

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS SETIF

THESE DE DOCTORAT D'ETAT

Présentée au Département d'Electronique
Faculté des Sciences de l'Ingénieur

Pour obtenir le titre de

Docteur d'Etat en Electronique

Par

KHOUNI Larbi

Thème

**AUTOMATIQUE AVANCEE POUR PROCESSUS
INDUSTRIELS
APPLICATION A LA COMMANDE DU pH**

Soutenue le 31/12/2008

devant la commission d'examen composée de

MOHAMMADI Tayeb
KHELLAF Abdelhafid
ZEGADI Ameer
CHICOUCHE Djamel
BOUTOUT Farid

Professeur à l'université de Sétif
Professeur à l'université de Sétif
Professeur à l'université de Sétif
Professeur à l'université de M'sila
M.C à l'université de M'sila

Président du jury
Directeur de thèse
Examineur
Examineur
Examineur

AUTOMATIQUE AVANCEE POUR PROCESSUS INDUSTRIELS APPLICATION A LA COMMANDE DU pH

Résumé

La capacité des réseaux neuronaux artificiels pour représenter des systèmes non linéaires en fait un puissant outil de modélisation et de commande des procédés industriels, ce qui est prouvé par le nombre important de travaux réalisés au cours de la dernière décennie. Toutefois, l'utilisation de réseaux neuronaux pour la modélisation et la commande dans l'industrie n'est pas encore très répandue. L'une des raisons est l'absence de procédures de conception qui existent pour la modélisation de systèmes linéaires, mais n'ont pas encore été mises au point pour la modélisation des procédés industriels à l'aide de réseaux de neurones.

Cette thèse présente en premier lieu l'étude et la réalisation d'un banc d'essai pour la mesure du pH des produits chimiques. L'étude souligne particulièrement la nature de la courbe de titration qui est fortement non linéaire. Par la suite certaines techniques qui peuvent aider à la conception d'un modèle de réseau de neurones pour procédés industriels sont traitées. Une méthode de sélection de la structure du modèle de réseau de neurones, sur la base de modèles de processus linéaires, est présentée. Certaines procédures qui peuvent aider à évaluer le signal d'excitation du processus et de valider des modèles de réseaux de neurones sont étudiées. L'importance de la conception de ces aspects d'un modèle de réseau de neurones et de la sélection d'un échantillon de temps de données approprié est démontrée.

Initialement, la modélisation de la simulation d'un processus CSTR pH est étudiée. Le processus du pH a des caractéristiques qui le rendent difficile à modéliser et à commander, telles que la non-linéarité, des mesures entachées de bruit et un temps de retard variable. La capacité du perceptron multi-couches pour prédire la réponse du processus de plusieurs étapes futures est évaluée en utilisant un modèle de simulation. Le modèle de réseau de neurones est employé dans un système de commande prédictive qui utilise l'algorithme Nelder-Mead d'optimisation.

La commande est appliquée à la simulation. La mise au point de la commande prédictive est décrite et ses performances sont comparées avec la commande linéaire. La commande pour la plage de fonctionnement du processus à l'aide de réseaux de neurones est nettement meilleure que la commande linéaire.

Advanced Process Control in Industry. A case study : Simulated pH process Control

Abstract

The artificial neural networks are a powerful tool for process modelling and control in non-linear systems and much work has been reported over the last decade. However, the use of neural networks for process modelling and control in industry is still rare. One of the reasons for this is the lack of established design procedures which exist for linear system modelling but have not yet been developed for neural network process modelling.

In this thesis, we first implement a microcontroller based system for an automatic control of pH and acidity degree of chemical products and consequently we came to the fact that the pH logarithmic non-linearity is severe.

Then we describe some techniques which can assist in the design of a neural network process model. A method of selecting the neural network model structure, based on linear process models, is presented and some procedures which can help to assess the process excitation signal and validate neural network models are illustrated.

Initially, the modelling of a simulated CSTR pH process is studied. The pH process has features which make it difficult to model and control, such as non-linearity, noisy process measurements and a non-constant time delay. The ability of the multi-layered perceptron neural networks to predict the process response multiple steps ahead is assessed using a simulated model. The neural network model is then employed in a predictive control scheme which uses a sequential Nelder-Mead optimisation algorithm. Initially, the controller is studied in simulation. The tuning of the predictive controller is described and its performance is compared with linear control. The best results obtained from a linear controller are comparable to the neural network controller. The network controller is significantly superior to linear control.

Dédicace

Mes remerciements les plus chaleureux sont adressés au Professeur A. Khellaf pour m'avoir encadré, supervisé le projet et m'avoir tout le temps encouragé. Je suis très touché par l'aide précieuse apportée par le collègue Lyes Saad Saoud et l'aide non négligeable apportée par H. Khouni et D. Mokadem.

Enfin, je remercie ma femme et mes enfants de manifester une patience exemplaire durant la période où j'ai été totalement absorbé par le projet.

Liste des Abréviations et Symboles

α	Coefficient de momentum, constante d'un filtre passe-bas
Δt	Intervalle d'échantillon
Δu_{max}	Intervalle de la variable manipulée
$\varepsilon(t)$	Erreur de prédiction
$\varphi_{vr}(\tau)$	Fonction d'intercorrélation pour $v(t)$ et $r(t)$
η	Taux du coefficient d'apprentissage de rétro-propagation
λ	Facteur d'oubli
θ	Vecteur paramètre
τ	Constante de temps
τ_D	Constante de temps
τ_i	Constante de temps
σ_x	Déviatiion standard de x
ω_{co}	Fréquence de coupure
AFPE	Erreur de prediction finale Akaike
ARX	AutoRegressive with eXogenous inputs
BP	Rétro-propagation
C	Horizon de commande
CAN	Convertisseur Analogique Numérique
C_A, C_B	Concentration acid and base
CSTR	Continuous Stirred Tank Reactor
$D(t)$	Matrice Diagonale
EEPROM	Mémoire programmable et effaçable électriquement
e' et e''	Erreur du processus et du modèle de référence
$e(t)$	Séquence discrète de bruit blanc, erreur de sortie du processus
f'	Dérivée première de f
$f(\cdot)$	Fonction non linéaire
F_A, F_B	Flux acide et base

G	Fonction de Transfert
E/S	Entrée /Sortie
IMC	Commande à Modèle Interne
J	Fonction de
J_p	Coût du $p^{\text{ième}}$ vecteur d'apprentissage
k	Temps mort du process
K_A, K_B et K_W	Constantes de dissociation de l'acide, la base et l'eau
K_C	Gain proportionnel
LCD	Afficheur à cristaux liquides
LF	Fonction de pertes
LMPC	Commande prédictive à modèle linéaire
LS	Moindres carrés
MBC	Commande basée sur modèle
MIMO	Multi Input Multi Output
MLP	Perceptron à multi-couches
MPC	Model Predictive Control
MSCE	Mean Square Control Effort
MSE	Mean Square Error
N	Nombre de données
n	Ordre du modèle
NARMAX	Non-linear AutoRegressive Moving Average with eXogenous inputs
NARX	Non-linear AutoRegressive with eXogenous inputs
NPI	Index de performance normalisé
n_u, n_y et n_e	Nombre d'entrées, de sorties et d'erreurs retardées
$P(t)$	Matrice de covariance
x_i	
P1, P2 et P	Horizon de prédiction, début, fin et taille
PI	Proportionnel plus Intégral, Index de Performance
PID	Proportionnel plus Integral plus Derivée
PRBS	Pseudo-random binary sequence
RAM	Mémoire à lecture et écriture
$r(t)$ et $r'(t)$	Consigne et Consigne Modifiée

RAS	Random Amplitude Signal
RLS	Moindres carrés Recursifs
RNA	Réseau de Neurones Artificiel
ROM	Mémoire à lecture seule
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
SP	Consigne
V	Volume du liquide dans le CSTR
\mathbf{v}	Vecteur de données E/S
W	Nombre total des poids du réseau
\mathbf{w}	Vecteur de poids
\mathbf{x}	Vecteur d'entrée
$y_p(t)$	Sortie estimée du processus
$y(t)$	Sortie du processus

Sommaire

INTRODUCTION	1
1. CONCEPTION ET REALISATION D'UN BANC D'ESSAI DE TITRATION A BASE DE MICROCONTROLEUR	3
1.1 NATURE ET FORME DU SIGNAL pH	3
1.1.1 Description d'une électrode	3
1.1.2 Temps de réponse	3
1.2. TRAITEMENT ET MISE EN FORME DU SIGNAL pH	4
1.3 PRESENTATION DU PIC16F876 (A)	5
1.3.1. Introduction	5
1.3.2 Caractéristiques générales du PIC16F876 (A)	6
1.3.3. Organisation de la RAM	7
1.3.4 Conversion Numérique Analogique	7
1.3.4.1 Temps d'acquisition	8
1.3.4.2 Algorithme de conversion	10
1.4 LES AFFICHEURS A CRISTEAUX LIQUIDES	10
1.4.1 Description	10
1.4.2 Principe de fonctionnement	11
1.4.2.1 Organigrammes d'écriture d'une commande ou d'une donnée	12
1.4.2.2 Organigrammes pour lire une donnée	13
1.5 LE pH METRE	14
1.5.1. Etalonnage	14
1.5.2. Influence de la température	16
1.5.3 Réalisation d'un pH mètre	16
1.6 LES COURBES DE TITRAGE	18
1.6.1 Titrage d'un acide fort par une base forte	18
1.6.2 Les tampons	19
1.6.3 Titrage d'un acide faible par une base forte	20
1.6.4 Les polyacides :	22
1.6.4.1 Titrage d'un acide faible monoprotone par une base forte	22
1.6.4.2 Titrage d'un acide faible triprotone par une base forte	26
1.7 DESCRIPTION DU BANC D'ESSAI	29
1.8 Détermination du point d'équivalence	31

1.8.1. Méthode manuelle	31
1.8.2 Méthode numérique	32
1.8.3 METHODE DE GRAN	33
1.8.3.1 Dosage d'un acide fort par une base forte	33
1.8.3.2. Cas de dosage d'un acide faible par une base forte	33
1.9 MESURES EXPERIMENTALES	34
1.9.1 Analyse de l'acide acétique	34
1.9.2 Dosage de l'acide éthanoïque contenu dans un vinaigre	35
1.9.3 Dosage de l'acide lactique contenu dans le lait :	38
2. RESEAUX DE NEURONES ARTIFICIELS POUR LA MODELISATION DES PROCESSUS CHIMIQUES	41
2.1 INTRODUCTION	41
2.1.1 Historique	41
2.2 Modélisation de processus et commande	42
2.2.1 Techniques des systèmes linéaires	45
2.2.2 Techniques des systèmes non linéaires	46
2.2.3 Motivation pour une approche de commande non linéaire en utilisant les réseaux de neurones	47
2.3 Propriétés de la commande du pH	48
2.3.1 Une approche de commande du pH par RNA	50
2.4 RESEAUX DE NEURONES ARTIFICIELS	51
2.4.1 RNA non bouclés	51
2.4.2 Le perceptron multicouches (MLP)	52
2.4.2.1 Le Perceptron	53
2.4.2.2 Apprentissage d'un MLP RNA	55
2.4.2.3 RNA MLP et rétro propagation	58
2.5 MODELISATION DE PROCESSUS UTILISANT LES RNA NON BOUCLES	59
2.5.1 Le modèle de structure NARX	60
2.5.2 Implémentation d'un modèle de processus RNA de structure NARX	61

