quad (2.4)$$

C'est-à-dire que l'on doit avoir : $n \leq d$

Cette condition est nécessaire, mais il y a des cas où elle n'est pas suffisante ; On voit par exemple qu'à l'instant k le signal de sortie du correcteur doit être synchrone de sa commande. Or les conversions AN et NA, d'une part, et le calcul de l'expression de

correcteur, d'autre part, ne sont pas instantanés ; il faut un certain temps entre la prise d'échantillon ε et la commande du processus. Si ce temps est faible vis-à-vis de la période d'échantillonnage, il n'y a pas de problème. Mais si l'échantillonnage est très rapide, ou si les calculs sont complexes et nombreux, on ne pourra pas réaliser la simultanéité de l'acquisition et de la commande. Dans ce cas, on aura tout intérêt à prévoir une condition plus restrictive pour la faisabilité du correcteur :

$$\lim_{z \rightarrow \infty} C(z) = 0 \quad (2.5)$$

C'est-à-dire : $n < d$

5. Critère du choix de correcteur

5.1 Condition générales et critère temporels

La détermination des organes de compensation d'un asservissement échantillonné s'appuie sur un certain nombre de démarches, plus ou moins efficaces, qui font appel à des critères spécifiques.

Une méthode classique, dite des pôles dominants de ZDAN consiste à imposer à un système asservi de présenter une fonction de transfert en boucle fermée dont le comportement soit voisin de celui d'un système du deuxième ordre, c'est-à-dire caractérisé essentiellement par une paire de pôles dominants. Cette méthode, basée sur la transformation en z , peut amener des résultats intéressants.

On peut aussi s'appuyer sur des considérations de précision statique et/ou de rapidité. Il s'agit pour l'essentiel de critères temporels, qui imposent au système de répondre à une entrée donnée, selon certaines spécifications sur ses régimes transitoire et permanent. En fait, on définit un signal de sortie qui doit répondre à certaines spécifications: progression d'un échantillon à l'autre, dépassement contrôlé, loi d'évolution en régime permanent, erreur permanente admise,... Les méthodes de calcul des réseaux correcteurs qui résultent de cette approche sont donc caractérisées par l'utilisation directe des spécifications sur la réponse du système à des entrées déterminées.

Il est bien entendu que, quel que soit le critère utilisé, il faudra en dernier lieu vérifier la faisabilité de la solution préconisée.

5.2 Méthode de calcul

Soit la structure fonctionnelle retenue pour représenter le système à corriger figure (2.5) :

On peut exprimer le signal d'erreur :

$$\varepsilon(z) = e(z) - s(z) \quad (2.6)$$

Soit :
$$\varepsilon(z) = e(z)[1 - H_{bf}(z)] \quad (2.7)$$

On a vu que la transformée en z d'une entrée canonique d'ordre m peut se mettre sous la forme :

$$e(z) = \frac{A(z)}{(1-z^{-1})^{m+1}} \quad (2.8)$$

Où $A(z)$ est un polynôme en z^{-1} , de degré au plus égal à m et dépourvu de racines égales à l'unité :

$$\varepsilon(z) = \frac{A(z)}{(1-z^{-1})^{m+1}} [1 - H_{bf}(z)] \quad (2.9)$$

On peut définir un système en évaluant sa fonction de transfert en boucle fermée de telle façon que, sollicité par une entrée donnée, il présente certaines conditions de précision en régime permanent :

$$\lim_{k \rightarrow \infty} \varepsilon(k) = \lim_{z \rightarrow 1} (1 - z^{-1})\varepsilon(z) \quad (2.10)$$

$$\lim_{k \rightarrow \infty} \varepsilon(k) = \lim_{z \rightarrow 1} [A(z)(1 - z^{-1})^{-m} [1 - H_{bf}(z)]] \quad (2.11)$$

Ainsi, pour une entrée (d'ordre m fixé), on peut vouloir que :

$$\lim_{k \rightarrow \infty} \varepsilon(k) = 0 \quad \text{Précision parfaite} \quad (2.12)$$

Ou
$$\lim_{k \rightarrow \infty} \varepsilon(k) = Cste \quad \text{Précision relative} \quad (2.13)$$

Partant de ces considérations, la relation précédente permet de calculer $H_{bf}(z)$, puis d'en déduire la fonction de transfert $C(z)$ du correcteur qui, à partir du système initial $F(z)$, conduit à obtenir les performances souhaitées ; bien évidemment, il conviendra de vérifier la faisabilité de la solution préconisée.

Par exemple : Si l'on désire parvenir à un système présentant une précision parfaite pour une entrée d'ordre m (la précision ne peut être définie que par rapport à l'ordre de la commande), il faudra que :

$$1 - H_{bf}(z) = (1 - z^{-1})^{m+1} \cdot B(z) \quad (2.14)$$

$$\text{Alors} \quad \lim_{k \rightarrow \infty} \varepsilon(k) = \lim_{z \rightarrow 1} (1 - z^{-1})A(z)B(z) = 0 \quad (2.15)$$

Et ceci quel que soit le polynôme $B(z)$, pourvu que celui-ci ne présente pas de racines égales à l'unité.

On déduit alors :

$$H_{bf}(z) = 1 - (1 - z^{-1})^{m+1}B(z) \quad (2.16)$$

$$\text{Et} \quad \varepsilon(z) = A(z)B(z) \quad (2.17)$$

Le correcteur qui permet d'atteindre ce résultat, *s'il remplit les conditions de faisabilité*, aura pour expression :

$$C(z) = \frac{1 - [1 - z^{-1}]^{m+1} \cdot B(z)}{(1 - z^{-1})^{m+1} \cdot F(z) \cdot B(z)} \quad (2.18)$$

En fait, le calcul montre que le signal d'erreur $\varepsilon^*(t)$ s'annule en un nombre fini d'échantillons ($\varepsilon(z)$ est un polynôme en z^{-1}). Les systèmes répondant à ces spécifications sont donc à temps d'établissement fini, pour l'entrée considérée. On constate ainsi que les considérations sur la précision entraînent des conséquences sur la rapidité du système, et réciproquement.

6. Correcteurs en dérivation

Dans tout calcul de systèmes de commande, l'une des préoccupations essentielles du concepteur est d'assurer la fiabilité de l'ensemble, c'est-à-dire de veiller à ce que la défaillance d'une partie de la chaîne n'entraîne pas la détérioration de tout l'ensemble.

On se souvient que les régulateurs P.I.D. universels continus sont prévus pour autoriser une conduite manuelle du processus en cas de panne de tout ou partie du régulateur. Dans le cas de la commande de systèmes échantillonnés, on suppose que l'on peut être confronté à des incidents similaires dus au calculateur numérique qui pilote le processus et qui assure en particulier la fonction-régulateur.

Une structure intermédiaire consiste à faire assurer la sûreté de fonctionnement par un équipement continu (analogique) et à provoquer des performances plus élaborées par une commande numérique du système.

Le diagramme fonctionnel ci-dessous illustre ce type de conception hybride.

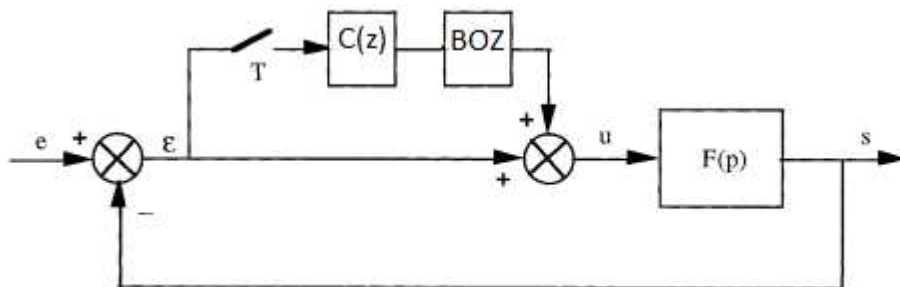


Figure (2.6) : structure d'un correcteur hybride [2]

Le correcteur numérique est en dérivation, ou en parallèle, avec une connexion continue entre le signal d'erreur et l'installation. Ainsi le signal de commande de l'installation résulte de la somme de signaux continus et numériques :

$$U(p) = \varepsilon(p) + C^*(p) \cdot \varepsilon^*(p) \quad (2.19)$$

Si le correcteur numérique est temporairement défaillant, le système asservi analogique continue à opérer ; l'installation fonctionne alors sous mode dégradé.

7. Les régulateurs discrets standards

7.1 Généralités

Les trois qualités d'un asservissement sont: la précision, la vélocité et l'amortissement. Les lieux de Bode de $T(j\omega)$ en boucle ouverte (*Figure 2.7*) permettent de bien comprendre le but de la correction par différents régulateurs :

- Pour améliorer la précision il faut, pour une classe donnée, remonter la courbe de gain dans le domaine des basses pulsations (augmenter le gain en position ou en vitesse).
- Pour améliorer la vélocité il faut augmenter la pulsation ω_{-6BO} donc tirer la courbe de gain vers la droite.

Nous voyons bien qu'il n'y a aucune incompatibilité entre ces deux premières qualités car si on relève l'ensemble de la courbe de gain on améliore l'ensemble la précision et la vélocité. Malheureusement il y a l'amortissement de la boucle qui est une qualité essentielle.

Pour avoir un bon amortissement il faut une bonne marge de phase $M\phi$. Et là nous voyons bien que c'est incompatible avec une translation vers le haut de la courbe de gain en boucle ouverte. Il y a donc toujours dans une boucle d'asservissement un dilemme précision-vélocité et la qualité amortissement (qualité indispensable).

On peut donc en déduire que le but de la correction est de remonter la courbe de gain (en boucle ouverte) dans le domaine des basses pulsations et de tirer cette courbe de gain vers la droite dans le domaine des hautes pulsations, tout en conservant des marges de stabilité satisfaisantes (c'est là qu'intervient la courbe de phase!).

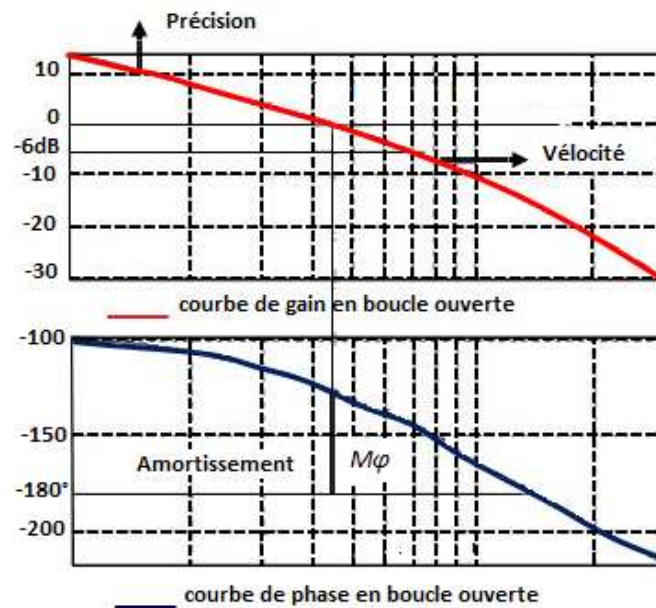


Figure (2.7) : courbe de gain et de phase en boucle ouverte

7.2 Les différentes actions

En pratique, à une catégorie donnée de systèmes à asservir correspond un type de correcteur adapté. Pour effectuer un choix judicieux, il faut connaître les effets des différentes actions : proportionnelle, intégrale et dérivée.

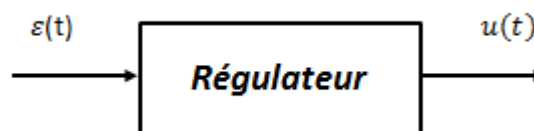


Figure (2.8) : Schéma synoptique d'un régulateur PID

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

- Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne
- Il élimine l'erreur statique grâce au terme intégrateur
- Il anticipe les variations de la sortie grâce au terme dérivateur.

La commande $u(t)$ donnée par le régulateur PID, dans sa forme classique est décrite par :

$$u(t) = K_p \cdot \varepsilon(t) + \frac{1}{T_i} \cdot \int_0^t \varepsilon(\tau) d\tau + T_d \cdot \frac{d\varepsilon(t)}{dt} \quad (2.20)$$

7.2.1 Action proportionnelle

La sortie $u(k)$ du régulateur proportionnel est donnée en fonction de son entrée $\varepsilon(k)$ qui représente l'écart entre la consigne et la mesure par la relation :

Équation discrète :

$$u^p(k) = K_p \cdot \varepsilon(k) \quad (2.21)$$

Fonction de transfert :

$$C_p(z) = \frac{u^p(z)}{\varepsilon(z)} = K_p \quad (2.22)$$

Diagramme structurel :

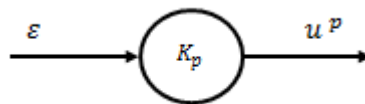


Figure (2.9) : Diagramme structurel de l'action proportionnelle

7.2.2 Action intégrale I

Le régulateur I intègre l'écart de réglage en fonction du temps. Cependant, dans le domaine des régulateurs discrets, l'intégration est remplacée par une sommation de l'écart de réglage discret $\varepsilon^*(t)$. En toute rigueur, on devrait donc parler de régulateur-sommeur...

Équation discrète :

$$u^i(k) = u^i(k-1) + K_i \cdot \varepsilon(k) \quad \text{avec} \quad K_i = \frac{1}{T_i} \quad (2.23)$$

Fonction de transfert :

$$C_i(z) = \frac{u^i(z)}{\varepsilon(z)} = K_i \frac{z}{z-1} = K_i \frac{1}{1-z^{-1}} \quad (2.24)$$

Diagramme structurel :

A l'aide de l'équation récurrente (forme récursive) et en observant que le bloc z^{-1} introduit le décalage de u^i en u^i_{-1} , on peut représenter la relation qui lie u^i à ε par le schéma ci-après, appelé diagramme structurel qui met bien en évidence la mise en mémoire de la valeur de la grandeur u afin de l'utiliser à la période suivante :

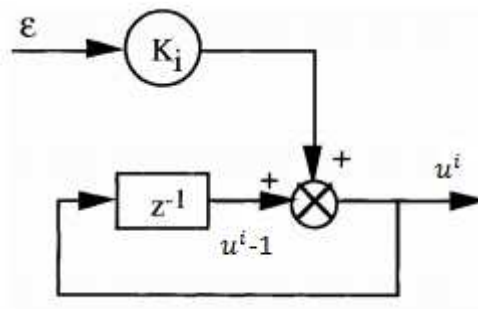


Figure (2.10) : Diagramme structurel de l'action intégral

7.2.3 action dérivée D

L'action dérivée se traduit par un terme proportionnel à la différence des écarts de réglage aux instants d'échantillonnage k et $(k - 1)$.