2.5.3 Configuration d'un RNA non bouclé dans une structure NARX	61
3. MODELISATION DU pH D'UN REACTEUR CSTR A L'AIDE DE RESEAUX DE NEURONES	64
3.1 INTRODUCTION	64
3.2 DESCRIPTION DU PROCESSUS	64
3.2.1 Le modèle mathématique du processus	65
3.3 APPRENTISSAGE ET VALIDATION DES MODELES RNA	67
3.3.1 Procédure expérimentale	68
3.3.2 Excitation du processus à l'aide d'un signal à amplitude aléatoire	69
3.4 UN MODELE INITIAL RNA	69
3.5 ETUDE DU MODELE LINEAIRE LOCAL	61
3.5.1 Structure du modèle ARX	72
3.5.2 Echantillons du temps	74
3.6 ETUDE DU MODELE RNA NON LINEAIRE	76
3.6.1 Sélection de l'échantillon du temps	81
3.6.2 Signaux d'excitation	83
4. SIMULATION DE COMMANDE DU PROCESSUS pH	89
4.1 Examen des stratégies de commande RNA	89
4.1.1 Les modèles Inverses	89
4.1.2 Commande à modèle interne	91
4.1.3 La commande à modèle de référence	92
4.1.4 Commande prédictive	93
4.2 La Commande Linéaire de simulation du processus	94
4.2.1 Commande PID	95
4.3 La commande prédictive RNA de simulation du processus pH	102
4.3.1 La structure de commande prédictive à RNA	102
4.3.2 Optimisation de la fonction de coût	105
4.3.3 Sommaire de l'algorithme MPC	106
4.3.4 L'horizon de commande	107
4.3.5 Construction du modèle CSTR	107
4.3.5.1 Conception du bloc calculant ζ et ξ	107
4.3.5.2 Conception du bloc calculant $[H^+]$	108
4.3.5.2 Conception du bloc calculant le pH	109

4.4 Simulation MPC	111
4.4.1 Horizon de Commande	113
Conclusion	119
Bibliographie	122

INTRODUCTION

Il y a de nombreuses raisons pour lesquelles les processus chimiques devraient être effectivement et efficacement commandés. Souvent, le désir de produire au maximum en utilisant moins de matières premières et d'énergie est présenté comme la première motivation pour la commande des processus. Le besoin de la commande des processus chimiques est fondamental dans la mesure où le succès de fabrication des produits dépend étroitement du maintien des conditions du processus dans certaines limites. En effet, pour certains processus chimiques, comme la production de toxines, la nature des produits et sous-produits demande une sécurité d'exploitation, et dans ce cas la commande du processus dans des limites sûres est évidemment essentielle. La nécessité pour la commande du pH, dans ses différentes formes, englobe toutes ces questions. Dans la nourriture et les industries biotechnologiques la commande du pH est nécessaire pour assurer le rendement et la qualité des produits (SHINSKEY, 1973). Pour le traitement des eaux usées, le pH des effluents doit être réglementé pour protéger les milieux aquatiques et le bien-être des êtres humains et de se conformer aux limites imposées par la législation. Enfin, il existe des différences significatives dans le coût de la conception et la construction d'usines de neutralisation. Ainsi, la conception d'un système de commande du pH peut avoir des incidences sur tous ces domaines.

Le chapitre 1 est consacré à l'étude et à la réalisation d'un banc d'essai de titration. On prévoit de doter le banc d'essai d'un pH mètre et d'une pompe doseuse qui sert à titrer les produits chimiques à analyser. Tout le système sera piloté par un microcontrôleur PIC 16F87X. Cette partie nous fera découvrir le comportement du processus pH à travers la courbe de titration.

Le chapitre 2 décrit le comportement du processus pH qui est considéré comme un phénomène fortement non linéaire. Ce qui nous conduit à étudier les topologies et les propriétés d'approximation des réseaux de neurones en vu de leur éventuelle utilisation pour modéliser le processus pH. Nous présentons dans ce chapitre la structure flexible du modèle RNA pour modéliser un processus dynamique et les méthodes pour sélectionner et valider une structure parcimonieuse.

Les techniques introduites dans le chapitre 2 sont appliquées dans le chapitre 3 à la modélisation du processus pH CSTR. Une méthode de sélection et de validation de la structure du modèle RNA est présentée.

Le chapitre 4 décrit les différentes commandes basées sur modèles RNA. Les simulations de commande linéaire et de commande MPC du processus pH CSTR sont effectuées en utilisant MATLAB et les performances des deux types de commande sont comparées.

1. CONCEPTION ET REALISATION D'UN BANC D'ESSAI DE TITRATION A BASE DE MICROCONTROLEUR

1.1 NATURE ET FORME DU SIGNAL pH

1.1.1 Description d'une électrode

De nombreuses constructions sont possibles pour réaliser une électrode indicatrice de pH. L'électrode commerciale SOLEA-TACUSSEL est celle que nous avons utilisée comme électrode de mesure du pH. C'est une électrode sélective à membrane de verre destinée à la détermination de l'activité des ions H^+ (figure 1.1)(Gulaian,1990).

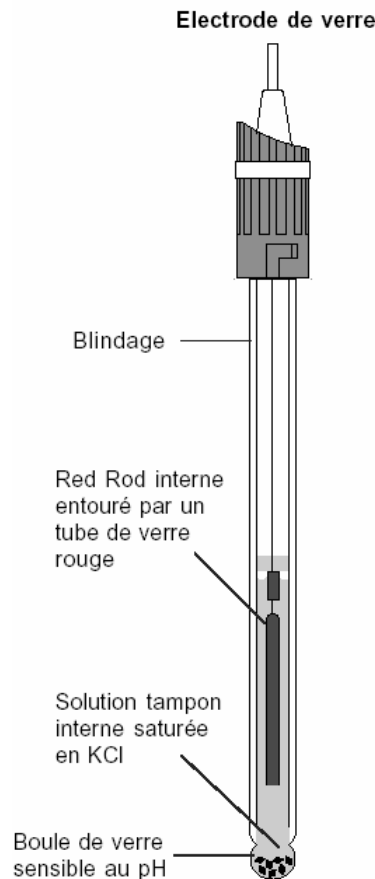


Figure 1.1 : Structure globale de l'électrode TACUSSEL

1.1.2 Temps de réponse

Le temps de réponse de l'électrode correspond au délai de stabilisation du potentiel dû à l'activité des ions. Ainsi pour un fonctionnement correcte du banc de mesure il faut que la

réponse du capteur soit nettement moins rapide que le temps correspondant à la variation de l'activité électrochimique.

La figure (1.2) ci-dessous donne une idée sur le temps de réponse d'une électrode de ce type.

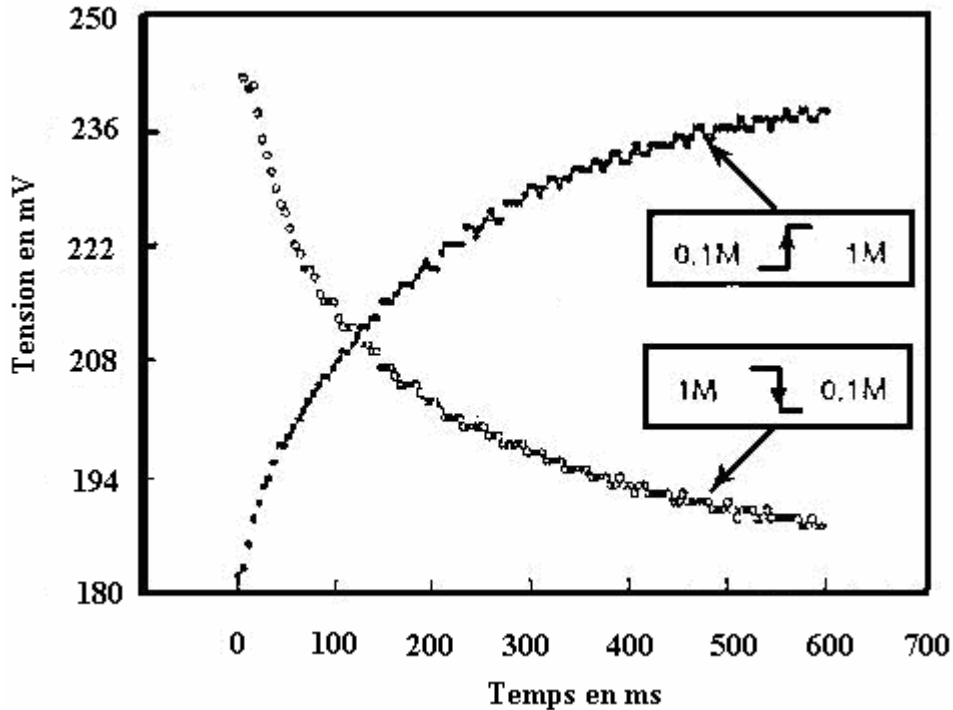


Figure 1.2 : Réponse de l'électrode Tacussel (MOKADEM,1998)

1.2. TRAITEMENT ET MISE EN FORME DU SIGNAL pH

Le signal issu du capteur de pH attaque via un connecteur BNC l'entrée non inverseuse d'un premier amplificateur opérationnel type JFET dont l'impédance d'entrée est considérée comme infinie dans ces conditions de travail. Après cette première amplification, l'amplitude du signal se trouve comprise entre -2.5 V et 2.5 V .

Le signal parasite de 50 Hz qui se superpose au signal utile du capteur est éliminé d'une manière efficace par la capacité de filtrage C1 suivie par l'action d'une cellule de filtre passe-bas à base d'amplificateur opérationnel (figure 1.3).

Au terme de cette étape, qui est fondamentale pour le traitement numérique et la fiabilité des résultats, le signal produit par le capteur est « propre » et sa valeur est proportionnelle au pH.

Vu que le convertisseur CAN du PIC ne peut supporter qu'une tension comprise entre 0 et 5 V, il faudrait que les tensions de sortie de la cellule de filtrage (-2.5V à $+2.5\text{V}$) soient

compatibles avec les entrées du CAN. Le dernier étage de la carte de mise en forme du signal est réalisé pour ce but (figure 1.3). (MILSANT,1981)

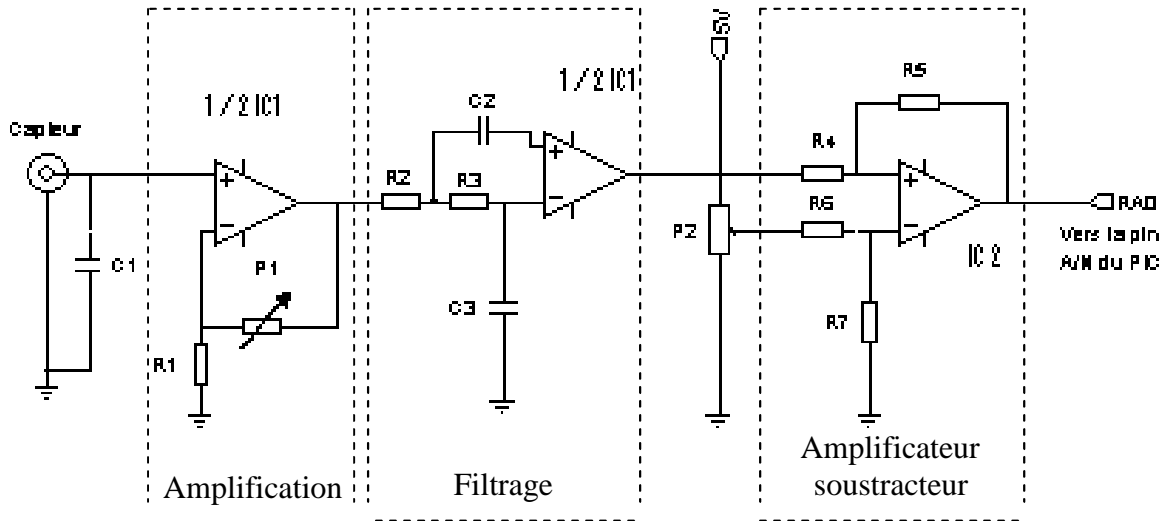


Figure 1.3 : carte d'acquisition et de traitement du signal pH

1.3 PRESENTATION DU PIC16F876 (A) : (Tavernier,2000)

1.3.1. Introduction

Les microcontrôleurs PIC (Peripheral Interface Controller) de MICROCHIP se distinguent des microcontrôleurs classiques (8051 ou 68HC11) par les deux aspects suivants :

a. Ce sont des microprocesseurs de structure Harvard :

C'est à dire que les instructions et les données sont stockées dans deux mémoires différentes. Les instructions ont une longueur allant de 12 à 14 bits suivants les modèles et sont écrites dans une ROM interne non accessible de l'extérieur. Il en résulte que toutes les instructions peuvent être codées sur un seul mot d'où une exécution rapide, mais en contre partie aucun fonctionnement avec la ROM extérieure n'est possible.

b. Ce sont des microprocesseurs à architecture RISC (Reduced Instructions Set Computers)

C'est à dire leurs instructions sont en nombre réduit, mais efficace, de plus ils possèdent une architecture push pull permettant l'exécution d'une instruction et simultanément le chargement de l'instruction suivante pendant un cycle d'horloge.

1.3.2 Caractéristiques générales du PIC16F876 (A)

- Horloge maximum de 20 MHz (5 millions d'instructions par seconde).
- 8 K mots de programme.
- 368 octets de RAM.
- 256 octets de mémoire EEPROM flash pour les données manipulables par le programmeur.
- 22 entrées/sorties.
- 3 timers dont deux de 8 bits et un de 2x8 bits
- Un convertisseur analogique numérique de 5 canaux.

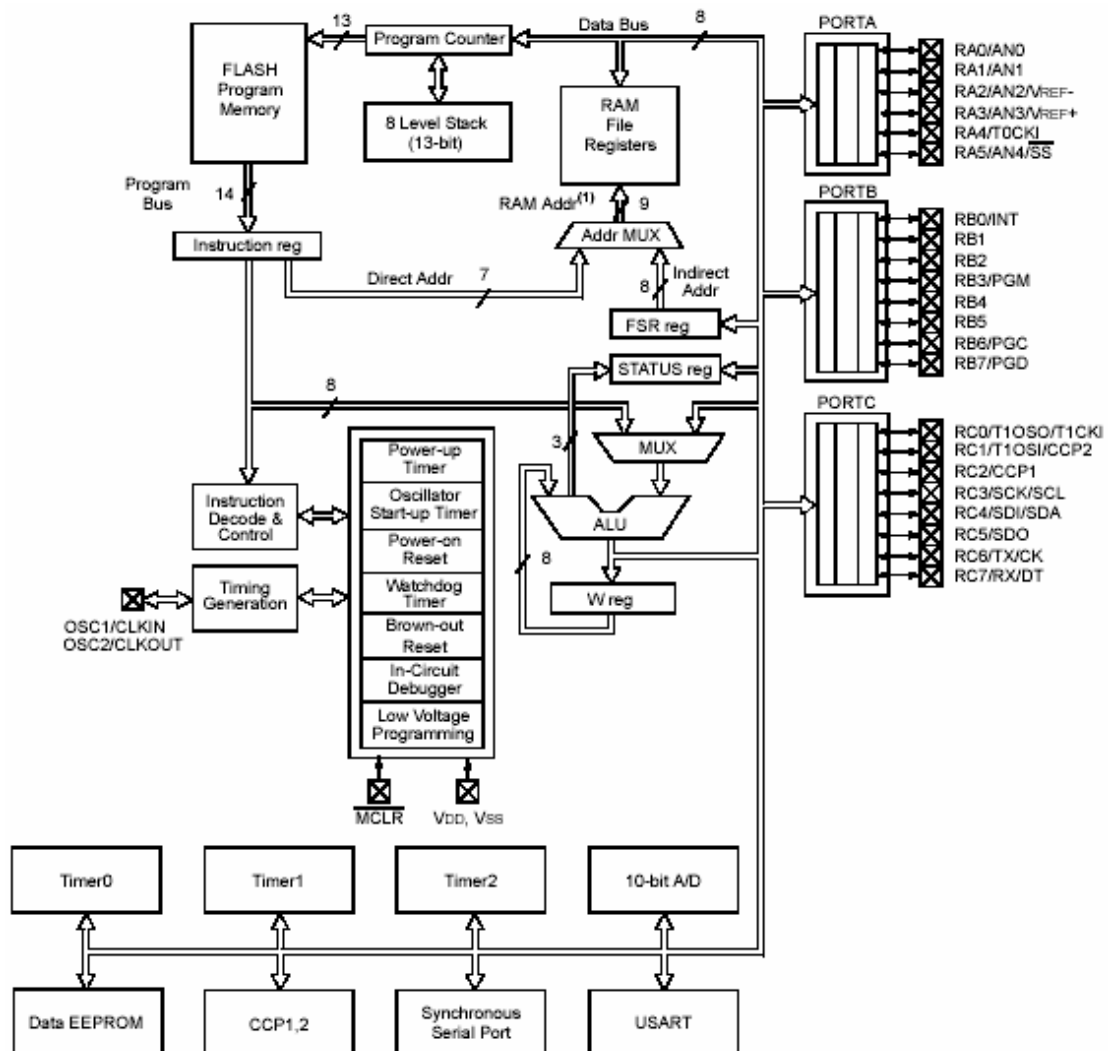


Figure 1.4 : Architecture interne du PIC16F876 (A)

Pour pouvoir convertir une grandeur, nous devons connaître la valeur minimale qu'elle peut prendre, ainsi que sa valeur maximale. Les pics considèrent par défaut que la valeur minimale correspond à 0, tandis que la valeur maximale correspond à la tension positive d'alimentation V_{cc} . Nous verrons cependant qu'il est possible d'utiliser d'autres valeurs.

La séquence de conversion est la suivante :

- Le pic connecte la broche sur laquelle se trouve la tension à numériser à un condensateur interne qui va se charger via une résistance interne jusqu'à la tension appliquée.
- La broche est déconnectée du condensateur et ce dernier est connecté sur le convertisseur analogique/numérique interne.
- Le pic procède à la conversion.

1.3.4.1 Temps d'acquisition

$$T_{acq} = T_{amp} + T_c + T_{coff} \quad (1.1)$$

T_{acq} = temps d'acquisition total

T_{amp} = temps de réaction des circuits

T_c = temps de charge du condensateur

T_{coff} = temps qui dépend du coefficient de température.

Le temps de réaction T_{amp} est typiquement de $2\mu s$.

Le coefficient de température, qui est typiquement de $0,05 \mu s$ par $^{\circ}C$, n'est nécessaire que pour les températures supérieures à $25^{\circ}C$. Pour les températures inférieures il n'entre pas en compte. Il s'agit bien entendu de la température du PIC et non de la température ambiante.

Donc, ce temps T_{coff} sera au minimum de 0 (à moins de $25^{\circ}C$) et au maximum de $(50-25)*0.05$, soit $1,25 \mu s$. La température du pic ne pouvant pas, en effet, excéder $50^{\circ}C$.

$$0 \leq T_{coff} \leq 1,25\mu s$$

T_c dépend de la résistance placée en série avec le condensateur. En fait, il y a deux résistances : celle de la source du signal et la résistance d'entrée du PIC. Il est recommandé que la résistance de la source reste inférieure à $10K\Omega$. La résistance interne totale (composée de 2 résistances internes) varie de $6K K\Omega$ à $6V$ pour atteindre $12 K\Omega$ à $3V$, en passant par $8K K\Omega$ à $5V$.

Sachant que le condensateur interne à une valeur de $120pF$ pour les versions actuelles de PIC (16F876), la formule du temps de charge du condensateur est :

$$T_c = C * (R_{interne} + R_{source}) * \ln(1/2047) \quad (1.2)$$

Comme « C » est fixe, la formule devient donc :

$$T_c = 0,914895 * 10^{-9} * (R_{interne} + R_{source}) \quad (1.3)$$

Si on se place dans le cas le plus défavorable (tension de 3V et résistance source de 10 KΩ), le temps de charge est de :

$$T_c = 0,914895 * 10^{-9} * (10 * 10^3 + 12 * 10^3) = 20,12 \mu s$$

Dans le cas le plus favorable (tension de 6V, et résistance source négligeable) :

$$T_c = 0,914895 * 10^{-9} * (10 * 10^3 + 6 * 10^3) = 5,48 \mu s$$

Si maintenant, nous prenons un cas typique, à savoir une tension d'alimentation de 5V et une résistance de source de 10 KΩ, nous aurons :

$$T_c = 0,914895 * 10^{-9} * (10 * 10^3 + 8 * 10^3) = 16,46 \mu s.$$

Nous allons maintenant calculer les temps : minimum, maximum et typique du temps total d'acquisition Tacq.

Le cas le plus défavorable est : une température de 50°C et un Tc maximal, ce qui donne :

$$T_{acq} = T_{amp} + T_{ac} + T_{coff}$$

$$T_{acq \text{ maximum}} = 2\mu s + 20,12\mu s + 1,25\mu s = 23,37 \mu s$$

Le cas le plus favorable : une température inférieure ou égale à 25°C et un Tc minimal :

$$T_{acq \text{ minimum}} = 2\mu s + 5,48\mu s = 7,48 \mu s.$$

Sous 5V, nous aurons dans le pire des cas :

$$T_{acq \text{ sous 5V}} = 2\mu s + 16,46\mu s + 1,25\mu s = 19,71\mu s.$$

Donc, un Tacq de 20μs est une valeur convenable pour notre PIC alimenté sous 5V.

D'autre part, il faut savoir que la conversion d'un bit, nécessite un temps qu'on va nommer Tad. Ce temps est obtenu par division de l'horloge principale. Le facteur de division peut prendre les valeurs 2, 8 ou 32. Le temps de conversion Tad ne peut descendre en dessous de 1,6μs pour les versions classiques de 16F87x et en dessous de 6μs pour les versions LC.

$T_{ad} = T_{osc} / \text{facteur de division}$.

Avec une fréquence maximale de 20Mz, et un facteur de 32, le temps est de 1,6 μs

Il faut à présent préciser que le PIC nécessite un temps Tad avant le démarrage effectif de la conversion et un temps supplémentaire Tad à la fin de la conversion. Le temps total de conversion est donc de :

- Tad : avant le début de conversion (le temps de connexion du condensateur est inclus)
- 10 * Tad pour un résultat de conversion de 10 bits
- Tad supplémentaire pour la fin de la conversion de b₀.

Temps entre deux échantillonnages :

$$T = T_{acq} + 12 T_{ad} + 2 T_{ad} = T_{acq} + 14 T_{ad}$$

Donc :

$$T = 19,71\mu s + 14 * 1,6\mu s = 42,11 \mu s.$$

Ce qui correspond à une fréquence d'échantillonnage de :

$$F = 1/T = 1 / 42,11 * 10^{-6} = 23747 \text{ Hz.}$$

1.3.4.2 Algorithme de conversion

Dans notre cas le PIC cadencé à 20Mhz, sous une tension d'alimentation de 5V, effectue l'échantillonnage du signal selon l'algorithme suivant :

- 1) Configurer ADCON1 en fonction des broches utilisées en mode analogique, ainsi que le registre TRISA.
- 2) Positionner le diviseur 32 sur ADCON0 (B'10000000').
- 3) Choisir le canal en cours de digitalisation sur ADCON0 et lancer le convertisseur (B'10000001').
- 4) Attendre le temps ($T_{acq} + 2T_{ad}$), soit $19,7\mu\text{s} + 3,2\mu\text{s} = 22,9\mu\text{s}$.
- 5) Démarrer la conversion en positionnant le bit GO du registre ADCON0.
- 6) Attendre la fin de la conversion.
- 7) Lire les registres ADRESH et si nécessaire ADRESL.
- 8) Reprendre l'opération à l'étape 4.