Equation discrète :

$$u^d(k) = K_d[\varepsilon(k) - \varepsilon(k - 1)] \text{ avec } K_d = T_d \quad (2.25)$$

Fonction de transfert :

$$C_d(z) = \frac{y^d(z)}{\varepsilon(z)} = K_d \frac{z-1}{z} = K_d(1 - z^{-1}) \quad (2.26)$$

Diagramme structurel :

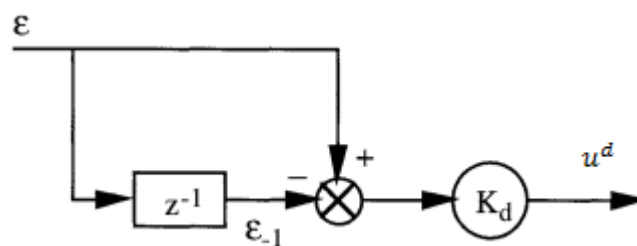


Figure (2.11) : Diagramme structurel de l'action dérivée

7.3 Les régulateurs standards

7.3.1 Le régulateur proportionnel-intégral PI

Le régulateur P.I est une combinaison d'un régulateur P et d'un régulateur I. il élabore alors une commande qui peut être donnée par :

Equation discrète :

$$u(k) = K_p \cdot \varepsilon(k) + K_i \sum_{j=0}^k \varepsilon(j) \quad (2.27)$$

Equation récurrente :

$$u(k) = u^i(k-1) + (K_p + K_i) \cdot \varepsilon(j) \quad (2.28)$$

Fonction de transfert :

$$C(z) = K_p + K_i \frac{z}{z-1} = \frac{b_1 z + b_0}{z-1} = \frac{b_1 + b_0 z^{-1}}{1-z^{-1}} \quad (2.29)$$

Avec : $b_1 = K_p + K_i$ et $b_0 = -K_p$

Le régulateur P.I est un système discret du premier ordre, ayant un pôle égal à 1. Le régulateur P.I peut donc annuler l'écart de réglage, en régime établi, dû à une sollicitation en échelon de position.

Diagramme structurel :

Le diagramme structurel du régulateur P.I se déduit aisément de son équation discrète :

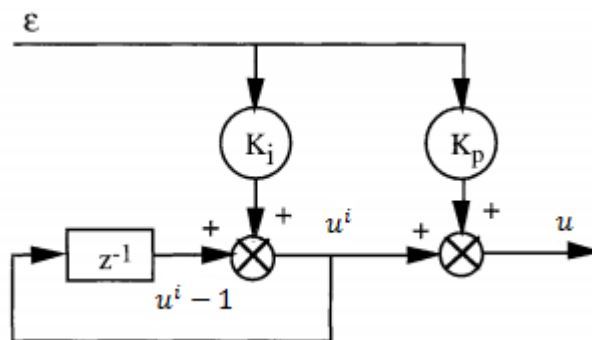


Figure (2.12) : Diagramme structurel d'un régulateur PI

7.3.2 Correcteur à retard de phase [5]

La forme approchée du correcteur PI est le correcteur à retard de phase.

$$C(p) = K \frac{1 + \tau_1 p}{1 + \tau_2 p} ; \quad \text{Avec } \tau_2 > \tau_1 \quad (2.30)$$

Ce n'est qu'une forme approchée car cette fois-ci on ne rajoute pas de pôle à l'origine, son objectif est identique à celui du correcteur PI, c'est-à-dire améliorer le gain statique (en laissant la précision dynamique inchangée). Ainsi, nous le présenterons directement sous la forme :

$$C(p) = Kp \frac{1+\tau p}{1+\beta \tau p} \quad ; \text{ Avec } \beta > 1 \quad (2.31)$$

On applique la règle des trapèzes :

$$p = \frac{2}{Te} \frac{(z-1)}{(z+1)} \quad (2.32)$$

à l'équation (2.31) on obtient la fonction de transfert discrète en z suivante :

$$C(z) = \frac{2\tau(\beta-1)(z-1)}{2\tau z - 2\tau + Te(z-1)} + 1 \quad (2.33)$$

Equation récurrente :

$$u(k) = -\frac{A_1}{B_1} \cdot u(k-1) - \frac{C_1}{B_1} \cdot \varepsilon(k) + \frac{C_1}{B_1} \cdot \varepsilon(k-1) \quad (2.34)$$

$$\text{Avec : } A_1 = Te - 2\tau ; B_1 = Te + 2\tau ; C_1 = 2(\tau - \beta\tau)$$

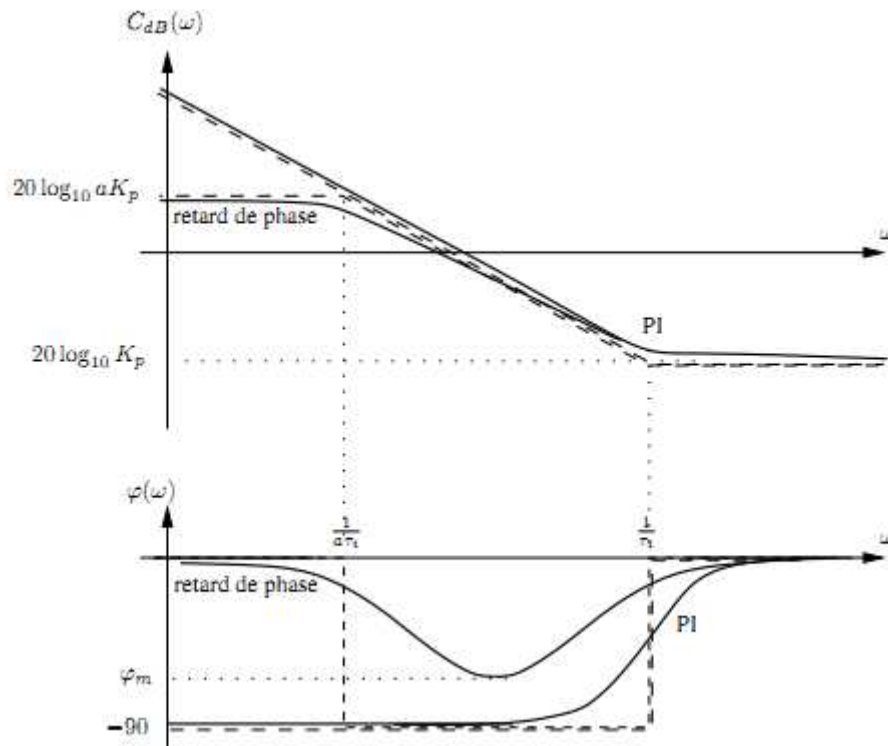


Figure (2.13) : Correcteurs PI et retard de phase

Ce correcteur possède une intégration. Il convient donc bien lorsque l'on souhaite annuler l'erreur statique d'un système de classe 0. Contrairement au correcteur PD, le PI est tout à fait réalisable physiquement ; c'est d'ailleurs le correcteur le plus utilisé. En revanche, du fait de son action intégrale, il présente l'inconvénient de saturer éventuellement l'entrée

du système. Il faut alors l'associer à un dispositif d'anti-saturation, constitué le plus souvent d'un simple écrêtage.

7.3.3 Le régulateur proportionnel-dérivée PD

Le régulateur P.D est une combinaison d'un régulateur P et d'un régulateur D. il élabore alors une commande qui peut être donnée par :

Equation récurrente :

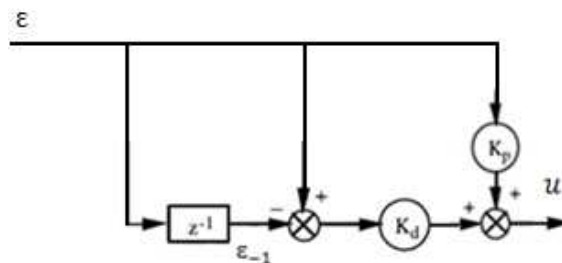
$$u(k) = (K_p + K_d). \varepsilon(k) - K_d. \varepsilon(k - 1) \quad (2.35)$$

Fonction de transfert :

$$C(z) = K_p + K_d. (1 - z^{-1}) \quad (2.36)$$

Ce correcteur n'apporte aucune précision. Il est utile en régime transitoire. L'action D, apporte une amélioration du comportement dynamique, en augmentant la vitesse de réaction de correcteur a la moindre variation du l'erreur. Elle anticipe l'évolution du système

Diagramme fonctionnel :



7.3.4 Correcteur à avance de phase [5]

Un correcteur à avance de phase a une fonction de transfert de la forme :

$$C(p) = K \frac{1 + \tau p}{1 + \alpha \tau p} ; \text{ avec } \alpha < 1 \quad (2.37)$$

Il peut être vu comme un correcteur PD approché, si $\alpha \ll 1$: l'action de ce correcteur approche en effet celle d'un correcteur PD aux pulsations inférieures à $\omega = \frac{1}{\alpha \tau}$, comme l'illustre la Figure (2.14)

Fonction de transfert discrète :

$$C(z) = \frac{1}{\alpha} - \frac{\frac{1}{\alpha} - 1}{\frac{2\tau\alpha(z-1)}{Te(z+1)} + 1}$$

Equation récurrente :

$$u(k) = \frac{A_2}{B_2} \cdot u(k-1) - \frac{C_2}{B_2} \cdot \varepsilon(k) + \frac{C_2}{B_2} \cdot \varepsilon(k-1)$$

Avec : $A_2 = 2\tau\alpha - Te$ $B_2 = 2\tau\alpha + Te$ $C_2 = 2\tau + Te$ Te : periode d'echantillonnage

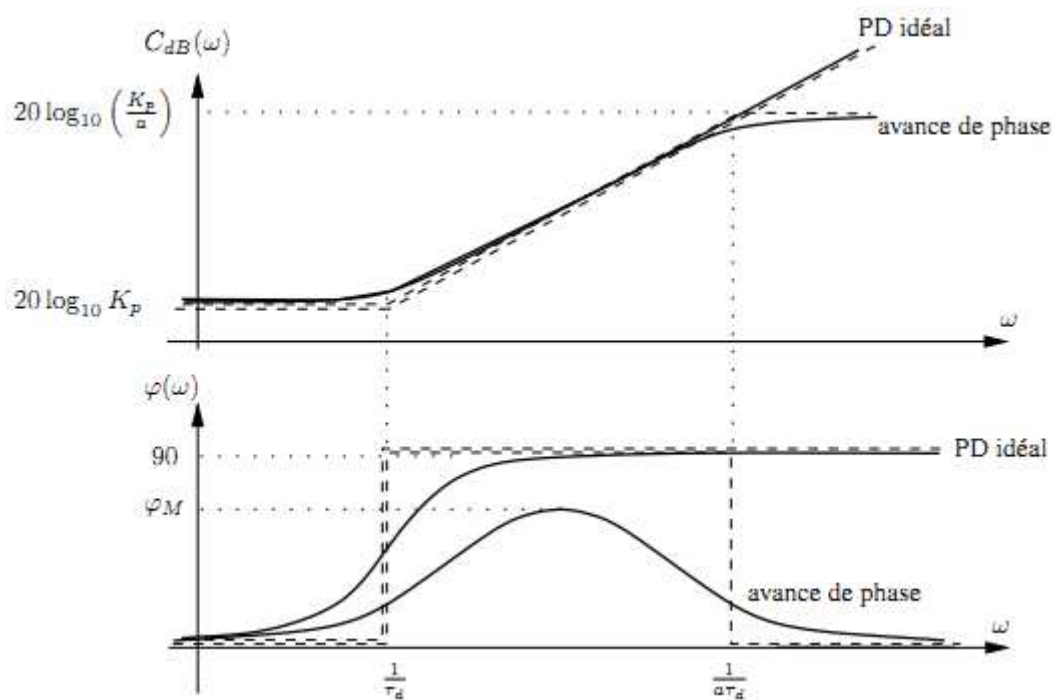


Figure (2.14) : correcteur à avance de phase et PD idéal

L'intérêt de ce type de correcteur est d'ajouter de la phase au système en boucle ouverte, dans une certaine bande de fréquence. Ceci peut permettre, sous certaines conditions, de rendre le système stable ou d'augmenter sa marge de phase. Pour cette raison le correcteur à avance de phase se prête bien à la correction des systèmes peu stables, comme les systèmes de classe supérieure ou égale à un. On montre que l'avance de phase maximale amenée par le correcteur vaut :

$$\sin \varphi_M = \frac{1-\alpha}{1+\alpha}$$

à la pulsation :

$$\omega_M = \frac{1}{\sqrt{\alpha} \tau_d}$$

A titre d'exemple, un coefficient $\alpha = 0.1$ occasionne une avance de phase maximale

$$\varphi_M = 54.9 \text{ deg.}$$

7.3.5 Le régulateur universel PID

Le régulateur P.I.D se base sur le régulateur P.I, auquel on ajoute une composante dérivée.

Equation discrète :

$$u(k) = K_p \cdot \varepsilon(k) + K_i \sum_{j=0}^k \varepsilon(j) + K_d [\varepsilon(k) - \varepsilon(k-1)] \quad (2.38)$$

Equation récurrente :

$$u(k) = u^i(k-1) + (K_p + K_i + K_d) \cdot \varepsilon(k) - K_d \cdot \varepsilon(k-1) \quad (2.39)$$

Fonction de transfert :

$$C(z) = K_p + K_i \frac{z}{z-1} + K_d \frac{z-1}{z} \quad (2.40)$$

Le régulateur P.I.D est un système du second ordre ; sa fonction de transfert possède un pôle nul et un pôle égal à 1. Comme le régulateur P.I, il est capable d'annuler l'écart de position en régime établi.

Ses pôles ne dépendent en aucune façon du réglage des actions Proportionnelle, Intégrale et Dérivée. Les coefficients K_p , K_i et K_d n'interviennent que dans le numérateur de la transmittance et par conséquent que sur ses zéros.

Diagramme structurel :

Le diagramme structurel d'un régulateur P.I.D peut prendre plusieurs formes comme indiqué dans les schémas ci-dessous :

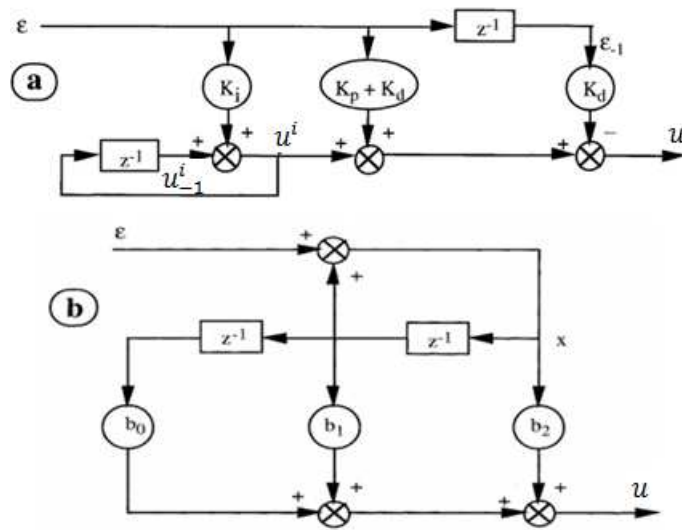


Figure (2.15) : Différent diagramme structurel d'un régulateur PID

La forme (a) dans la Figure (2.15) convient mieux à la programmation de ce régulateur. En effet, cette représentation s'appuyant sur l'équation récurrente suivante :

$$u = (K_p + K_i + K_d). \varepsilon + u_{-1}^i - K_d. \varepsilon_{-1} \quad (2.41)$$

Permet d'écrire l'algorithme de réglage à l'aide d'un pseudo-langage de programmation, du type :

Données : K_p
 K_i .
 K_d .
 $K_{pid} = K_p + K_i + K_d$.
 $x = u_{-1}^i$.
 $u = K_{pid}. \varepsilon + x - K_d. \varepsilon_{-1}$.