1.4 LES AFFICHEURS A CRISTEAUX LIQUIDES

1.4.1 Description

Les afficheurs à cristaux liquides sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils sont relativement bon marché et s'utilisent avec beaucoup de facilité.

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage consomme beaucoup de courant (jusqu'à 250 mA max.).

1.4.2 Principe de fonctionnement

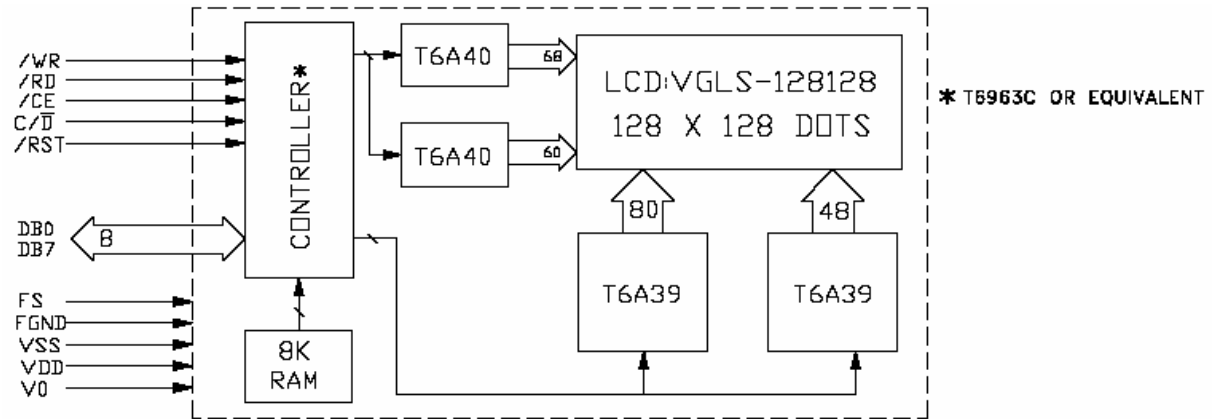


Figure 1.6 : Schéma fonctionnel d'un afficheur LCD

Comme le montre le schéma fonctionnel, l'affichage comporte d'autres composants que l'afficheur à cristaux liquides (LCD) seul. Un circuit intégré de commande spécialisé, le LCD-contrôleur, est chargé de la gestion du module. Le "contrôleur" remplit une double fonction: il commande l'affichage et se charge de la communication avec l'extérieur.

Pour envoyer une instruction à l'afficheur, il faut vérifier l'état de l'afficheur, régler les bits /R au niveau haut, /W au niveau bas, spécifier le bit C/D (si c'est une commande ou une donnée), mettre le byte de la commande ou la donnée en question sur les bits D7~D0, et enfin valider, en mettant d'abord le bit CE à 1 puis 0.

1.4.2.1 Organigrammes d'écriture d'une commande ou d'une donnée

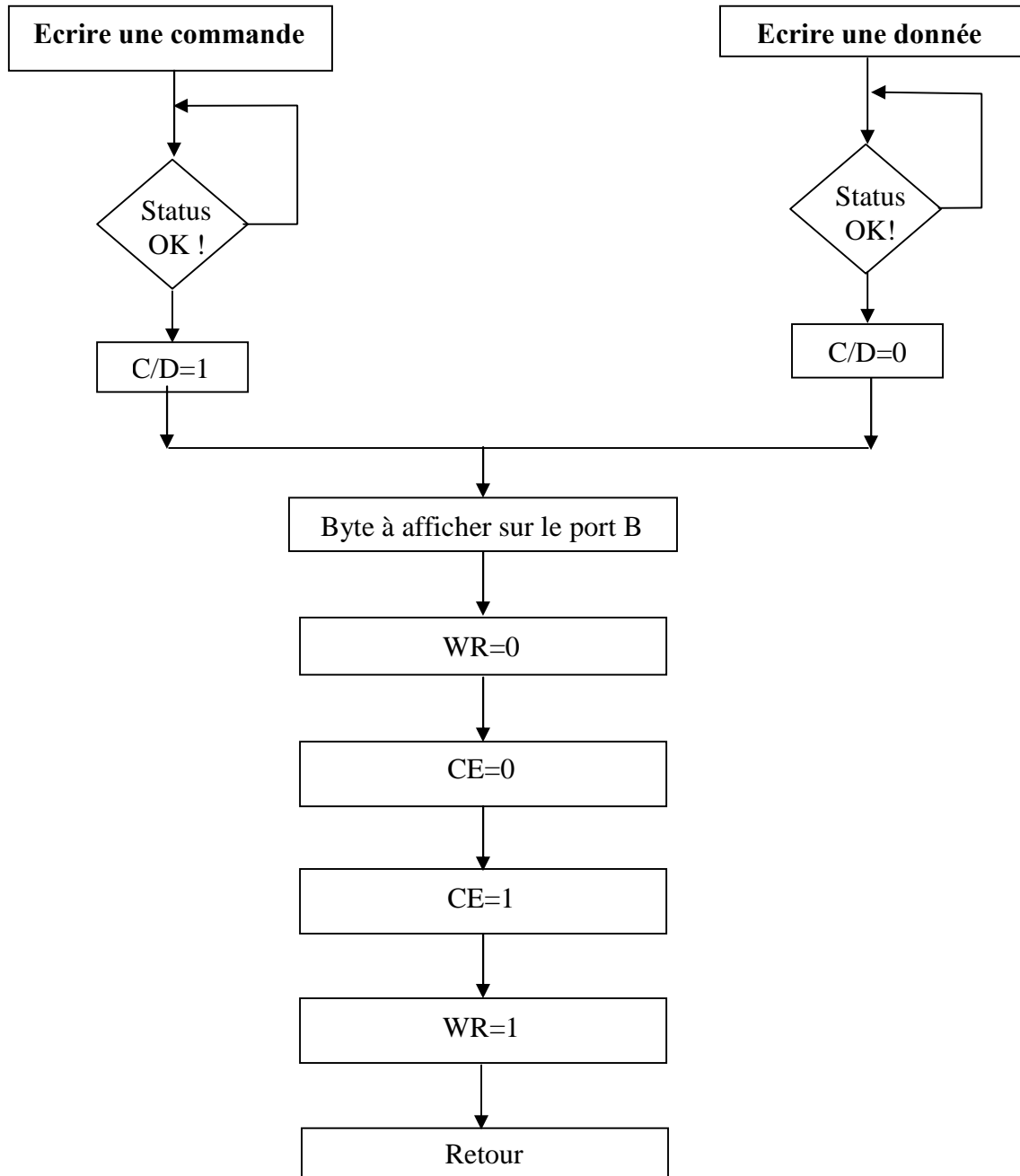


Figure 1.7 Organigrammes d'écriture d'une commande ou d'une donnée

1.4.2.2 Organigrammes pour lire une donnée

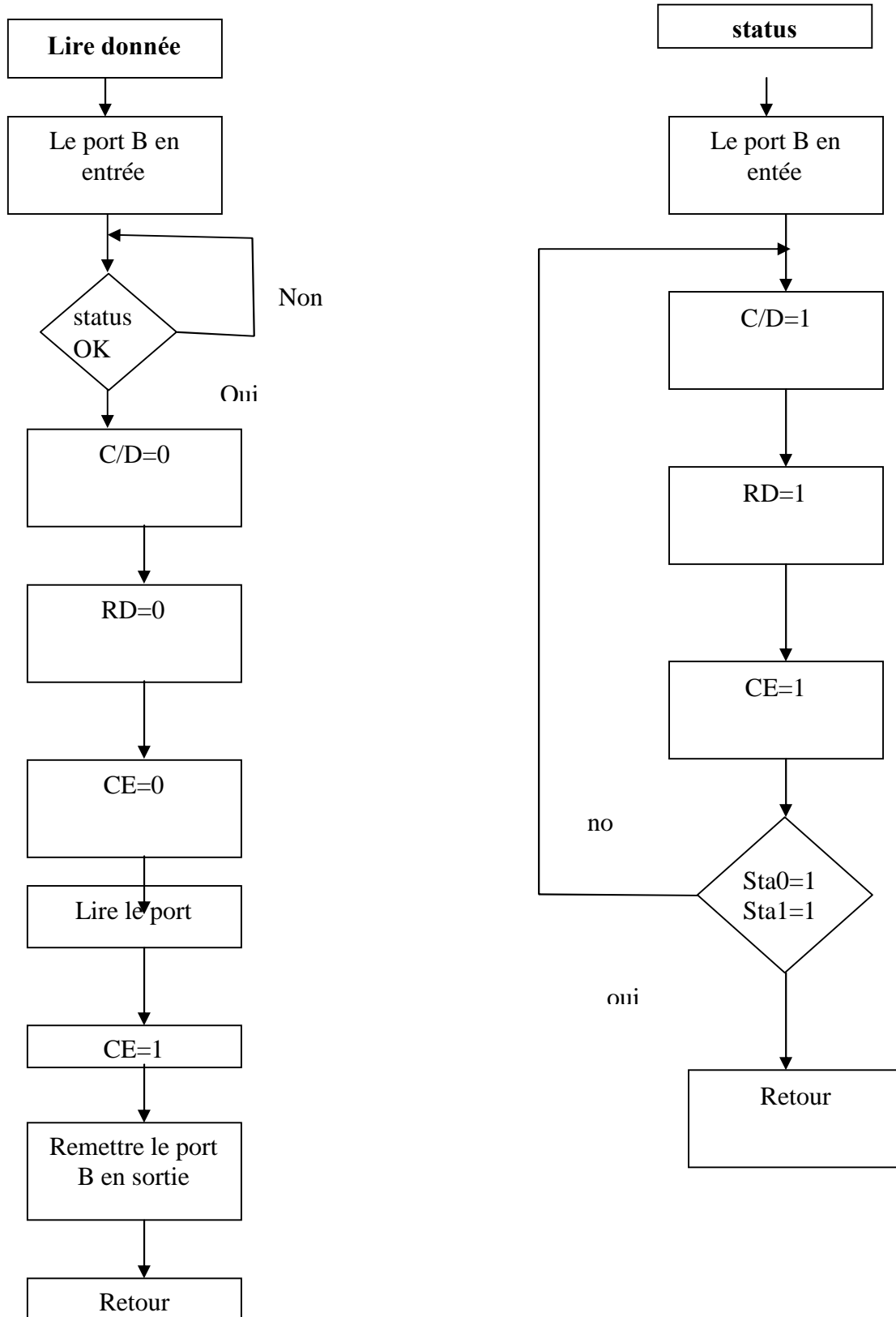


Figure 1.8 Organigrammes pour lire une donnée

1.5 LE pH METRE

1.5.1. Etalonnage

Les caractéristiques des électrodes diffèrent en fonction des fabricants et des modèles. Elles peuvent avoir une certaine dispersion dans la même série. De plus, ces caractéristiques (pH_0 , pente) évoluent dans le temps.

C'est lors de l'étalonnage, que le pH mètre va déterminer et prendre en compte les caractéristiques réelles de l'électrode. Pour étalonner, on utilise généralement deux solutions tampons différentes. Le résultat d'un étalonnage nous permet de déterminer deux paramètres : le pH_0 et la pente (sensibilité).