Sortie : u
 $x = x + K_i. \varepsilon$.
 $\varepsilon = \varepsilon_{-1}$.

Pour l'exécution de l'algorithme de réglage, on doit respecter la séquence prescrite. Tout de suite après le calcul de la grandeur de commande u , le calculateur de processus peut appliquer cette nouvelle valeur au système à régler. L'incrément de x par $K_i. \varepsilon$ et

ensuite l'échange de ε a ε_{-1} peuvent se faire, en temps masqué, dans l'intervalle jusqu'au prochain instant d'échantillonnage.

Dans ce cas on doit imposer une valeur initiale adéquate à la grandeur auxiliaire x . En plus, on a aussi besoin d'une valeur initiale pour l'écart de réglage ε_{-1} .

En général, on choisira au début : $x = 0$ et $\varepsilon_{-1} = 0$.

Remarque : bien souvent dans les régulateurs standard, aussi bien continus que numériques, l'action P est commune aux autres actions I et D. La structure d'un régulateur P.I.D se présente alors selon le schéma ci-dessous :

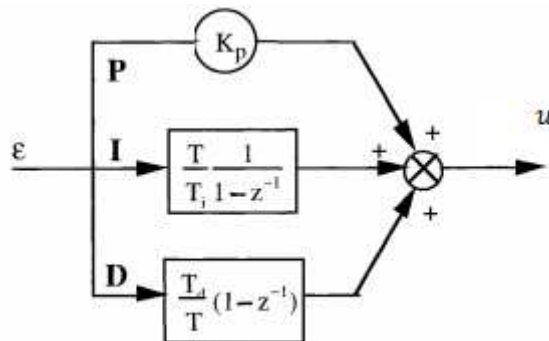


Figure (2.16) : présentation de la structure d'un régulateur PID

7.4 Choix et dimensionnement des régulateurs PID numériques

7.4.1 La boucle de régulation

La structure classique d'une boucle de régulation numérique se présente de la manière suivante :

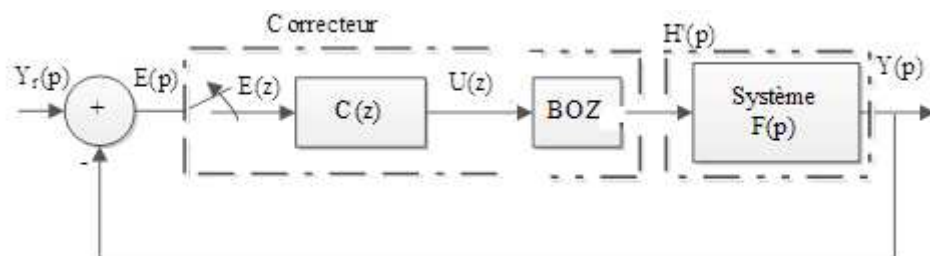


Figure (2.17) : schéma d'une boucle de régulation

L'ensemble Bloqueur-Processus a pour fonction de transfert :

$$H'(z) = (1 - z^{-1})Z\left[\frac{F(p)}{p}\right] \equiv \frac{B(z)}{A(z)} \quad (2.42)$$

Selon le type d'actions choisi, le régulateur présente une transmittance que l'on peut mettre sous la forme :

$$C(z) = \frac{R(z)}{S(z)} \quad (2.43)$$

Où seuls les coefficients de $R(z)$ sont dépendants des constantes de réglage des actions P, I et D.

De plus, on introduit dans la chaîne une possibilité de réglage du gain statique en boucle ouverte de telle sorte que la fonction de transfert en boucle ouverte du système corrigé soit

$$H_{bo}(z) = k \cdot C(z) \cdot H'(z) = k \cdot \frac{R(z)}{S(z)} \cdot \frac{B(z)}{A(z)} \quad (2.43)$$

Le choix du réglage des différentes actions du régulateur est conditionné par les caractéristiques présentées par le processus à contrôler.

Le choix et le dimensionnement des régulateurs-standard se basent sur le principe de la compensation des pôles du système à régler par les zéros du régulateur, de telle sorte que le degré de la fonction de transfert du circuit de réglage ouvert soit plus petit que celui de l'ensemble régulateur-processus.

7.4.2 Réglage des coefficients

Nous allons désormais voir comment trouver les valeurs à attribuer aux trois coefficients (T_i , K_p et T_d) de l'asservissement PID. Il existe deux façons de procéder, l'une par la modélisation, l'autre par l'expérimentation, sachant que souvent l'on complète la première à travers la deuxième.

Modéliser le système consiste à déterminer par le calcul son comportement et de là déduire des valeurs plausibles pour les coefficients. L'approche par l'expérimentation signifie que l'on va utiliser une réponse réelle du système pour régler d'abord grossièrement puis finement les coefficients.

7.4.3 Approche par modélisation

Le choix de procéder à la modélisation du système ou non est dicté par les contraintes inhérentes au système. Souvent la complexité des systèmes réels place la modélisation hors d'atteinte, mais dans certains cas, en chimie ou en mécanique par exemple, les règles qui régissent le système sont suffisamment simples pour que l'on puisse tenter de modéliser le système. D'autre part, il se peut que dans les cas où le système est particulièrement critique ou difficile d'accès (processus industriel lourd, système qui ne peut être mis hors service), il soit indispensable de modéliser le système afin d'avoir un jeu de coefficients suffisamment précis pour obtenir d'emblée un régulateur PID qui soit fonctionnel.

Nous donnons ici un exemple de modélisation d'un système simple; le moteur électrique.

Comme les équations de la dynamique et de l'électromagnétisme sont connues et utilisables, nous pouvons déterminer les équations qui régissent le système.

Voici les équations

$u(t) = e(t) + R \cdot i(t) + L \frac{di(t)}{dt}$	u Tension appliquée au moteur
$e(t) = K_e \cdot W(t)$	e Force électromotrice
$C_m(t) = K_m \cdot i(t)$	i Intensité traversant le moteur
$C_m(t) \cdot C_r(t) \cdot J_T \cdot \frac{d\omega(t)}{dt}$	ω Vitesse de rotation du rotor
	C_m Couple moteur généré
	C_r Couple résistant

On déduit de celle-ci la fonction de transfert du système :

$$\frac{\Omega(p)}{U(p)} = \frac{A}{1 + \frac{2h}{\omega_0} p + \frac{1}{\omega_0} p^2} \quad (2.44)$$

Avec : $A = \frac{1}{K_e}$ Gain statique

$h = \frac{R}{2} \sqrt{\frac{J_T}{K_e \cdot K_C \cdot L}}$ Facteur d'amortissement

$\omega_0 = \sqrt{\frac{K_e \cdot K_C}{L \cdot J_T}}$ Pulsation propre

A partir de celle-ci, on peut déduire la fonction de transfert du système asservi et donc les valeurs à attribuer aux coefficients pour obtenir une réponse qui nous satisfait.

Le but de cet exemple est de montrer comment un système simple régi par des équations différentielles linéaires simples, donne lieu à des valeurs complexes pour les coefficients. On peut ainsi justifier l'intérêt pratique de la méthode expérimentale

7.4.4 Approche par expérimentation

Nous présentons ici deux approches possibles pour la détermination des coefficients par expérimentation : une méthode avec l'utilisation du régulateur PID et l'autre qui ne l'utilise pas.

Nous indiquons aussi les avantages et inconvénients de ces deux méthodes.

L'existence de la deuxième méthode est justifiée par le fait que dans certains régulateurs matériels, les valeurs des coefficients ne sont pas modélisables, c'est le cas si par exemple le régulateur est un circuit électronique et que les coefficients sont déterminés par les composants (résistances, condensateurs)

7.4.4.1 Méthode d'oscillation limite en BF (Ziegler-Nichols)

- ❖ **Principe** : Amener le système dans un état d'oscillations puis en déduire les valeurs des coefficients.
- ❖ **Protocole** : Fixer T_i et T_d à 0 puis faire varier K_p jusqu'à obtenir des oscillations périodiques stables, c'est à dire non amorties et non amplifiées

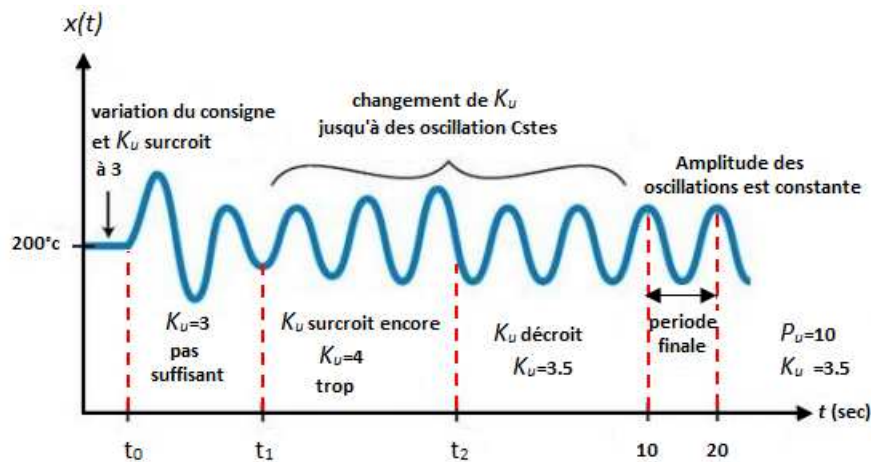


Figure (2.18) : Méthode de Ziegler-Nichols en BF [6]

On note : $K_u = K_p$ oscillations, $P_u =$ Période des oscillations

Alors il faut choisir $K_p = \frac{K_u}{1.7}$; $T_i = \frac{P_u}{2}$; $T_d = \frac{P_u}{8}$

Ces valeurs pour les coefficients sont celles que Ziegler et Nichols ont trouvées pour avoir un amortissement de l'amplitude des oscillations de 1/4 à chaque pseudo-période.

❖ **Avantages**

- La méthode est facile à mettre en œuvre physiquement et au point de vue calcul
- Elle peut être appliquée à un système déjà en production et permet une adaptation automatisée du régulateur pour s'adapter à l'évolution des paramètres intérieurs (usure) et extérieurs (environnement) au système

❖ **Inconvénients**

- Le système peut devenir instable ou passer dans des états dangereux (par exemple pour les systèmes chimiques)
 - La méthode peut nécessiter beaucoup de temps si le système réagit très lentement (jours, semaine dans le cas de certaines réactions chimiques)

7.4.4.2 Méthode de la réponse indicielle (Ziegler –Nichols en BO) [6]

S'applique pour les systèmes dont la réponse indicielle est de type apériodique possédant un retard.

❖ **Protocole** : On applique un créneau au système et on enregistre sa réponse

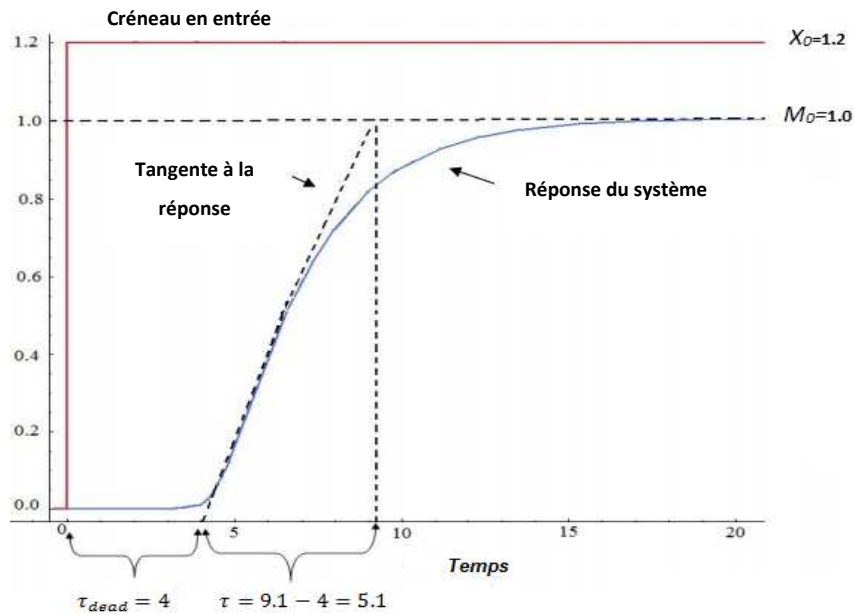


Figure (2.19) : Méthode de la réponse indicielle

On pose :

$$K_0 = \frac{X_0}{M_0} \cdot \frac{\tau}{\tau_{dead}} \quad (2.45)$$

On a alors :

$$K_p = 1.2 \cdot K_0 \quad K_i = 2 \cdot \tau_{dead} \quad K_d = 0.5 \cdot \tau_{dead}$$

❖ **Avantages**

- Cette méthode ne nécessite pas d'avoir un système déjà asservi, ni de deviner une valeur pour K_p

❖ **Inconvénients**

- Elle nécessite τ_{dead} matériel : en τ ment d'une courbe de réponse.
- Elle oblige à mettre le système "offline"
- Si la réponse du système est trop différente de la réponse montrée ci-dessus, cette méthode donne des valeurs qui peuvent ne pas convenir du tout (certains systèmes par exemple ne sont pas du tout réglables par des régulateurs PID)

7.5 Limites et amélioration du régulateur

Comme nous l'avons vu précédemment, un asservissement PID peut comporter un certain nombre de limites qui, si elles ne sont pas prises en compte, peuvent altérer le bon fonctionnement du système voire le détériorer.

7.5.1 Contraintes électrique et matérielles

Tout d'abord, n'oublions pas que nous travaillons sur des systèmes réels donc ils possèdent des caractéristiques électriques (courant maximum, tension maximale, ...), matérielles (vitesse maximale, ...). Par conséquent, la consigne qui sera envoyée par l'asservissement au système se doit d'être en accord avec les contraintes globales de notre système.

Prenons un exemple très simple, un moteur supporte une tension en entrée maximale de V_1 or pour atteindre un résultat optimal l'asservissement calcule une tension de consigne de $V_2 > V_1$. Si cette consigne n'est pas modifiée, le moteur va être endommagé. Comment donc pallier ce problème?

Une solution consiste à rajouter en sortie de l'asservissement PID un module d'écrêtage. Ainsi, si la tension de consigne fournie par l'asservissement est supérieure à la tension maximale supportée par le système alors la tension sera modifiée pour valoir V_{max} .

Si $V_{consigne} > V_{max}$, Alors $V_{consigne} = V_{max}$

Nous pouvons par ce biais obtenir un asservissement PID qui respectera les contraintes électriques et matérielles du système à contrôler.

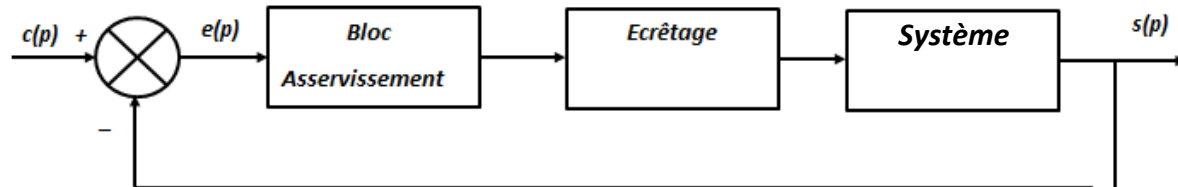


Figure (2.20) : Asservissement complété par un écrêtage

7.5.2 Instabilité de la dérivation [7]

Dans un asservissement PID, le terme Dérivation peut parfois poser problème. En effet, prenons le cas d'un système fortement bruité comme un asservissement de la vitesse d'un moteur. Si on dérive la vitesse on obtient l'accélération or celle-ci peut s'avérer très instable si la vitesse mesurée est trop bruitée. Comme illustré ci-dessous, l'asservissement est alors fortement altéré et inutile

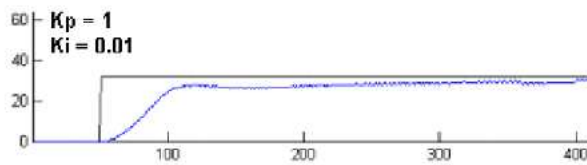


Figure (2.21) : Asservissement PI

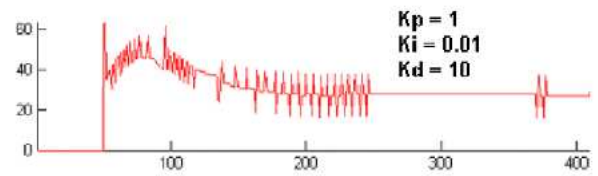


Figure (2.22) : Asservissement PID

Deux solutions sont alors envisageables :

- La première consiste à effectuer un filtrage à l'aide d'un filtre passe-bas afin de limiter le bruit.

Cette solution peut parfois poser problème car un filtrage entraîne une perte d'information qui peut nuire à l'asservissement.

- La seconde et la plus simple est d'enlever le paramètre Dérivé de l'asservissement, on obtient donc un asservissement PI. C'est cette dernière solution qui est la plus souvent utilisée.

Il est intéressant de noter que ce type de problème intervient essentiellement dans le cas d'asservissement en vitesse. Dans la plupart des autres systèmes, la dérivée est peu bruitée permettant un asservissement d'autant plus efficace.

8. Conclusion

Les méthodes expérimentales présentées précédemment permettent d'avoir de bonnes estimations génériques pour un système raisonnablement simple (c'est à dire réglable par un PID).

Pendant il faut très souvent procéder à un réglage des coefficients jusqu'à obtenir un système respectant le cahier des charges. Pour cela on utilise les règles suivantes:

- Si K_p augmente alors la montée sera plus rapide mais il y aura plus de dépassement
- Si T_i augmente alors la montée sera plus rapide et l'erreur statique sera plus faible mais le régime stationnaire sera plus long à atteindre

- Si T_d augmente alors le dépassement diminuera et le temps d'établissement du régime stationnaire aussi, mais la sensibilité au bruit augmentera.

CHAPITRE 3

1. Introduction

Nous consacrons ce troisième chapitre à la simulation et réalisation pratique du procédé, celui-ci se décompose en étage distincts respectant la composition et l'architecture des systèmes régulé par régulateurs numérique (PID, Avance et retard de phase).

Pour la réalisation du programme inscrit dans la mémoire flash de notre microcontrôleur, nous avons utilisé le compilateur CCS «PIC C Compiler» qui est téléchargeable gratuitement sur internet.

Nous nous étalerons sur chaque étage, ses composants spécifiques et son action sur le fonctionnement global.

2. Cahier de charge

L'objectif de ce projet est de concevoir un module effectuant les fonctions de corrections numériques PID, avance ou retard de phase, selon le choix du correcteur utilisé, d'un signal d'entrée analogique. Les valeurs K_d , K_p et K_i sont réglables ou réinitialisables par l'utilisateur à tout instant et affichées sur un écran LCD 2*16 caractères. Le PIC utilisé dispose d'un étage de conversion analogique numérique, le signal de sortie est analogique. La structure générale de l'application est celle du schéma ci-dessous.

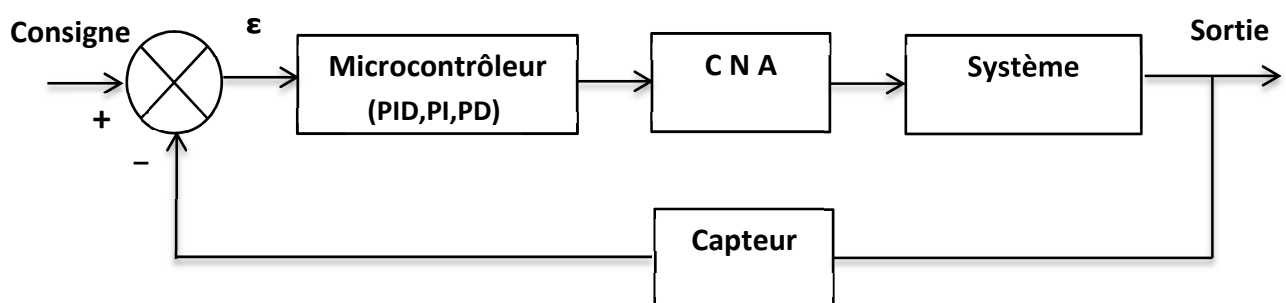


Figure (3.1) : La structure générale de l'application

3. Principe de fonctionnement

Le module de correction PID, avance et retard de phase à réaliser propose le réglage des valeurs K_p , K_i et K_d par l'utilisateur grâce à un clavier. Les réglages sont visualisés sur un afficheur LCD contrôlé par un driver intégré dans le PIC (programme). Les corrections sont effectuées par un microcontrôleur qui prend en charge la conversion analogique-numérique du signal d'entrée. Le signal numérique corrigé est ensuite reconverti en signal analogique par un convertisseur numérique-analogique. Enfin le signal est amplifié à l'aide d'un amplificateur avant de commander le système.

La réponse est récupérée en sortie de système et une fois filtrée elle est comparée au signal d'entrée afin de déterminer l'erreur pour effectuer la correction souhaitée.

Un correcteur alliant la combinaison des trois actions PID. Peut être assimilé à un correcteur série au sein d'un système.

4. Guide d'utilisation

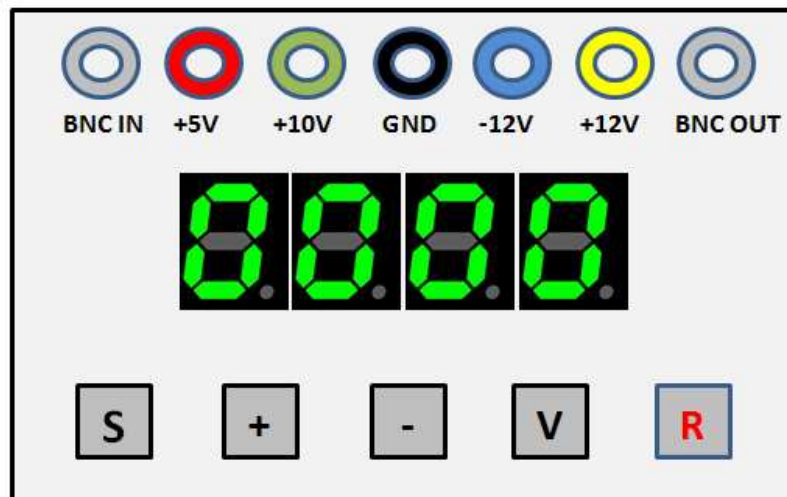


Figure (3.2) : Le boîtier de la réalisation pratique de régulateur

Pour s'assurer du bon fonctionnement du module d'asservissement, il faut suivre les étapes suivantes lors de sa mise en marche.

Après avoir alimenté les amplificateurs opérationnels avec les tensions -12V et +12V (bleu et jaune), le PIC avec +5V (rouge) et le convertisseur DAC 802 avec +10V (verte), on peut passer aux tests et expliquer le fonctionnement du correcteur conçu.

Après avoir réglé le GBF, on le relie à l'aide de câbles coaxiaux branchés sur les fiches BNC. On branche un oscilloscope en sortie pour visualiser le signal corrigé. Le signal d'entrée ne peut excéder +5V et -5V.

C'est à partir de ce moment-là que l'on peut allumer l'alimentation général du montage on le basculant sur la position ON.

Une fois la maquette alimentée et un signal d'entrée injecté, on doit d'abord initialiser tous les paramètres (K_p , K_i et K_d), ensuite choisir un correcteur voulu (soit PID, PI ou PD) à l'aide de bouton d'incrémentement (BP2 ou +) puis validé le choix de correcteur par le bouton de validation (BP4 ou V).

Après avoir choisi le model de correcteur on passe aux réglages des paramètres, on sélectionne les paramètres par impulsion sur (BP1 ou S), commençant par le premier paramètre K_p , incrémenté par le bouton poussoir (BP2 ou +) ou décrétementé par le bouton poussoir (BP3 ou -) entre 0 et 9 puis le validé, le deuxième paramètre est sélectionné automatiquement, incrémenté ou décrétementé K_i entre 0 et 9 puis le validé, de même pour le troisième paramètre K_d .

C'est à partir de là qu'on la conversion analogique-numérique commence et le calcul de l'asservissement se fait, les paramètres sont directement visualisés sur l'afficheur LCD.

L'opération de modification de correcteur ou des paramètres de corrections peut être effectué à tout instant. Lorsque l'on appuie sur la touche "Reset", tous les paramètres sont remis à 0. On peut ainsi réaliser un nouvel asservissement.

5. Système à microprocesseur

5.1 Définition d'un microcontrôleur

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires, unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un gigahertz) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Un microcontrôleur peut être programmé afin qu'il effectue une ou des tâches précises au sein d'un appareil électronique. Mais les μC récents peuvent être reprogrammés et ceci grâce à leur mémoire permanente de type FLASH.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc. Les éléments essentiels d'un microcontrôleur sont

- Un microprocesseur (CPU ou μP) qui traite des données logiques et arithmétiques et réalise le séquençage des opérations
- Une mémoire de programme
- Une mémoire de données
- Des interfaces d'entrée sortie qui permettent de communiquer avec des périphériques externes.

5.1.1 L'unité de traitement (CPU : Central Processing Unit)

L'unité de traitement numérique exécute les instructions du programme (codées sur 14 bits dans notre cas). Il traite et produit des opérations sur des informations numériques uniquement.

5.1.2 Les mémoires

Elles se divisent en trois blocs distincts :

- la mémoire de programme de type EPROM Flash. Cette mémoire est destinée à être lue uniquement. Elle contient les instructions que doit exécuter la CPU.
- la mémoire de données de type RAM est utilisée pour le stockage temporaire des données et résultats. La CPU peut lire cette mémoire mais aussi modifier son contenu.
- la mémoire de sauvegarde de type EEPROM qui peut être lue et écrite depuis le programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents.

5.1.3 L'horloge

Signal qui cadence l'exécution des instructions. Chacune des instructions du programme est traitée en un cycle machine : une période de l'horloge avec une division interne de la fréquence par 4. (Exemple : avec une horloge externe de 8MHz, la fréquence cycle est de 2 MHz soit une durée d'exécution d'une instruction de 500ns).

5.1.4 Les ports d'entrée / sortie

- Pour communiquer avec l'extérieur le PIC dispose de plusieurs ports.
- Les ports sont bidirectionnels, ce qui signifie qu'ils peuvent être configurés et utilisés comme des entrées ou des sorties.

5.2 Description du PIC 16F876

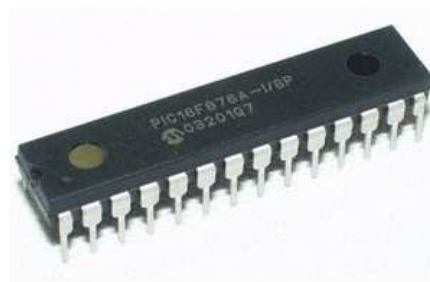


Figure (3.3) : Photographie du PIC 16F876

5.2.1 Caractéristiques essentiels du PIC 16F876 [8]

- Consommation : moins de 2mA sous 5V à 4 MHz.
- Architecture RISC : 35 instructions de durée 1 ou 2 cycles.
- Durée du cycle : Période de l'oscillateur quartz divisée par 4 soit 200 ns pour un quartz de 20 MHz.
- Deux bus distincts pour le code programme et les data.
- Code instruction : mot de 14 bits et compteur programme (PC) sur 13 bits, ce qui permet d'adresser 8 K mots (de h'0000' à h'1FFF')
- Bus DATA sur 8 bits.
- 22 Ports Entrée-Sortie bidirectionnels pouvant produire 25 mA par sortie.
- 4 sources d'interruption :
 - Externe par la broche partagée avec le Port B : RB0
 - Par changement d'état des bits du Port B: RB4 RB5 RB6 ou RB7
 - Par un périphérique intégré dans le chip: écriture de Data en EEPROM terminée, conversion analogique terminée, réception USART ou I2C.
 - Trois TIMERS avec leurs percales, TMR0, TMR1, TMR2.
- 2 Compteurs 8 bits et 1 compteur 16 bits avec pré diviseur programmable.
- Convertisseur analogique 10 bits à 5 canaux.
- UART pour transmission série synchrone ou asynchrone.
- Interface I2C.
- 2 modules pour PWM avec une résolution de 10 bits.
- Interface avec un autre micro: 8 bits + 3 bits de contrôle pour R/W et CS.
- 368 Octets de RAM
- 256 Octets d'EEPROM Data.
- 8K mots de 14 bits en EEPROM Flash pour le programme (h'0000' à h'3FFF').
- 1 registre de travail : W et un registre fichier : F permettant d'accéder à la RAM ou aux registres internes du PIC. Tous les deux sont des registres 8 bits.