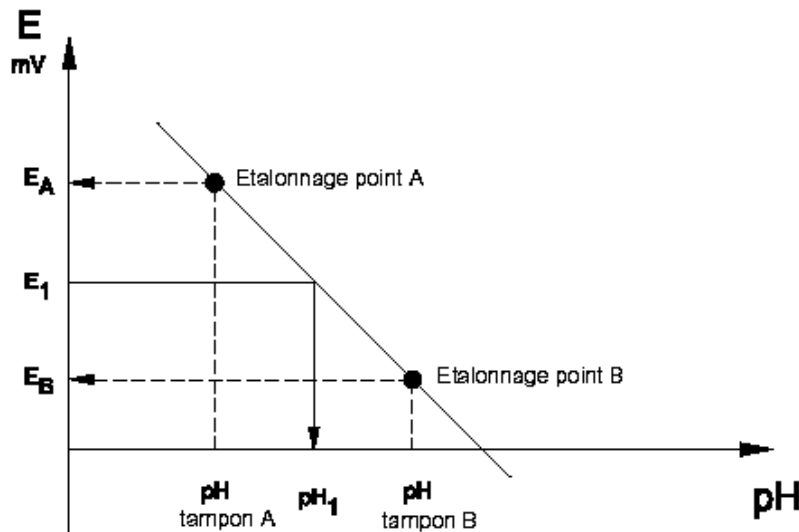


Figure 1.9 : Courbe d'étalonnage

Le microcontrôleur doit reconnaître automatiquement les tampons utilisés pour l'étalonnage. Cette opération est primordiale pour la détermination de la caractéristique de l'électrode ou en d'autres termes de la relation « pH-tension » de l'électrode pour le traitement numérique ultérieur. Les solutions tampons utilisées selon la norme Tacussel sont pH 4, pH 7 et parfois pH 10. (MOKADEM,1998)

La procédure d'étalonnage utilise deux (2) points (figure 1.9). Il suffit de plonger l'électrode dans le premier tampon et attendre que la mesure se stabilise. Le PIC mémorise la tension correspondante, la même opération est faite pour un deuxième tampon. Le PIC détermine alors la pente exacte de la caractéristique du capteur, cette procédure est donnée par l'organigramme suivant :

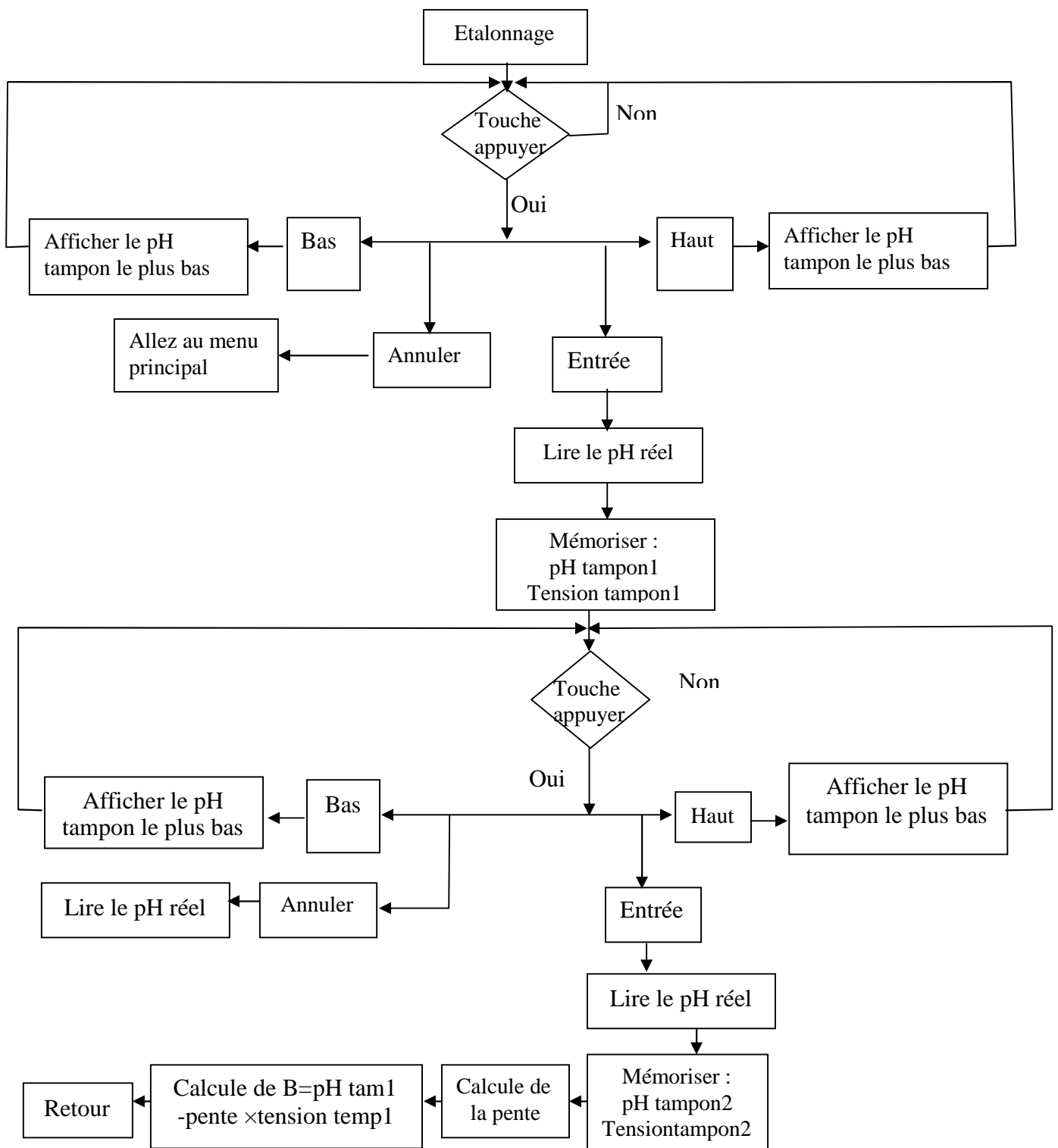


Figure 1.10 Etalonnage du pH mètre

1.5.2. Influence de la température

La température joue un rôle important sur l'échantillon, sur les solutions tampons et sur les caractéristiques de l'électrode. L'influence de la température sur la valeur des solutions tampons est parfaitement connue et est indiquée sur tous les flacons de tampons qu'on a utilisés. Ces valeurs sont mémorisées dans l'EEPROM externe. La variation du pH, due à la température, est minimale pour les solutions inorganiques de tampons acides, alors qu'elle est significative pour les tampons alcalins et quelques tampons organiques.

Idéalement, la température doit être la même durant la phase d'étalonnage et les mesures. Dans la pratique, une différence de 2 à 5 °C est acceptable dans la plupart des cas. Lorsque l'on trace les droites $pH = f(mV)$ à différentes températures, on s'aperçoit qu'elles se croisent presque toutes en un même point (figure 1.11).(Mokadem,1998)

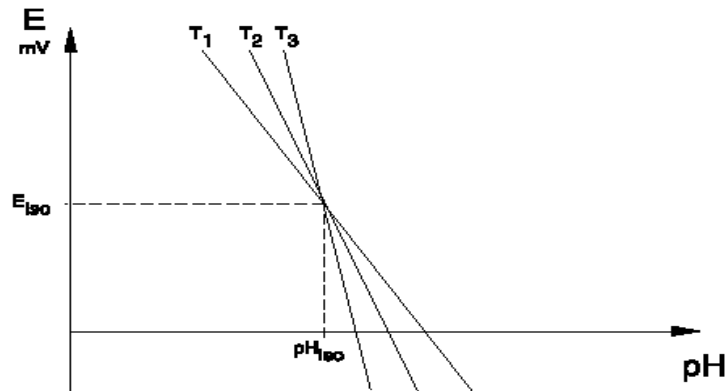


Figure 1.11 : Représentation graphique du pH_{iso} .

Dans notre cas le microcontrôleur va lire la valeur de pH réelle correspondant à la température introduite par l'utilisateur.

1.5.3 Réalisation d'un pH mètre

Le premier résultat logique intermédiaire que nous avons obtenu est la réalisation d'un pH-mètre à base de microcontrôleur PIC16F876 (A). Le schéma électrique est donné ci-dessous :

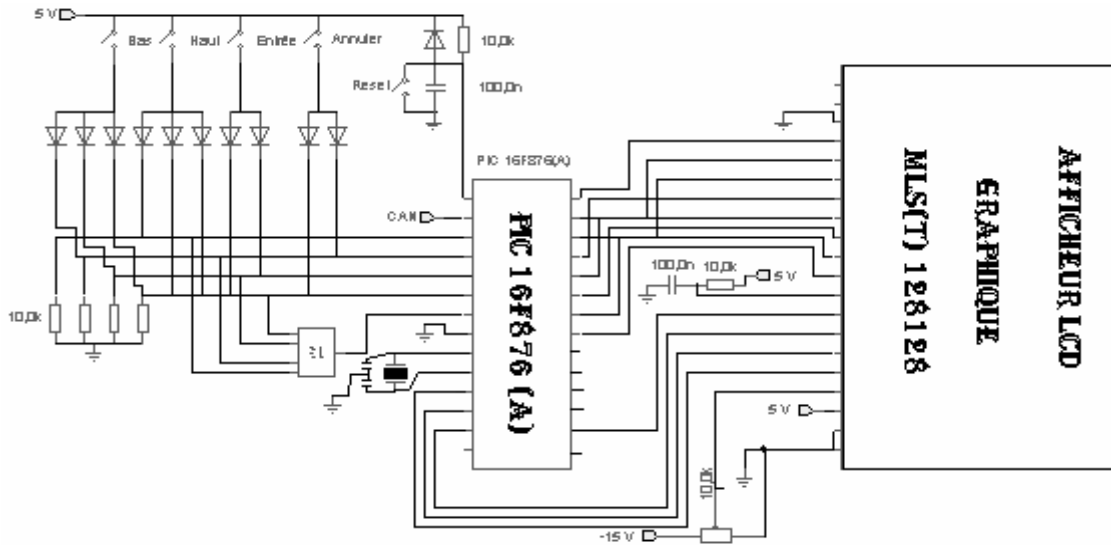


Figure 1.12 : Schéma électrique d'un pH mètre

Il contient un circuit de reset manuel en cas de blocage du programme de gestion, des boutons poussoir pour la manipulation des données, un oscillateur, un LCD et un circuit de détection des touches. L'organigramme pour gérer le pH mètre est le suivant :

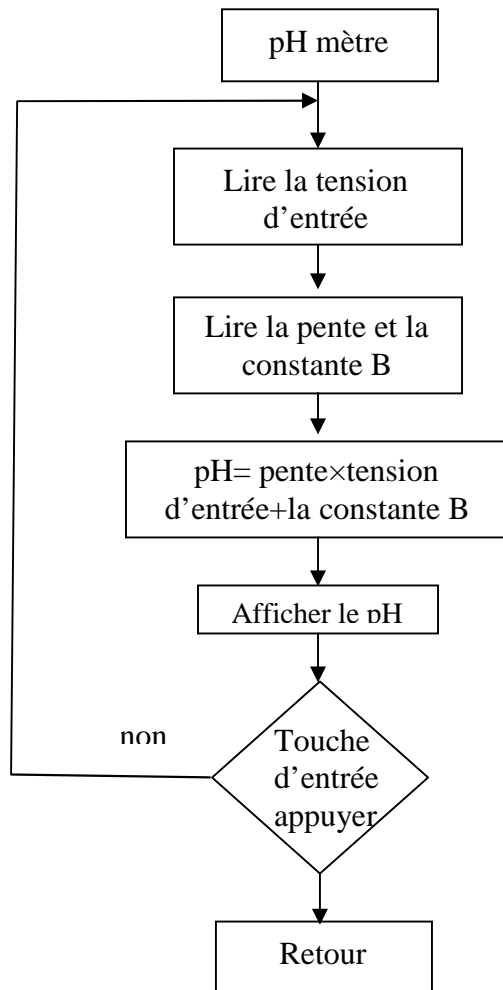


Figure 1.13 Organigramme de gestion du pH mètre

1.6 LES COURBES DE TITRAGE (DIDIER et GRECIAS, Paris 1995)

Il existe deux manières de tracer une courbe de titrage:

i) les courbes sigmoïdales: lorsque d'importantes variations s'observent dans une petite région entourant le point d'équivalence, on reporte la fonction p (ou la fonction logarithmique) du réacteur en fonction du volume du réactif. C'est ce type de représentation qui est le plus souvent utilisé.

ii) les courbes à segments linéaires: ce type de graphique est approprié lorsque les réactions ne sont complètes qu'en présence d'un grand excès de réactif ou de réacteur.