PORTA : 6 entrées -sorties. 5 entrées du CAN. Entrée CLK du Timer 0.

PORTB : 8 entrées-sorties. 1 entrée interruption externe. Clk et Data pour prog.

PORTC : 8 entrées-sorties. Clk Timer1 et PWM1. USART. I2C.

5.2.2 Structure interne du microcontrôleur 16F876

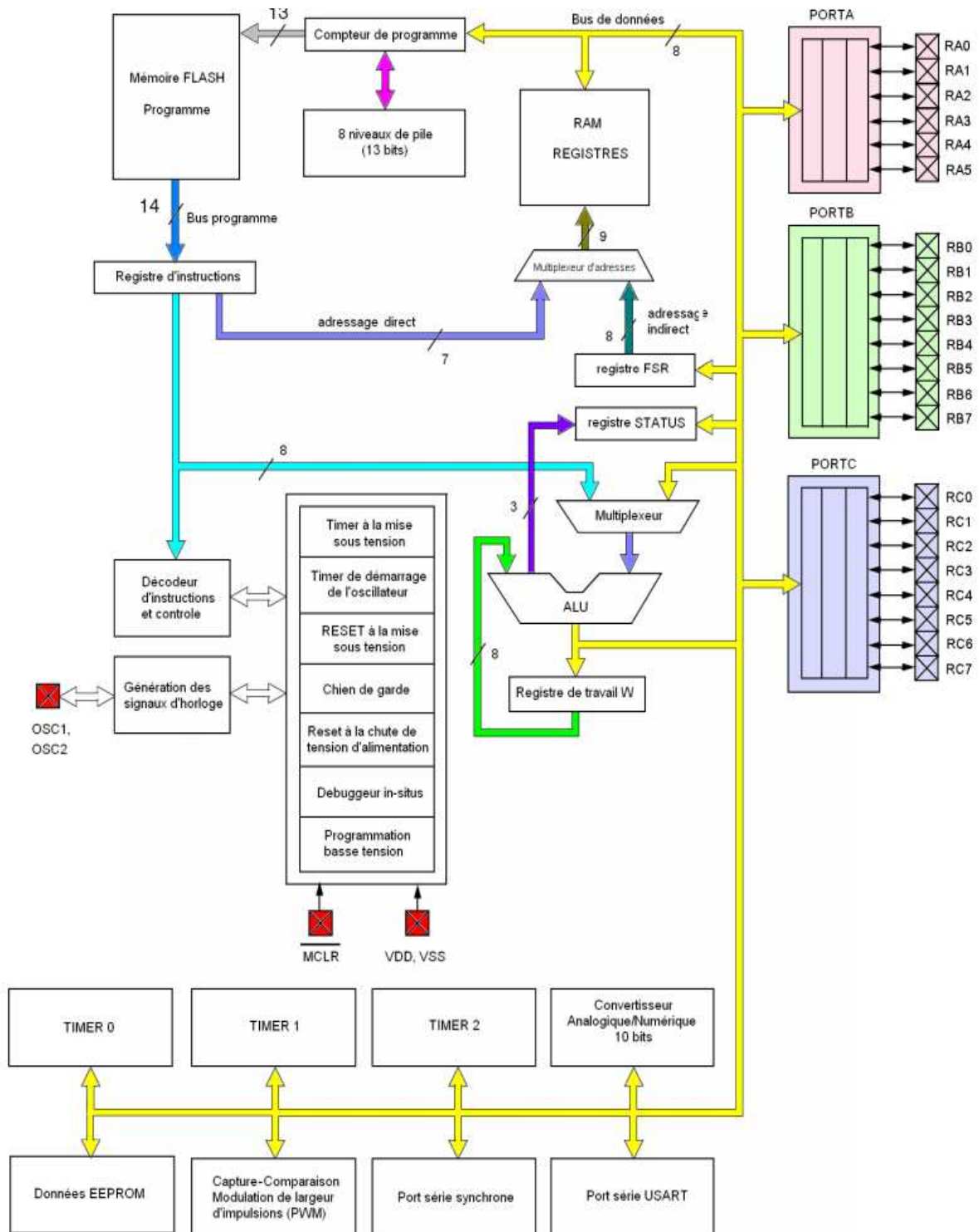


Figure (3.4) : La structure interne du 16F876

5.2.3 Brochage du 16F876 [8]

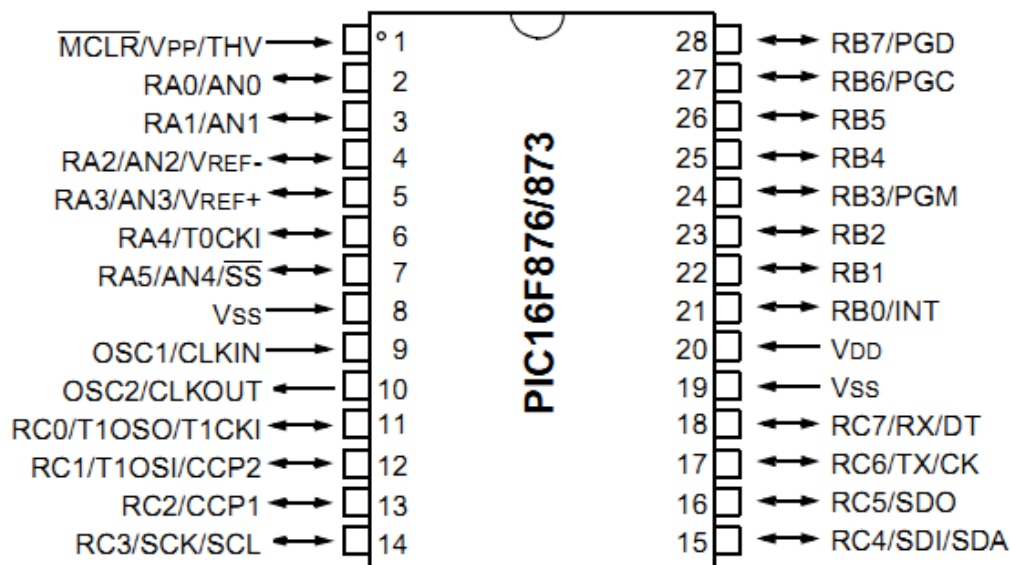


Figure (3.5) : Brochage de 16F876 en boîtier DIP

5.2.4 Le convertisseur analogique numérique CAN du 16F876

Le principe de la conversion analogique-numérique est d'attribuer une grandeur numérique correspondant à un état physique. Le CAN convertit donc une tension dite analogique en une valeur numérique.

5.2.4.1 Résolution du convertisseur analogique-numérique

La résolution d'un CAN correspond à la plus petite mesure réalisable par le CAN, autrement dit la précision du CAN.

Dans notre cas, on a un CAN de 10 bits, ce CAN peut fournir 1024 valeurs différentes, La plus petite différence de tension mesurable à ses bornes va être fonction de la plage de mesure en tension utilisée. Si la tension est comprise entre 0 et 5V, la mesure minimale correspond à : $5/1024 = 4.88 \text{ mV}$. On dira que la résolution d'un CAN 10 bits pour une tension de plage de mesure de 5V est de 4.88 mV.

D'une façon générale, la résolution est égale à la plage de tension de la mesure divisée par le nombre de bits du CAN :

$$\text{Résolution} = \frac{\text{plage de mesure}}{\text{nombre de bits du CAN}}$$

5.2.4.2 Détail de fonctionnement du CAN contenu dans le 16F876

Le microcontrôleur 16F876 intègre un module de conversion analogique-numérique qui est associé à un multiplexeur analogique multivoie des entrées.

Autrement dit on peut réaliser des mesures sur plusieurs voies d'entrée analogiques (jusqu'à 5). Ces voies analogiques correspondent aux broches du port A. Module CAN 10 bits implique que le résultat de la conversion est un nombre de 10 bits, c'est-à-dire compris entre 000H et 3FFH (11 1111 1111 en binaire ou 1023 en décimal). Ceci correspond donc à 1024 niveaux de mesure, ce qui pour 5V donne une résolution de 4,88 mV, précision largement suffisante. Le temps de conversion de ce module est de l'ordre de : 40 µs ce qui permet en pratique de réaliser jusqu'à 25000 mesures par seconde.

5.2.4.3 Structure interne du convertisseur du 16F876 [8]

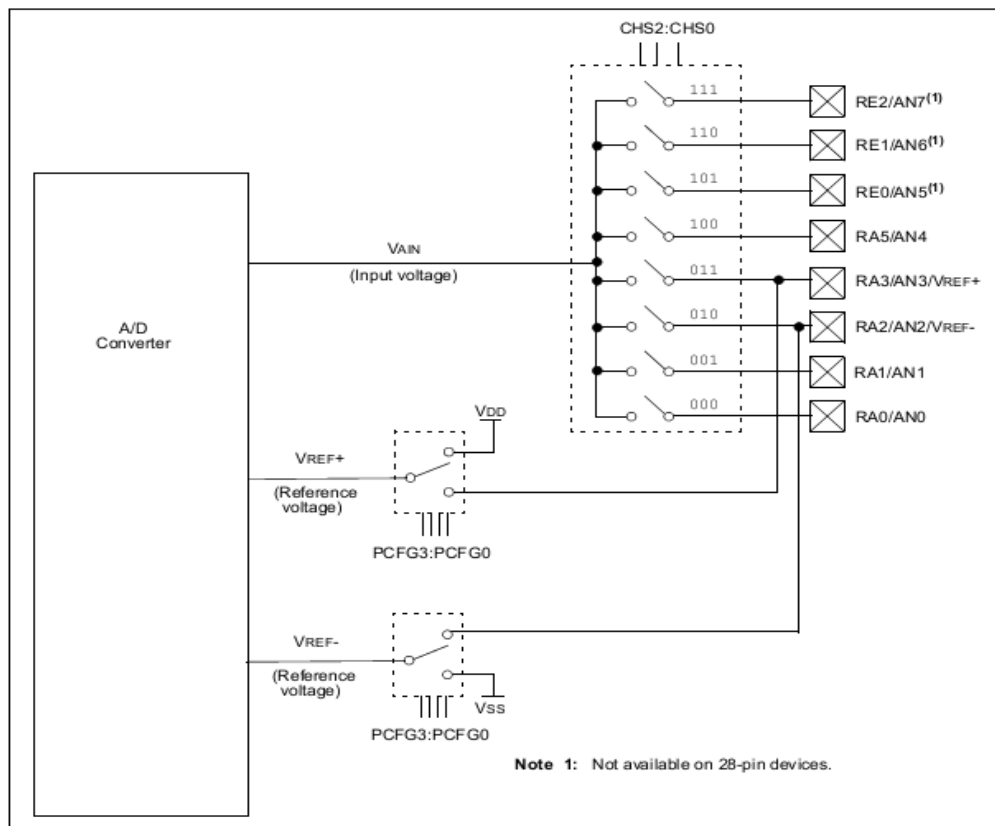


Figure (3.6) : Structure interne du CAN

5.2.4.4 La fréquence d'échantillonnage

L'oscillateur utilisé étant fixé à 4 MHz, donc la fréquence d'échantillonnage maximum que l'on puisse choisir est de 500KHz.

Une relation existe entre la fréquence maximale du signal d'entrée et la fréquence d'échantillonnage, en effet, pour éviter le phénomène de recouvrement spectral, il faut respecter le théorème de Shannon (chapitre 1).

5.3 Convertisseurs numérique analogique (DAC0800 / DAC0802)

Les circuits intégrés industriels utilisent un principe similaire au précédent. A titre d'exemple, le DAC0800 de Motorola utilise des commutateurs analogiques en guise

d'interrupteurs. Le DAC0800 est un CNA 8 bits rapide à sorties différentielles en courant. La tension différentielle en sortie peut atteindre 20 V en chargeant avec deux résistances comme l'indique la Figure(3.8).

5.3.1 Caractéristiques principales du DAC0800 / DAC0802 [9]

- Temps de conversion : 100 ns
- Erreur en pleine échelle : ± 1 LSB
- Tension en sortie jusqu'à 20 V
- Sorties complémentaires en courant
- Interfaçage direct en TTL et CMOS
- Alimentation de $\pm 4,5V$ à $\pm 18V$
- Basse consommation : 33 mW à $\pm 5V$
- Conversions unipolaires ou bipolaires
- Coût modéré.

5.3.2 Brochage de DAC0802

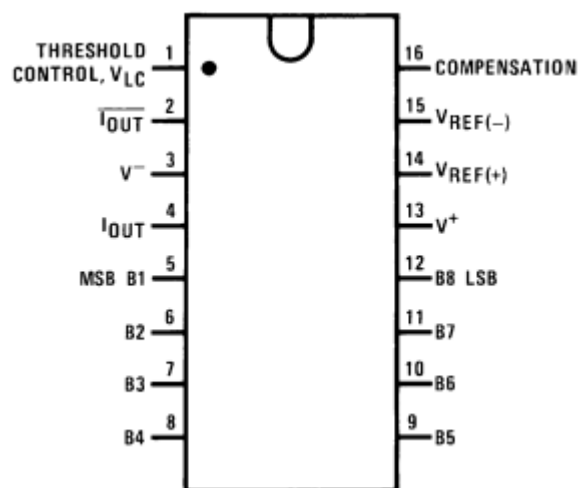


Figure (3.7) : Brochage de DAC 0802

5.3.3 Structure interne de DAC0802 [9]

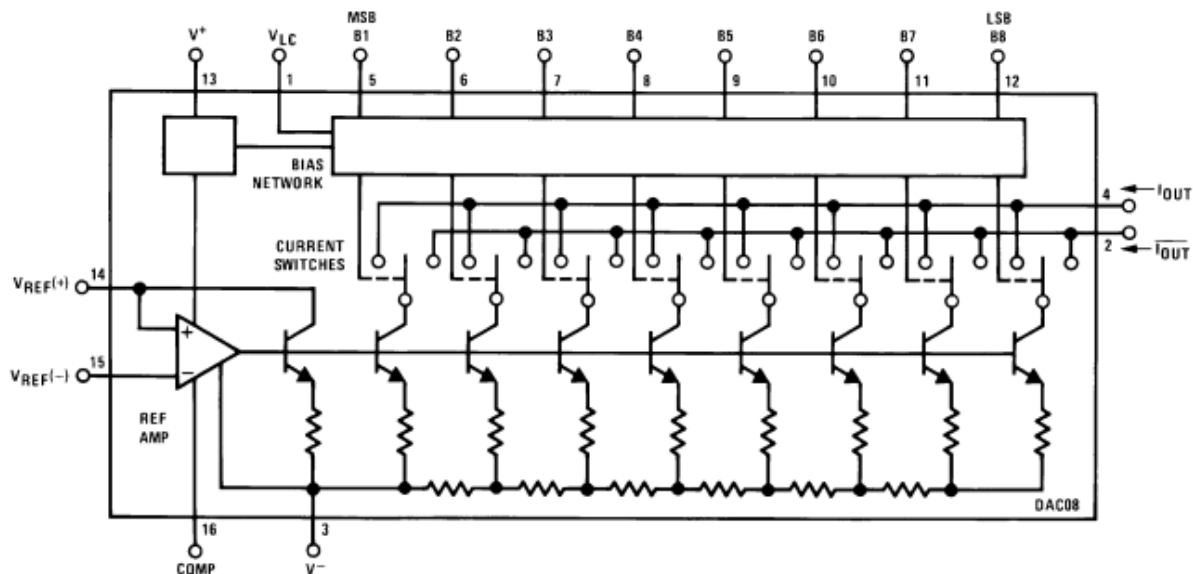


Figure (3.8) : Architecture interne DAC 0800 (Doc. Motorola).