On reporte en abscisse le volume du réactif et en ordonnée les mesures instrumentales qui sont directement proportionnelles à la concentration de réacteur ou du réactif. Dans ce cas, on ignore les mesures proches du point d'équivalence pour ne considérer que les points qui sont éloignés de celui-ci.

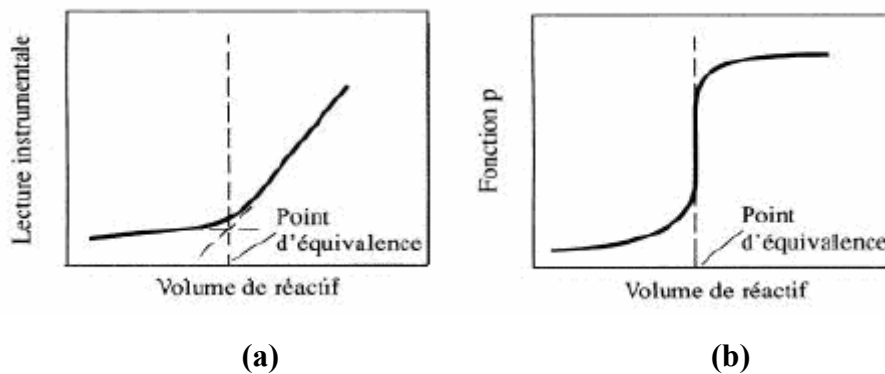


Figure 1.14 Les deux types de la courbe de titrage.

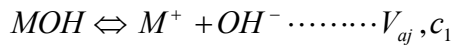
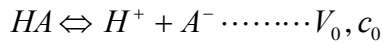
(a) les courbes sigmoïdales (b) les courbes à segments linéaires

1.6.1 Titrage d'un acide fort par une base forte

Il faut remarquer que, quel que soit le titrage envisagé, le volume total de la solution à titrer varie tout au long du titrage. Il faut donc corriger la concentration C_0 en fonction du volume de réactif ajouté

$$C_{0_{\text{corrigé}}} = C_0 \frac{v_0}{v_0 + v} \quad (1.4)$$

Dans le cas d'un titrage d'un acide fort par une base forte, il suffit de déterminer $[H_3O^+]$ en considérant que l'acide et la base sont totalement dissociés.



$$[H_3O^+] = \frac{\text{nbre.mmol.initial} - \text{nbre.mmol.ajouté}}{\text{volume total}} \quad (1.5)$$

$$[H_3O^+] = \frac{(V_0 c_0) - (V_{aj} c_0)}{V_0 + V_{aj}}$$

La réaction peut être décrite par l'équation:



Bilan des charges:

$$[H^+] + [M^+] = [OH^-] + [A^-] \quad (1.7)$$

Bilan des masses:

Puisqu'il s'agit d'un acide fort, il est totalement dissocié.

Alors

$$c_0 = [A^-] \quad (1.8)$$

A l'aide de ces deux équations et du produit ionique de l'eau, on peut écrire une équation quadratique:

$$[H^+] + a.c_0 = \frac{K_e}{[H^+]} + c_0 \Leftrightarrow [H^+]^2 + [H^+]c_0.(a-1) - K_e = 0 \quad (1.9)$$

Le pH est donc déterminé à l'aide de l'équation:

$$[H_3O^+] = \frac{C_0.(1-a) + \sqrt{\{c_0.(a-1)\}^2 + 4K_e}}{2} \quad (1.10)$$

L'importance du saut de pH est déterminée par la concentration initiale de l'acide fort présent puisque le pH d'une solution d'acide fort est donné par $pH = -\log c_0$. Afin d'obtenir des courbes de titrage représentatives, il faut travailler avec des concentrations assez élevées.

Avant de discuter des titrages mettant en présence un acide faible, il convient d'introduire la notion de tampon.

1.6.2 Les tampons (Mokadem, 1998)

Un tampon est un mélange d'un acide faible et de sa base conjuguée. Un tel mélange résiste aux variations imposées à la solution. Chaque fois qu'un acide faible est titré par une base, il se forme une solution tampon. Les tampons sont utiles dans tous les types de réaction où l'on désire maintenir le pH à une valeur constante prédéterminée.

Le calcul du pH d'une solution tampon s'effectue à l'aide de la constante de dissociation K_a

$$\begin{aligned}
 HA + H_2O &\Leftrightarrow H_3O^+ + A^- \dots K_{a_{1-0}} \\
 K_a &= \frac{[H_3O^+][A^-]}{[HA]} \Leftrightarrow -\log K_a = -\log[H_3O^+] - \log \frac{[A^-]}{[HA]} \\
 &\Leftrightarrow pH = pK_a + \log \frac{[A^-]}{[HA]}
 \end{aligned} \tag{1.11}$$

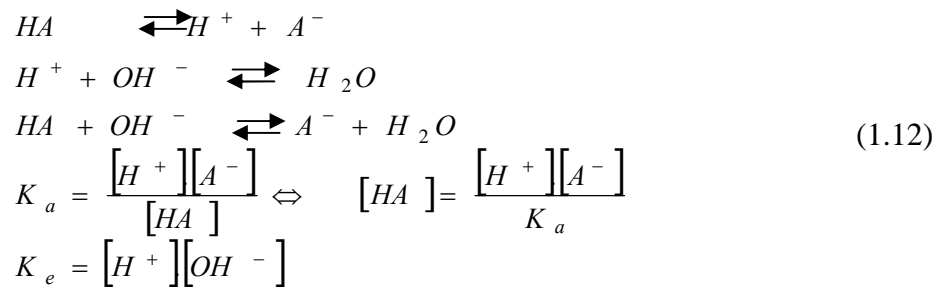
Le pH d'une solution est déterminé par le pKa de l'acide présent ainsi que le rapport des concentrations de l'acide et de sa base conjuguée. Il est indépendant de la dilution: c'est seulement le rapport des concentrations qui détermine le pH. Néanmoins plus le système est concentré, plus sa capacité à résister à un saut de pH sera grande.

Avec l'équation d'Hasselbalch-Henderson (1.6), il est aisé de décrire qualitativement l'évolution du pH d'une solution tampon en fonction du rapport $\frac{[A^-]}{[HA]}$.

1.6.3 Titrage d'un acide faible par une base forte (Falzoni, 1998)

Dans ce cas, il faut tenir compte de la dissociation partielle de l'acide faible.

La résolution exacte donne lieu à une équation du 3ème degré.



Bilan des charges:

$$[H^+] + [M^+] = [A^-] + [OH^-] \tag{1.13}$$

Bilan de masse:

$$c_0 = [HA] + [A^-] \tag{1.14}$$

On substitue [HA] par la valeur obtenue à partir de Ka:

$$\begin{aligned}
 c_0 &= \frac{[H^+][A^-]}{K_a} + [A^-] \Leftrightarrow c_0 = [A^-] \left(1 + \frac{[H^+]}{K_a} \right) \\
 \Leftrightarrow [A^-] &= \frac{c_0}{1 + \frac{[H^+]}{K_a}}
 \end{aligned} \tag{1.15}$$

Puisque $[M^+] = a \times c_0$, on obtient:

$$\begin{aligned}
 [H^+] + a.c_0 &= \frac{K_e}{1 + \frac{c_0}{K_a}[H^+]}} \\
 \Leftrightarrow \frac{[H^+]^3}{K_a} + [H^+]^2 \left(\frac{a.c_0}{K_a} + 1 \right) + [H^+] \left\{ c_0.(a-1) - \frac{K_e}{K_a} \right\} - K_e &= 0
 \end{aligned}
 \tag{1.16}$$

Plusieurs étapes sont nécessaires pour établir une courbe de titrage d'un acide faible par une base forte:

- i) Initialement, la solution ne contient qu'un acide faible et le pH se calcule à partir de sa concentration initiale et de sa constante de dissociation.
- ii) Après l'addition de différents volumes du réactif titrant (dans tout le domaine qui précède le point d'équivalence), la solution est une solution tampon. Le pH peut être calculé à l'aide de l'équation d'Hasselbalch-Henderson.
- iii) Au point d'équivalence, la solution ne contient que le conjugué de l'acide faible qui est titré. Le pH se calcule à partir de la seule concentration de ce produit.
- iv) Au delà du point d'équivalence, l'excès de base forte, masque le caractère basique de la base conjuguée. Le pH dépend essentiellement de la concentration du réactif titrant présent en excès.
- v) pH de la solution de départ, avant l'adjonction du titrant (la base forte), on se trouve en présence d'un acide faible dont la concentration $c_0 \gg [H^+]$.

$$\begin{aligned}
 [H^+] \ll c_0 - [H^+] \approx c_0 \Leftrightarrow K_a &= \frac{[H^+]^2}{c_0 - [A^-]} \\
 \Leftrightarrow [H^+] = \sqrt{c_0 \cdot K_a} \Leftrightarrow pH &= \frac{1}{2} \cdot (pK_a - \log c_0)
 \end{aligned}
 \tag{1.17}$$

vi) Zone tampon.

Dès l'adjonction d'une base forte (le titrant), la solution forme un tampon. Le titrant réagit quantitativement avec l'acide faible pour donner un couple acide-base conjugué. L'équation d'Hasselbalch-Henderson permet de déterminer le pH du tampon correspondant.

$$\begin{aligned}
 K_a = \frac{[H_3O^+][A^-]}{[HA]} \Leftrightarrow -\log K_a &= -\log [H_3O^+] - \log \frac{[A^-]}{[HA]} \\
 \Leftrightarrow pH = pK_a + \log \frac{[A^-]}{[HA]}
 \end{aligned}
 \tag{1.18}$$

vii) Le point d'équivalence.

Lorsqu'un équivalent de titrant (la base forte) a été ajouté, on peut considérer que la solution ne contient que le sel conjugué de l'acide faible à titrer. Par conséquent, le pH de cette solution est déterminé par K_a .

viii) Après le point d'équivalence.

Passé le point d'équivalence, la présence d'excès d'ions hydroxydes de la base forte masque l'équilibre entre l'acide faible et sa base conjuguée.

1.6.4 Les polyacides : (DIDIER, GRECIAS, 1995)

Lorsqu'un acide contient plusieurs protons dissociables, la déprotonation se fait le plus souvent par étapes successives. Lorsque la différence des pK_a est suffisamment grande ($\Delta pK_a \geq 4$), les dissociations peuvent être considérées comme des dissociations isolées.

Le plus souvent, cette condition est remplie pour les polyacides pour deux raisons:

i) Electrostatique:

- le premier proton est extrait d'une molécule neutre.
- le second proton par contre doit être extrait d'une molécule chargée négativement, ce qui est nettement moins favorable d'un point de vue électrostatique.

ii) Statistique: lors de la première déprotonation, il y a deux sites équivalents qui peuvent perdre un proton. La seconde déprotonation est statistiquement deux fois moins probable, car il n'existe plus qu'un site protoné.

D'une manière générale, plus les deux sites sont éloignés géométriquement (et découplés électroniquement), plus la différence de pK_a entre les deux sites sera faible.

1.6.4.1 Titrage d'un acide faible monoproté par une base forte (Jutila, 1984)

On simule le titrage de 30ml d'une solution aqueuse d'acide acétique CH_3COOH de concentration en soluté apporté $CA=0.1\text{mole/L}$, par une solution de soude $NaOH$ de concentration $CB=0.1\text{mole/L}$. La figure (II.2) fournit la variation du pH en fonction du volume de soude versée au cours de titrage.

Pour la simulation de ce processus les réactions suivantes sont modélisées :

$NaOH$: une base forte qui se protolise dans l'eau totalement



CH_3COOH : est un acide faible qui se dissocie dans l'eau partiellement



Et pour l'eau :



Calcul du pH

Dans ce qui suit le pH est évalué à des instants différents d'ajout de base,

Le point initial :

$$[H^+]_{CH_3COOH} = \sqrt{C_{acide} \cdot K_a} \quad (1.22)$$

Par ce que :

$$\begin{aligned} [C = 0.1] \text{ et } [K_a = 1.7378 \times 10^{-5}] \\ [H^+] = 1.32 \times 10^{-3} \text{ M} \Rightarrow pH = 2.88 \end{aligned}$$

Pour :

$$V = 0; pH = 2.8$$

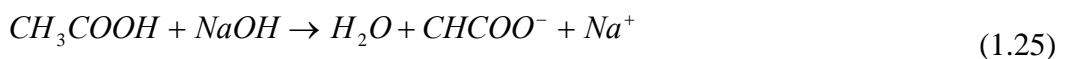
Dans la zone entre le point initial et le point d'équivalence nous avons :



Pour le tampon on a :

$$pH_{tampon} = pK_a + \log \frac{[CH_3COO^-]}{[CH_3COOH]} \quad (1.24)$$

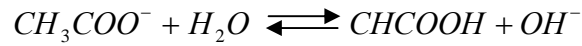
Mettons 5.00mL de NaOH, il se produit :



$$pH = 4.76 + \log \frac{0.5 \cdot 10^{-3}}{2.5 \cdot 10^{-3}} = 4.061$$

A 15mL, pH=4.76 c'est le pKa de l'acide.

Au point d'équivalence : tout l'acide est neutralisé soit :



$$[OH^-] = [K_e C_{sel} / K_a]^{1/2}$$

$$[CH_3COO^-] = 3.10^{-3} M$$

$$[OH^-] = 5.364 \times 10^{-6} M, pOH = 5.27$$

$$pH = 8.73$$

$$V = 30 mL; pH = 8.73$$

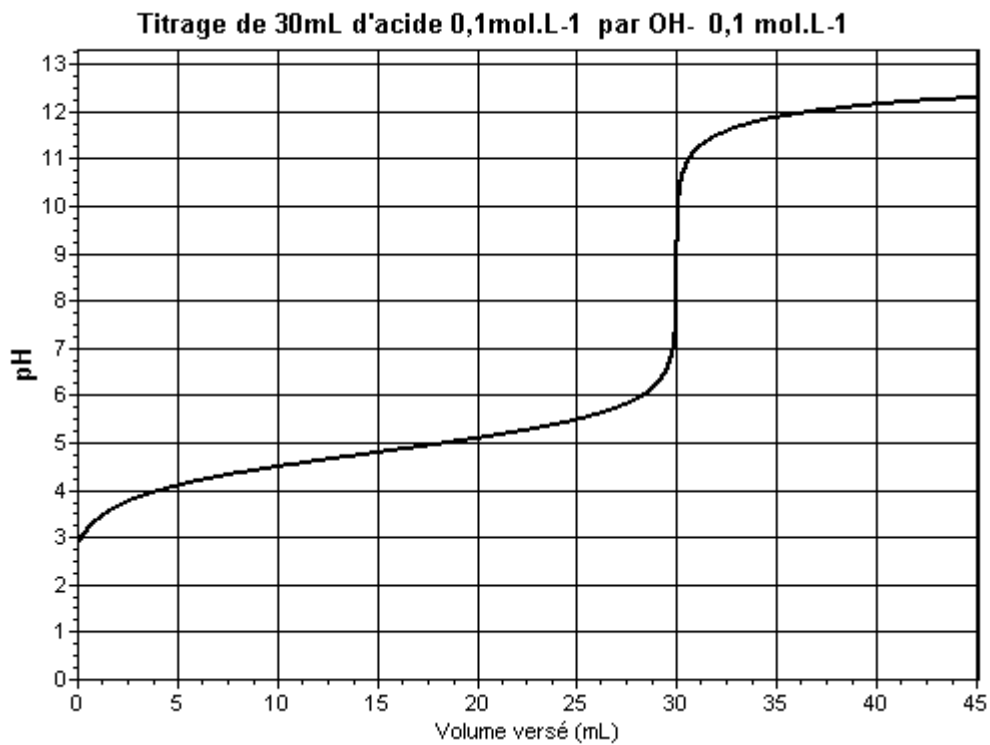


Figure 1.15 : Neutralisation de CH_3COOH par la soude (Bonnin,2004)

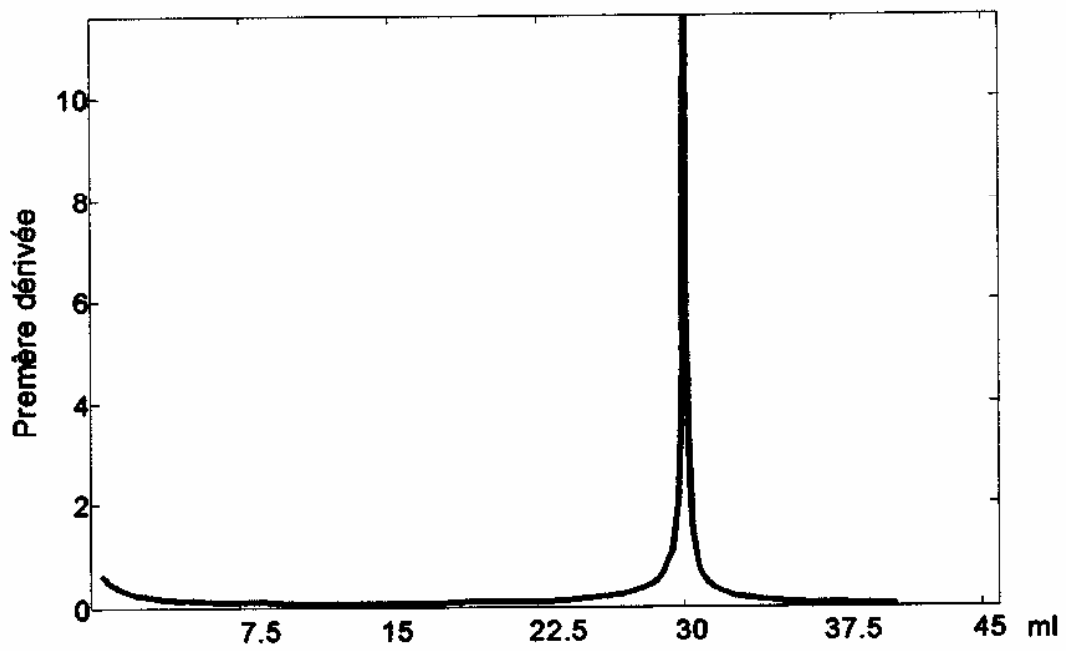


Figure 1.16 : la première dérivée de la courbe de titrage

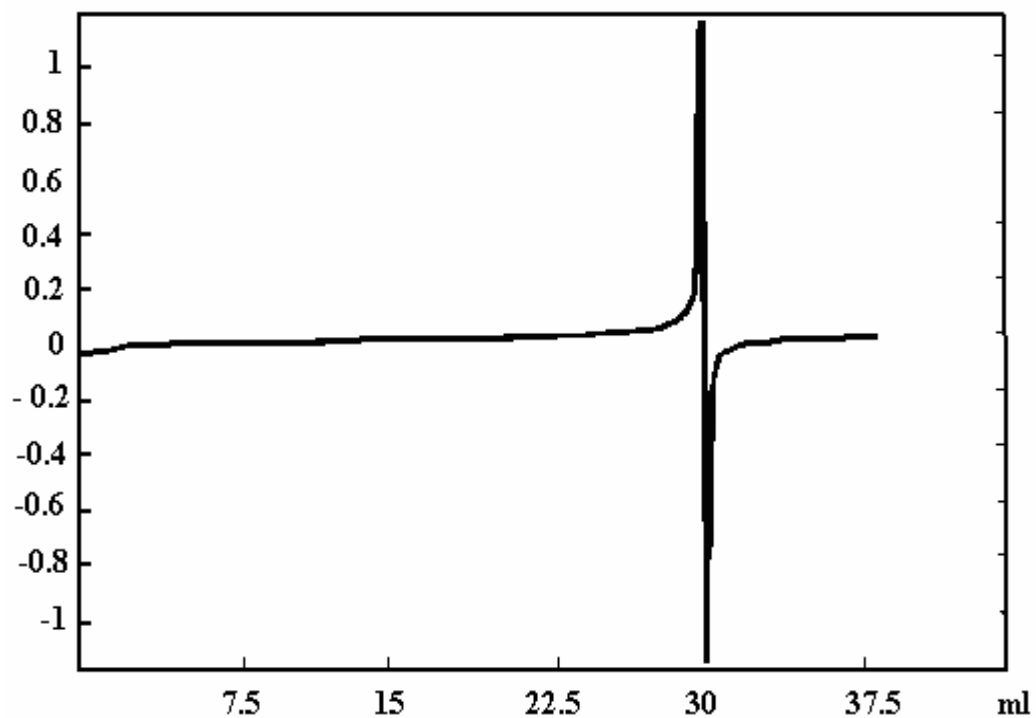


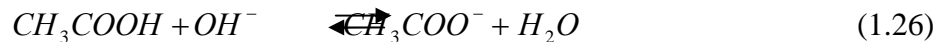
Figure 1.17: la deuxième dérivée de la courbe de titrage

L'allure représentée dans la figure 1.17 permet de conclure qu'elle vire pour une quantité de soude versée égale à celle nécessaire pour la neutralisation de l'acide.

Sur la courbe de titrage acide fort -base forte en raison de l'équilibre d'ionisation de l'acide faible, comme les conditions d'une solution tampon sont établies, il y a, au départ, un accroissement lent du pH. Le changement brusque survient lorsqu'on on approche le point d'équivalence. Il y a, cette fois encore, une augmentation rapide du pH autour du point d'équivalence.

Un titrage est réalisable, au voisinage de l'équivalence, s'il se produit une modification décelable d'une des propriétés de la solution (pH, température, couleur). Si c'est le cas, le pH du point d'équivalence est déterminé avec précision.

Le pH à l'équivalence est déterminé selon l'équation de la réaction :



La réaction étant totale, à l'équivalence les espèces CH_3COOH et OH^- ont totalement disparu. La solution est alors basique, le pH est donc supérieur à 7.

Pour n'importe quel acide la constante de dissociation K_a ne dépend que de la température.

La courbe de titrage, donnée au schéma précédent, montre qu'autour du point d'équivalence un changement brusque du pH de plusieurs unités survient dans un intervalle de quelques centièmes de millimètre.

1.6.4.2 Titrage d'un acide faible triprotoné par une base forte

On a fait le titrage de 10 ml d'une solution aqueuse d'un acide phosphorique, de concentration en soluté 0.1 mole/l, par une solution de soude de concentration 0.1 mole/l. L'acide phosphorique est un triacide, avec des acidités ayant des forces différentes. La première fonction acide est une fonction acide forte. La deuxième est une fonction acide moyenne, quant à la troisième c'est une fonction acide faible.

La courbe de titrage présente deux points d'inflexion correspondant à la neutralisation de la première et de la deuxième acidité. La dérivée de la courbe montre que le point d'équivalence passe par un maximum, pic étroit.