La structure interne indiquée à la figure ci-dessus, montre le bloc de décodage qui reçoit les 8 bits à convertir (broches 5 à 12) et contrôle les commutateurs. Ici les courants sont entrant dans le convertisseur et leurs poids binaires sont obtenus par des miroirs de courant. Les broches 4 et 2 fournissent les courants complémentaires, image de la conversion. La grandeur de référence, en tension ou en courant, est appliquée au niveau des broches 14 et 15. La notice complète fournit le mode de câblage et de nombreuses applications.

6 Conception matériel

6.1 Choix des composants

Le cahier des charges nous impose d'utiliser certains composants. Ils ont été choisis pour leurs caractéristiques techniques et les performances nécessaires à la bonne exécution du système.

Le microcontrôleur est le cœur du système. Il doit effectuer en permanence le calcul de la correction, mais il doit aussi gérer l'envoi de données à l'afficheur LCD afin d'effectuer un affichage cohérent des paramètres du correcteur.

Le PIC doit également prendre en compte les interruptions engendrées par les modifications possibles des paramètres à l'aide de boutons poussoirs. Il est de plus équipé d'un convertisseur analogique-numérique (CAN) 10 bits en entrée.

En sortie du PIC nous avons choisi d'utiliser un convertisseur numérique-analogique (CNA) 8 bits de référence DAC 0802. Il délivre deux sorties qui sont associées à un soustracteur pour avoir une seule tension de sortie qui doit être adaptée au signal d'entrée, ce convertisseur est placé avant l'adaptateur de sortie.

L'utilisation de ce PIC et du CAN impose quelques contraintes. Il ne peut recevoir en entrée qu'un signal compris entre 0 et 5 Volts. Or nous effectuons l'asservissement d'un système du signal compris entre -5 et +5 Volts en entrée du PIC.

Pour remédier à ce problème on utilise un étage d'adaptation du signal d'entrée de microcontrôleur.

6.2 Schéma synoptique

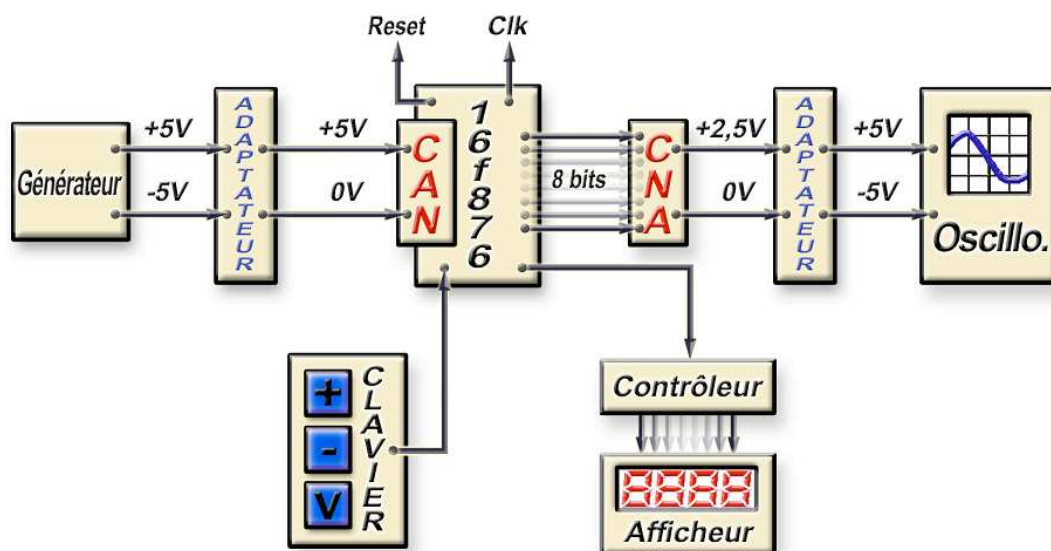


Figure (3.9) : Schéma synoptique de la réalisation

6.3 Etage d'adaptation du signal d'entrée

Le but de cette étage est de convertir la tension de sortie de notre système de [-5v +5v] à [0v +5v] qui sera injecter sur le PIC, vue qu'il n'accepte que une tension entre 0 et 5 volts, afin de faire la conversion analogique numérique et le calcul de la commande.

Équation à réaliser :
$$Vs = \frac{Ve+5}{2}$$

Pour réaliser cette opération, on utilise deux amplificateurs opérationnels (AOP) de référence TL084 en série. Le premier est utilisé en montage sommateur inverseur, on additionne le signal d'entrée de régulateur à une tension de 5 V afin de le décalé vers le haut ; le second en montage amplificateur inverseur recevant sur sa borne V- (borne inverseuse) la sortie du premier AOP et la borne V+ (borne non-inverseuse) reliaer vers la masse, illustré par la figure suivante.

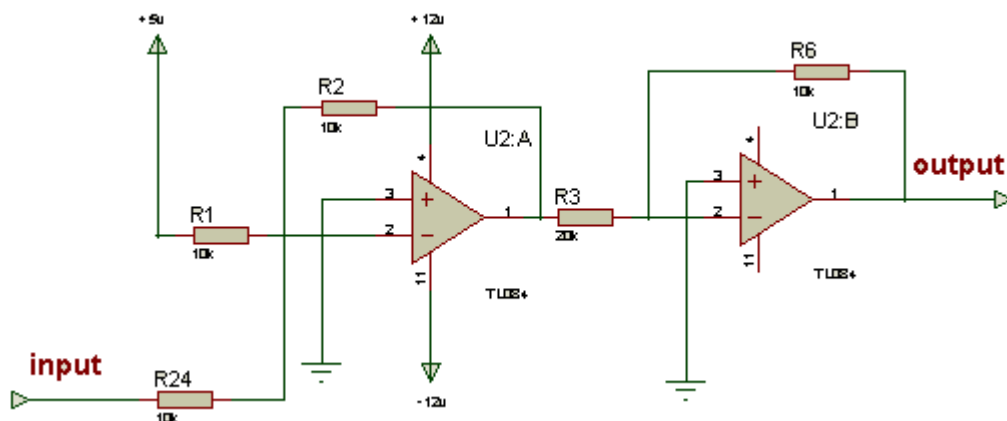
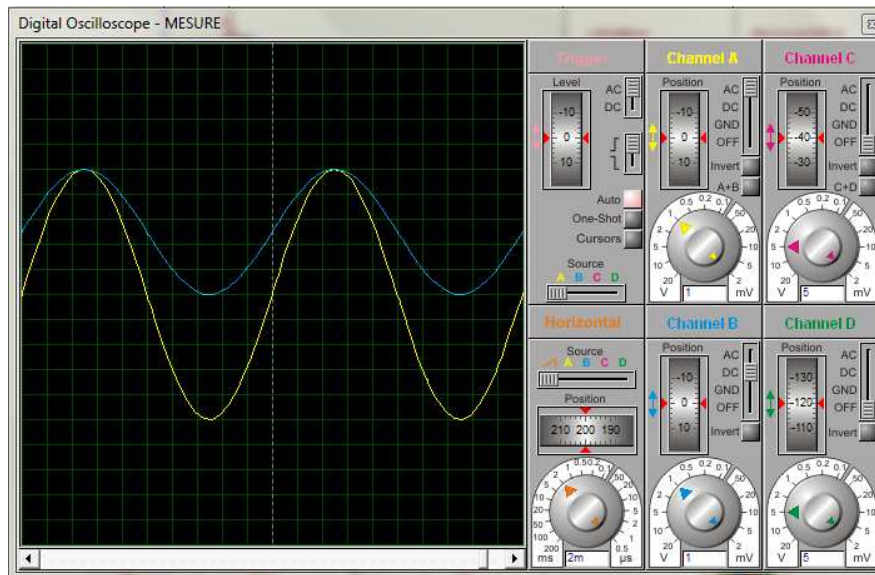


Figure (3.10) : Circuit d'adaptation du signal d'entrée

Une fois les calculs théoriques effectués, nous avons simulé l'ensemble du montage d'adaptation à l'aide du logiciel Proteus (ISIS). Les courbes que nous obtenons après simulation de l'adaptation d'entée données à la figure (3.11).



_____ : signal d'entrée

_____ : signal adapté

Figure (3.11) : simulation de l'adaptation du signal d'entrée

6.4 Etage de conversion

6.4.1 Temps de conversion

Le temps de conversion est égal à 12 TAD, TAD est le temps de conversion d'un bit, il dépend de la fréquence du quartz et du pré diviseur (div) choisi : $TAD = div \cdot 1/fosc$. Le choix de div doit être ajusté pour que TAD soit $\geq 1,6 \mu s$

Div \ Quartz	20 Mhz	5 Mhz	4 Mhz	2 Mhz
2	0.1 μs	0.4 μs	0.5 μs	1 μs
8	0.4 μs	1.6 μs	2 μs	4 μs
32	1.6 μs	6.4 μs	8 μs	16 μs

Tableau (3.1) : Temps de conversion d'un bit TAD (les cases grisées sont hors plage d'utilisation)

Avec un quartz de 4 MHz, il faut choisir div=8 ce qui donne TAD = 2 μs soit un temps de conversion : TCONV = 24 μs

6.4.2 Temps d'acquisition

Temps d'acquisition = TACQ = Tc + CT + 2 μs

Tc : temps de charge du condensateur = (Ric + Rss + Rs) C Ln(2047)

Ric = Résistance d'interconnexions, elle est inférieure à 1k

Rss = Résistance du l'interrupteur S (Sampling switch), elle dépend de la tension d'alimentation Vdd. Elle est égale à 7kΩ pour Vdd=5V

Rs : Résistance interne de la source du signal analogique. Microchip recommande de ne pas dépasser 10 kΩ

C : Capacité de blocage = 120 pF

CT : Coefficient de température = (Tp - 25°C) 0.05 μs/°C

Tp = Température Processeur, voisine de 45°C en temps normal

Exemple :

Ric = 1k, Rss = 7k, Rs = 2k, Tp = 45 °C

Tc = 10k x 120pF x Ln(2047) = 9 μs

CT = 20 x 0.05 μs = 1 μs

TACQ = 2 + 9 + 1 μs = 12 μs

Dans cette conception on utilise deux convertisseurs, le premier est un convertisseur analogique numérique celui du microcontrôleur 16F876, ces paramètre de réglage sont intégré dans le programme (son activation, période d'échantillonnage, choix de canal d'entrée utilisé).

Le deuxième est un DAC 0802 convertisseur numérique analogique 8 bits qui convertie le signal de sortie du PIC (la commande calculé) en un signal analogique, il recommande une

configuration minimale, ce DAC a deux sortie distincte I_{OUT} et $\overline{I_{OUT}}$ alors on lui associe un soustracteur pour avoir une sortie de référence 0 V, la figure suivante illustre etage de configuration du convertisseur numérique analogique.

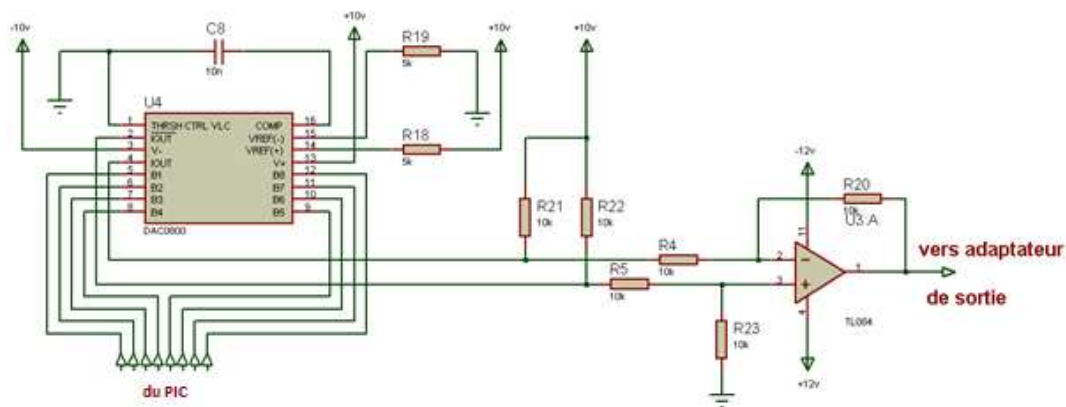


Figure (3.12): Configuration de convertisseur DAC0802

6.5 Etage d'adaptation du signal de sortie

Le but de cette étage est de convertir la tension de sortie de convertisseur numérique analogique DAC0802 de $[0v +5v]$ à $[-5v +5v]$ qui sera injecté sur le système à corriger.

Équation à réaliser : $V_s = 2V_e - 5$

On utilise à nouveau deux AOP en série. Le premier en tant qu'amplificateur inverseur et le second en montage soustracteur. La borne V_- du premier AOP reçoit en entrée le signal de sortie du CNA, et celle du second AOP est connectée à la sortie du premier. Les bornes V_+ des deux AOP sont reliées à la masse, illustrée par la figure (3.13).

Le choix des résistances est fait de telle sorte que :

$$R_8 = 2 R_7 \quad \text{et} \quad R_7 = R_9 = R_{10} = R_{11}$$

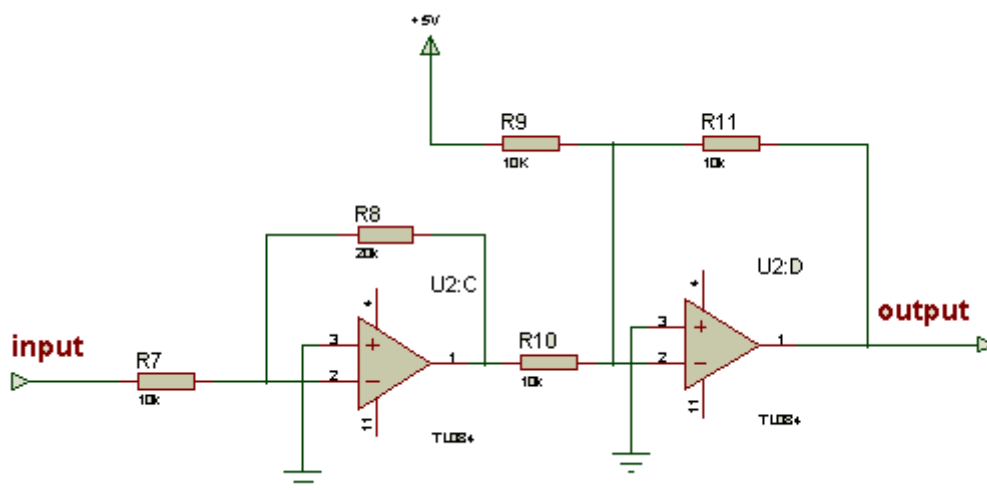
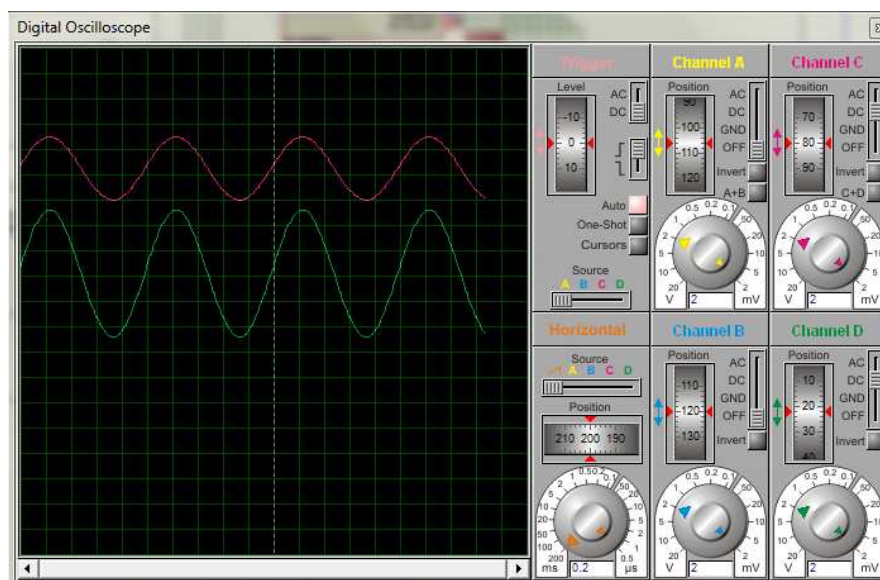


Figure (3.13) : Circuit d'adaptation du signal de sortie

Une fois de plus après avoir effectué les calculs théoriques nous avons simulé l'ensemble et voici les résultats obtenus :



_____ : signal après conversion (CNA) _____ : signal de sortie (adapté)

Figure (3.14) : Simulation de l'adaptation du signal de sortie

Nous remarquons que le signal est réadapté et l'amplitude du signal de sortie a bien une amplitude comprise entre -5 et +5 volts, comme le signal injecté à l'entrée du module.

6.6 Le clavier

Le clavier est un périphérique d'entrée essentiel qui nous permet de saisir les paramètres du correcteur (la commande) pour l'exécution du programme.

Il se compose de 5 boutons :

Un bouton de sélection de correcteur (avance de phase, retard de phase, ou PID), un bouton incrémentation, un bouton de décrémentation, un bouton de validation des paramètres et un bouton de reset pour réinitialisé le programme à l'instant voulu.

Les boutons poussoirs sont les plus simples dispositifs qui peuvent changer l'état des entrées du microcontrôleur. Or, de tels dispositifs utilise des contacte mécaniques qui produisent des rebondissent. Ces derniers se présentent sous forme de plusieurs impulsions sur une durée de 0.01 à 100 ms.

Pour atténuer ce phénomène, nous allons utiliser un condensateur en parallèle avec le bouton et une résistance. Ce composant servira ici "d'amortisseur" qui absorbera les rebonds (comme sur une voiture avec les cahots de la route). Le condensateur, initialement chargé, va se décharger lors de l'appui sur le bouton. S'il y a des rebonds, ils seront encaissés par le condensateur durant cette décharge. Il se passera le phénomène inverse (charge du condensateur) lors du relâchement du bouton

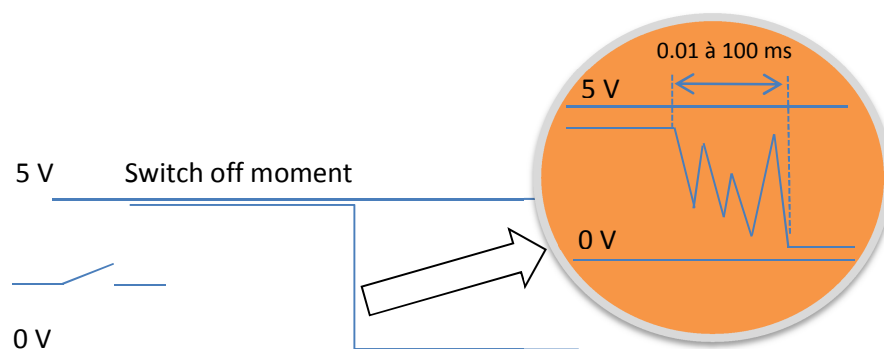


Figure (3.15): Effet de rebondissement de bouton poussoir

6.7 L'afficheur LCD

6.7.1 Rôle d'un afficheur LCD

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage de paramètres de fonctionnement. Grâce à la commande par un microcontrôleur ces afficheurs permettent de réaliser un affichage de messages aisés.

6.7.2 La mise en œuvre d'un afficheur LCD

Un afficheur LCD sera commandé par un microcontrôleur, il faut donc penser aux mises en œuvre :

- matérielle : connexion des broches du microcontrôleur à l'afficheur LCD
- logicielle : utilisation de sous programmes permettant de commander l'afficheur LCD (initialisation, effacement de l'afficheur, affichage d'un caractère, affichage d'une variable,...).

6.7.3 Schéma fonctionnel de l'afficheur LCD [10]

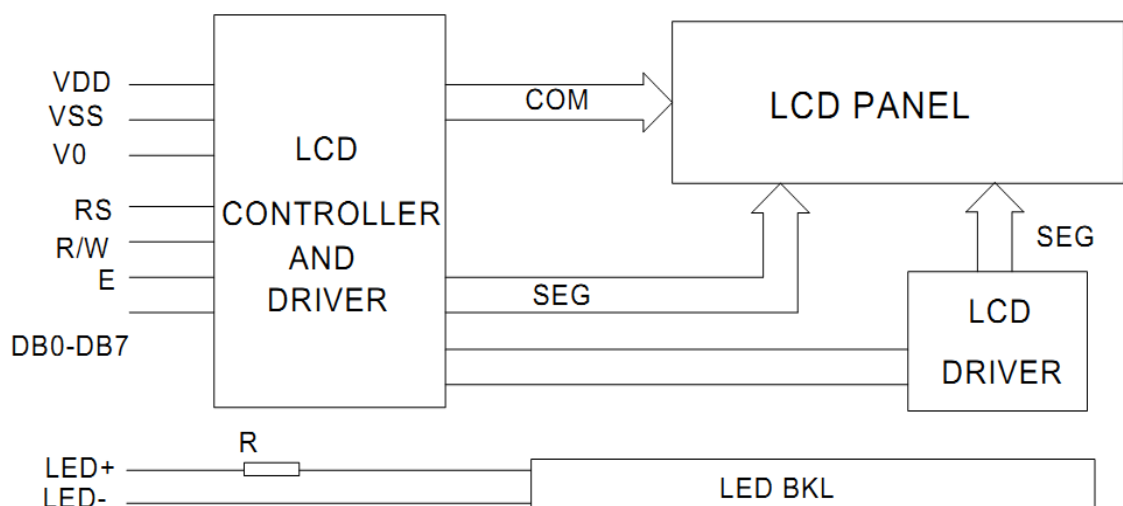


Figure (3.16) : Schéma fonctionnel de l'afficheur GDM1620A

6.7.4 Rôle des différentes broches de l'afficheur LCD [10]

VDD, VSS : alimentation de l'afficheur LCD. Un afficheur LCD s'alimente en 0V-5V.

VO : entrée permettant de régler le contraste de l'afficheur LCD. Il faut appliquer une tension continue réglable (entre 0V et 5V) à l'aide d'un potentiomètre.

RS : Register Select cette entrée permet d'indiquer à l'afficheur si l'on souhaite réaliser une commande (RS=0) par des instructions spécifiques ou écrire une donnée (envoi du code du caractère à afficher) sur le bus (RS=1).

R/W : entrée de lecture (R/W=1) et d'écriture (R/W=0). Lorsqu'on commande l'afficheur LCD il faut se placer en écriture.

E : entrée de validation (ENABLE), elle permet de valider les données sur un front descendant. Lorsque E=0 alors le bus de données est à l'état haute impédance.

D7...D0 : bus de données bidirectionnel, il permet de transférer les instructions ou les données à l'afficheur LCD.

LED+ LED - : différence de potentiel permettant de commander le rétro éclairage.

On a utilisé dans notre projet un afficheur LCD 2x16 caractères de référence GDM1602A

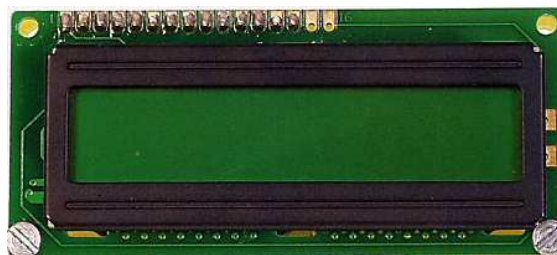


Figure (3.17) : Afficheur LCD 2x16 caractères GDM1602A

6.8 Le reset et l'oscillateur

Pour fonctionner le PIC réclame une configuration hardware minimale composée d'un circuit reset afin de réinitialiser à tout instant et un oscillateur pour générer l'horloge. Un oscillateur électronique est un montage électronique, dont la fonction est de produire

un signal périodique, de forme sinusoïdale, carrée, voire en dents de scie, ou quelconque. L'oscillateur peut avoir une fréquence fixe ou variable, dans notre cas elle est fixe.

Comme le montre la figure suivante

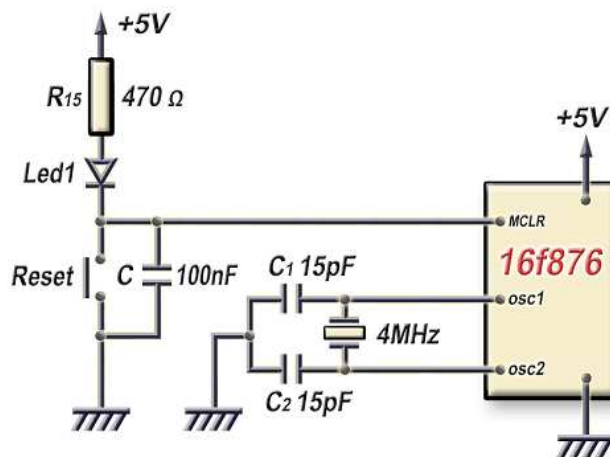


Figure (3.18) : Circuit de reset et de l'oscillateur

7. Conception logiciel

7.1 Organigramme

Avant de procéder à l'élaboration du programme contenu dans le PIC, il est préférable d'établir au préalable un algorithme qui définit le principe de fonctionnement.

L'organigramme suivant décrit sommairement l'organisation du programme. Il modélise l'effet d'un reset, représente la détection et le traitement des interruptions, et enfin décrit la boucle principale réalisant la correction, illustré par l'organigramme juste après.

7.2 Langage de programmation

Pour la programmation de PIC 16F876 on a choisi d'utiliser le langage C de fait qu'il est un langage évolué et très simple à utiliser, dans notre projet on a choisi le CCS « PIC C Compiler ».

Un langage évolué est un ensemble de fonction ou chaque une fonction est un programme en assembleur. Par exemple, pour programmer $\cos(x)$ en assembleur, en doit faire tout un programme compliqué, mais en langage C en fait juste appel à un fichier.h de la bibliothèque (`#include math.h`), ce fichier contient presque toutes les équations mathématique, en suite en va écrire juste `cos (x)`.

L'intérêt de simuler un programme est bien sûr pédagogique car comme nous le verrons, nous allons pouvoir faire visualiser le fonctionnement du PIC avec tous les périphériques de la carte de commande. De plus, nous allons pouvoir vérifier le fonctionnement du programme avant de le transférer vers la mémoire de microcontrôleur. Cela nous permettra d'économiser du temps et également d'augmenter la durée de vie de notre PIC.

7.3 Présentation générale de Logiciel de simulation utilisé PROTEUS

Le logiciel PROTEUS est composé de trois modules :

- L'éditeur de schéma ISIS
- Le simulateur LISA
- L'outil de conception de circuit imprimé ARES

ISIS produit d'une part, une liste d'équipotentiels qui peut être utilisée par le simulateur LISA et l'outil de conception de circuit imprimé ARES, d'autre part, une liste de matériel et des rapports de contrôle des règles électriques.

LISA est un ensemble de modules de simulation lié à ISIS. Le noyau de simulation PROSPICE est basé sur la version 3F5 du moteur SPICE publié par l'université de Berkeley.

ARES est un module de conception de circuit imprimé. Il permet le placement des composants en mode automatique, manuel ou semi-automatique et le routage des liaisons sur plusieurs couches en mode automatique manuel ou semi-automatique.

Pour le dessin du schéma électrique (figure 3.1) on a utilisé le logiciel ISIS et à travers de ce logiciel on a fait charger le fichier hexadécimal (.hex) dans le PIC pour faire des tests.

8. Tracé du circuit imprimé sous ARES

Le dessin de circuit imprimé se fait par des logiciels de conception assisté par ordinateur, notre conception a été réalisée en double face.

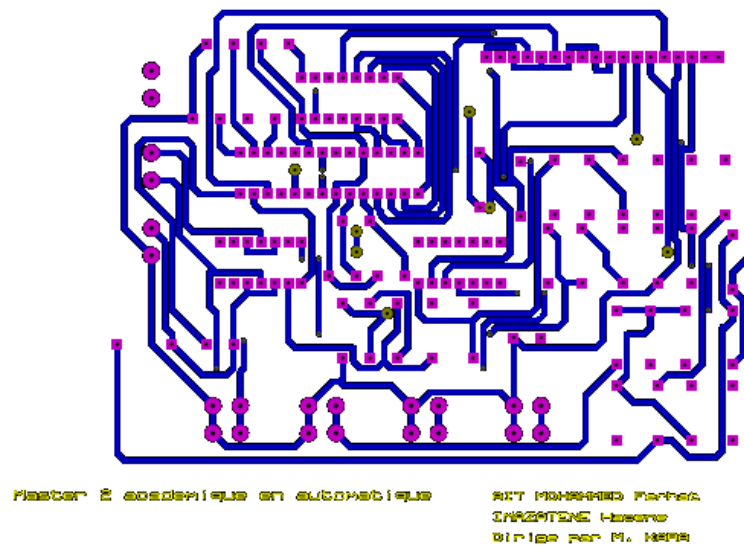


Figure (3.19) : Tracé de la première face du circuit imprimé

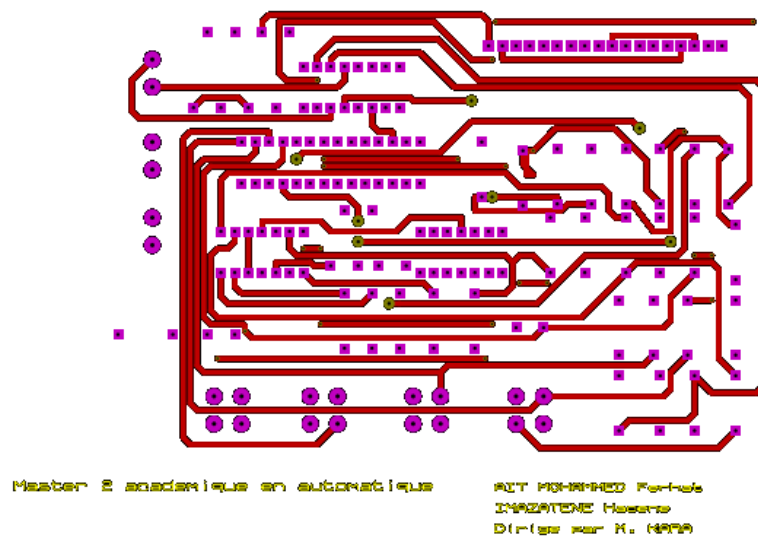


Figure (3.20) : Tracé de la deuxième face du circuit imprimé

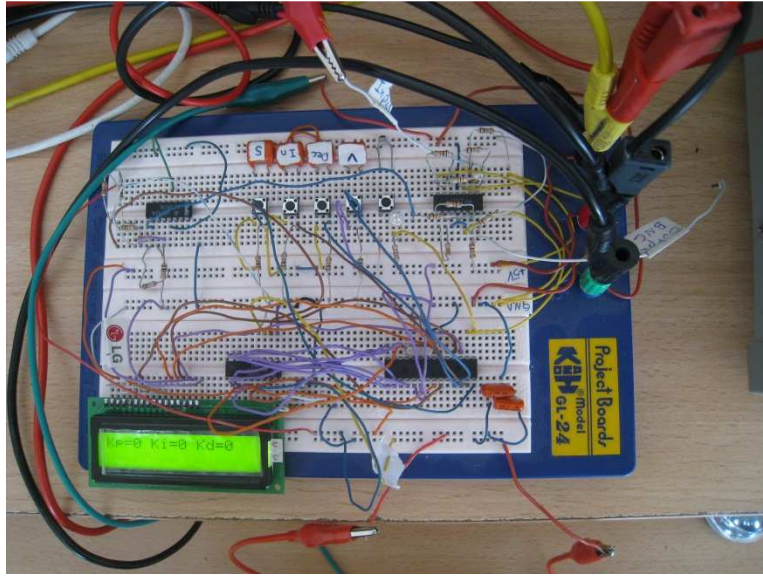


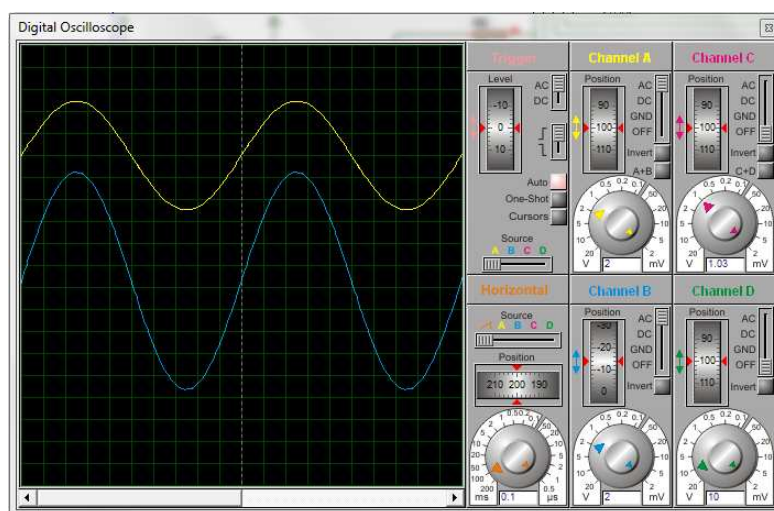
Figure (3.21) : Photographie de la réalisation sur lab d'essai

9. Circuit complet de la réalisation sous ISIS

10. Procédures de Test

10.1 Les tests souhaités

Le test du CNA se fait en même temps que le test du correcteur proportionnel, quand K_p vaut 1. Nous injectons en entrée un signal sinusoïdal et nous devons visualiser sur l'oscilloscope une sinusoïde similaire de même amplitude. Nous effectuons le même test avec cette fois-ci K_p égal à 2. Nous devons pouvoir visualiser une sinusoïde ayant une amplitude 2 fois plus grande que celle injectée en entrée.

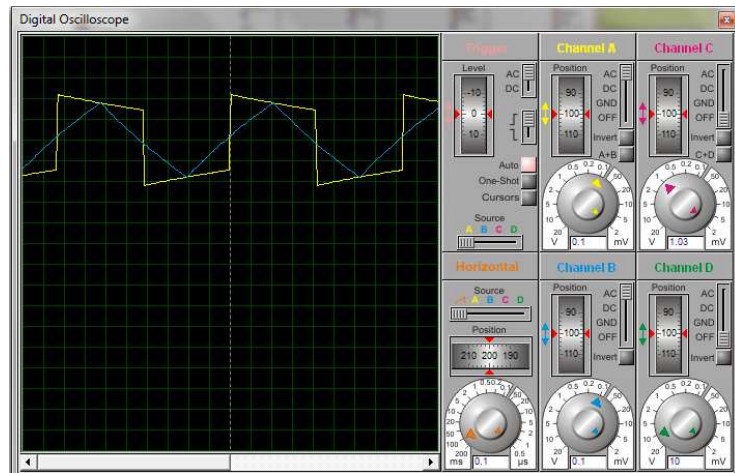


_____ : signal d'entrée du régulateur

_____ : signal de sortie du régulateur

Figure (3.22) : test de l'action proportionnelle avec $K_p=2$

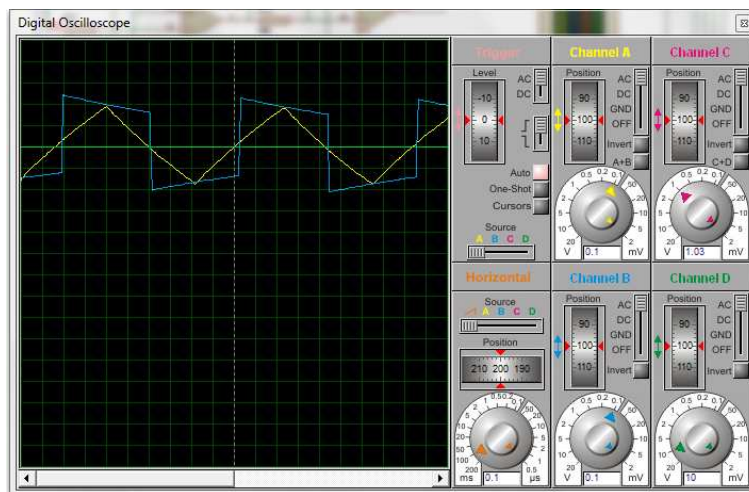
Pour tester l'action intégrale du correcteur on injecte un signal carré en entrée et nous devons visualiser un signal triangulaire sur l'oscilloscope.



_____ : signal d'entrée du régulateur _____ : signal de sortie du régulateur

Figure (3.23) : Test de l'action intégral

Pour tester l'action dérivée il faut injecter un signal triangulaire en entrée et nous devons récupérer en sortie un signal carré.



_____ : signal d'entrée du régulateur _____ : signal de sortie du régulateur

Figure (3.23) : Test de l'action dérivée

10.2 Test obtenu

Pour l'action proportionnelle avec K_p égale a 1 nous avons obtenu la figure suivante

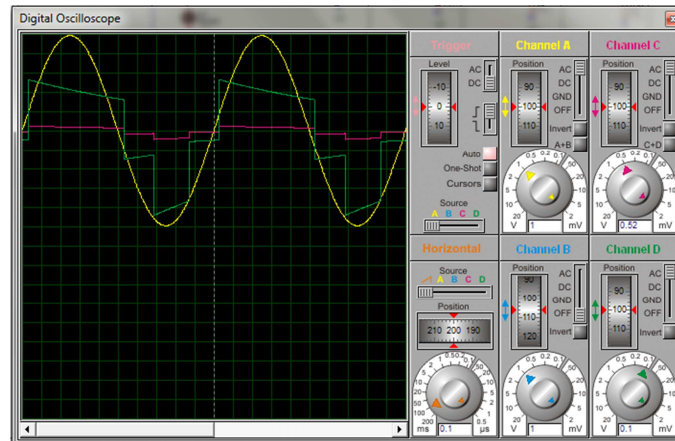


Figure (3.24) : Test de l'action proportionnelle avec K_p égale à 1

11. Conclusion

Les tests que nous avons effectués n'étaient pas concluants. Dans le sens où nous n'avons pas eu les réponses (les signaux de sortie) souhaités. Nous avons aussi, vérifié plusieurs fois la maquette et les différents composants un par un sans pour autant trouver l'origine du problème. Néanmoins les différents tests et vérifications que nous avons effectués nous ont permis de connaître dans le détail les différents composants que nous avons utilisés.

CONCLUSION
GENERALE

Conclusion générale

Nous avons traité dans ce mémoire de l'étude, de la conception et de la réalisation des régulateurs numériques PID, avance et retard de phase à base du PIC 16F876. Cette étude nous a permis de nous familiariser avec des méthodes de programmation des PICs et le monde de la réalisation électronique.

Nous avons conclu que l'utilisation de la commande numérique pour le pilotage des procédés, le régulateur PID en particulier, permet d'affiner la précision et d'augmenter considérablement la fiabilité des systèmes de commande.

Dans ce travail, nous avons touché à plusieurs domaines. Nous avons appris à programmer les PICs sous CCS, qui est un logiciel de programmation des PICs en utilisant le langage C et par la suite simuler le comportement du circuit réalisé sous ISIS. Enfin, nous avons utilisé ARES pour réaliser le circuit imprimé de la maquette à réaliser.

Malgré les tests non concluants, nous pensons avoir appris beaucoup de choses de cette expérience "le projet de fin d'étude"

Comme perspective, il est très intéressant d'introduire des algorithmes qui peuvent faire adapter les paramètres du correcteur vis-à-vis du système utilisé.

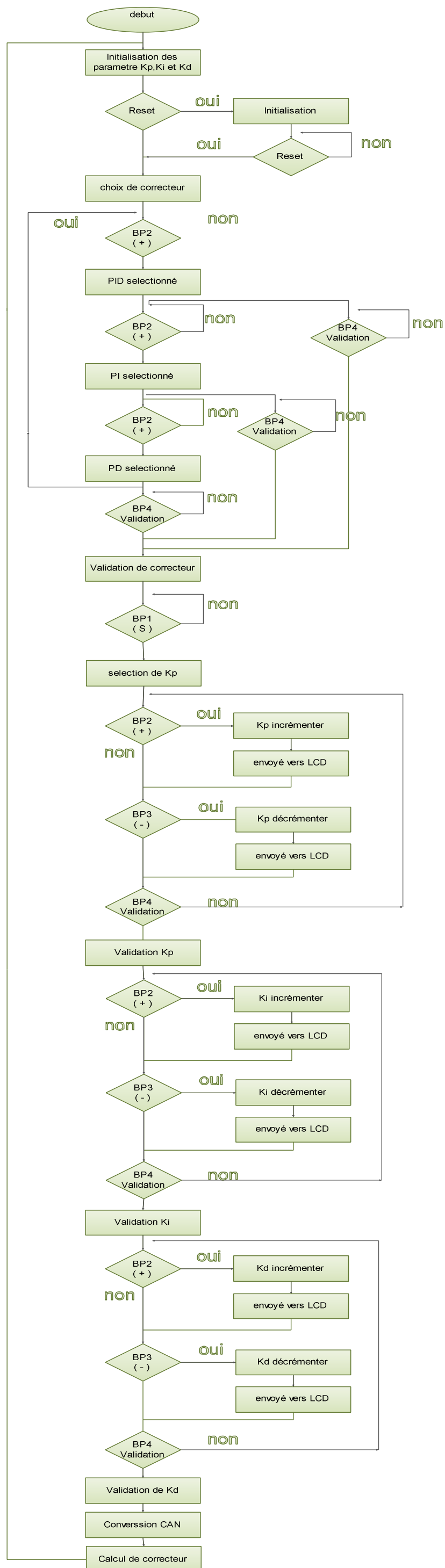
Enfin, nous espérons que ce travail sera utile pour les promotions à venir.

Bibliographie

- [1] P. Borne, G. Dauphin-Tanguy, J-P. Ricard, F. Rotella, I. Zambettakis " analyse et régulation des processus industriels " tome 2 régulation numérique, Edition TECHNIP [1993].
- [2] A. JUTARD M. BETEMEMPS " système asservis linéaires échantillonnées ", INSA de Lyon, [1998].
- [3] ROLAND LONGCHAMP " commande numérique des systèmes dynamiques ", INSA, [1995].
- [4] L.D LANDAU, "identification et commande des systèmes" PARIS Hermès.
- [5] Bernard BAYLE "système et asservissements à temps continu " [2008]
<http://eavr.u-strasbg.fr/bernard>.
- [6] The Michigan Open Control and Process Textbook :
<http://controls.engin.umich.edu/wiki/index.php/PIDTuningClassical>.
- [7] L'asservissement PID :
[http://ancrobot.free.fr/fiches/pdf/index\(2\).pdf](http://ancrobot.free.fr/fiches/pdf/index(2).pdf)
- [8] datasheet PIC 16F876.
- [9] datasheet DAC0802.
- [10] datasheet Afficheur LCD GDM1602A.
- [11] Pascal MAYEUX "apprendre la programmation des PIC par l'expérimentation et la simulation" PARIS Dunod, [2005].
- [12] E. DIEULESAINT, D.ROYER "automatique appliqué Tome2 système linéaire de commande à signaux échantillonnés" PARIS Masson [1990].
- [13] DE LARMINAT Philippe "commande des systèmes linéaire" PARIS Hermès [2000].

Bibliographie

- [14] GILLE J.Ch, DECAULNE Paul “théorie et calcul des asservissements linéaires” PARIS Dunod [1992].
- [15] Christian TAVERNIER, “programmation en C des PIC” PARIS Dunod [2005].



Organigramme