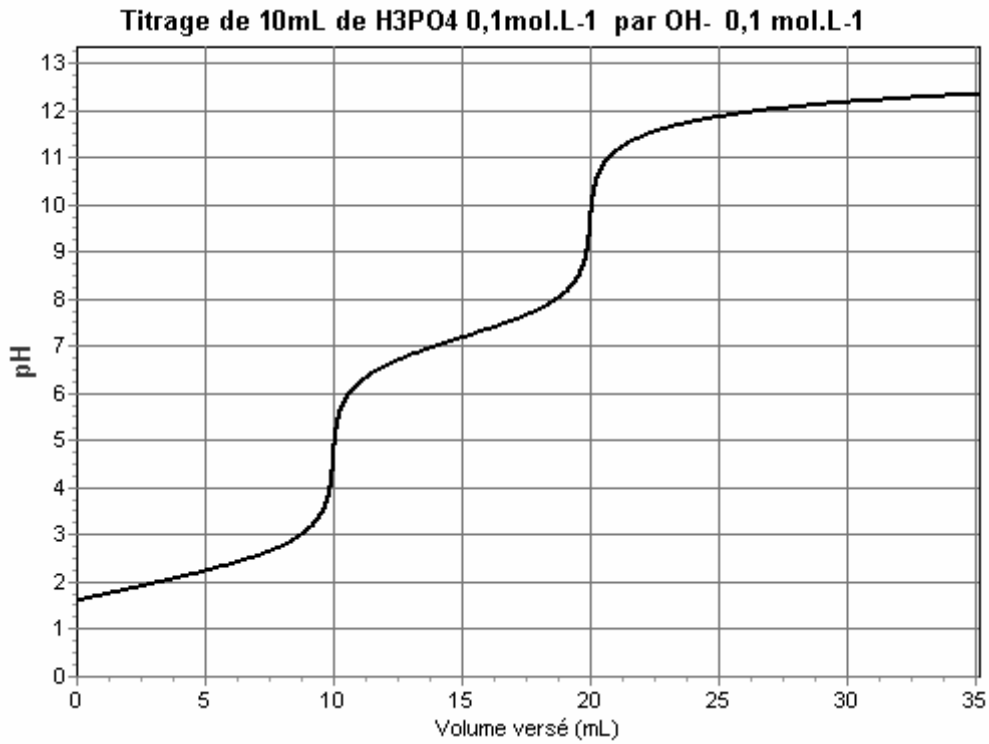


Figure 1.18 : Neutralisation de H_3PO_4 par $NaOH$

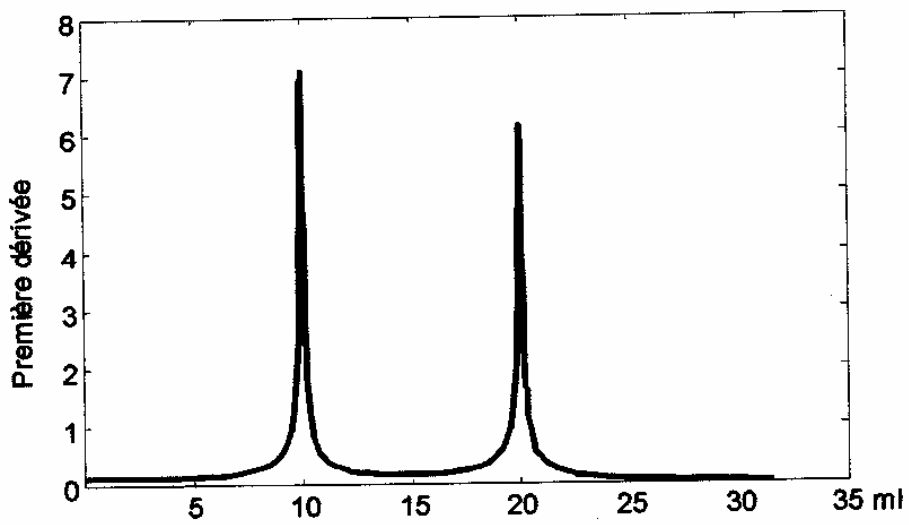


Figure 1.19 : la première dérivée de la courbe de titrage



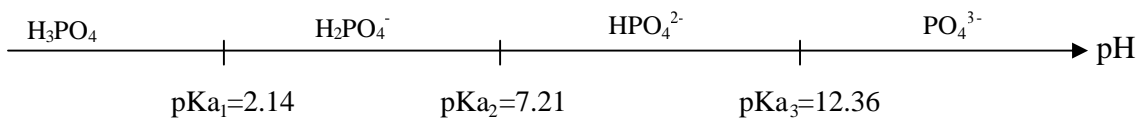
Figure 1.20 : la deuxième dérivée de la courbe de titrage

On peut noter que les domaines tampons sont centrés sur les valeurs des pKa qui sont :

$$pK_1=2.14$$

$$pK_2=7.21$$

$$pK_3=12.36$$



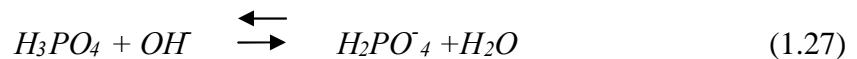
Les points d'équivalence :

$$pH = \frac{1}{2}(pk_1 + pk_2) = 4.68$$

$$pH = \frac{1}{2}(pk_2 + pk_3) = 9.78$$

Au départ, l'acide phosphorique (H_3PO_4) se comporte comme un monoproté faible, la première dissociation a lieu au point ($V_e=10\text{ml}$, $pH= 4.68$).

Dans cet intervalle (0 à 10 ml) il se produit la réaction suivante :



La valeur du pH au premier point d'équivalence est donnée par la relation suivante :

$$pH = \frac{1}{2}(pk1 + pk2)$$

Entre le premier point et le deuxième point d'équivalence nous avons la réaction suivante :



D'où le pH au deuxième point d'équivalence :

$$pH = \frac{1}{2}(pk2 + pk3) \tag{1.28}$$

Après le deuxième point d'équivalence, le pH évolue très lentement comme le décrit la figure 1.18.

1.7 DESCRIPTION DU BANC D'ESSAI

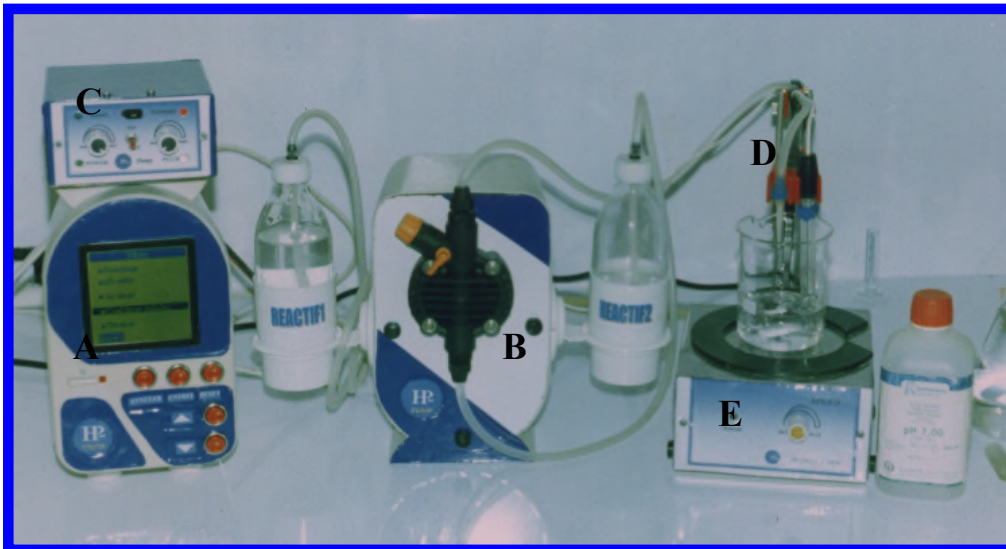


Figure 1.21 : Banc de titration

L'ensemble est une solution complète offrant tous les éléments nécessaires à la mise en place d'une station autonome de travail, il est composé des parties suivantes (figure 1.21) :

A- La centrale de commande:

C'est la partie intelligente du système. Elle assure l'acquisition, le traitement et l'affichage des données sur écran graphique en temps réel. De plus nous avons prévu un menu avec des textes clairs pour l'exploitation des programmes.

B- La tête doseuse :

Elle est fabriquée en PVC avec une membrane en téflon incorporée dont le principe de fonctionnement est très simple, elle aspire le produit doseur et l'injecte dans le réacteur avec une haute précision. L'ensemble résiste à tous les acides.

Chaque impulsion délivrée par le microcontrôleur (ou le circuit intégré NE555) atteint la gâchette d'un triac à travers un opto-coupleur (figure II.9). Ce triac en mode passant ferme un pont de diodes et délivre à sa sortie une tension continue de 220v qui alimente l'électro-aimant. Le piston de ce dernier sous l'effet du champ magnétique crée une pression dans le doseur grâce à une membrane en téflon et expulse par le mouvement de celle-ci le réactif par le clapet de refoulement. D'autre part le même piston comprime un ressort de rappel par l'intermédiaire d'un disque en fer doux. Quand l'impulsion disparaît le ressort de rappel ramène le piston (la membrane) à sa position initiale avec cette fois-ci une aspiration du réactif grâce à un clapet prévu à cet effet.

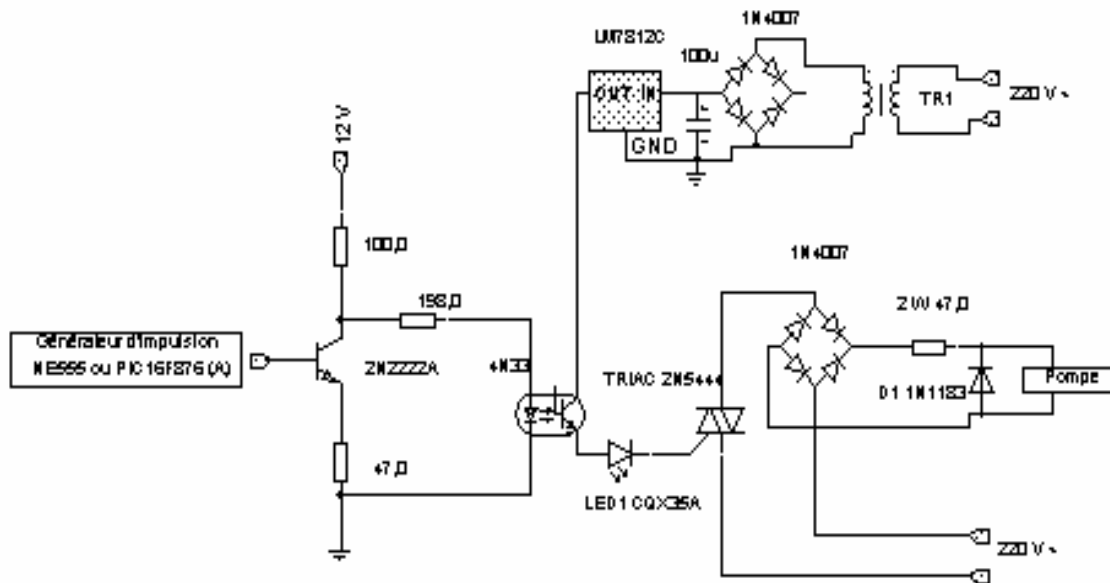


Figure 1.22 : Commande de la pompe doseuse.

C- L'électrode de pH, munit d'un support (statif), adaptable à toutes les situations de l'expérience. Il comporte aussi un embout pour une injection convenable du réacteur.

D - L'agitateur magnétique non chauffant :

C'est un appareil très important pour une opération de titration automatique, il a été conçu au laboratoire et il comporte un système d'aération pour le refroidissement du moteur. Ces caractéristiques sont citées ci-dessous :

- Une capacité d'agitation de 12 l.
- Une vitesse allant de 100 à 1800 tours/ minute.

-Une puissance de 500 Watts.

-Poids 1,2 Kg.

1.8 Détermination du point d'équivalence

La courbe de titrage, qui donne la variation du pH en fonction du volume v de solution titrante versée (par exemple base forte NaOH), présente un point d'inflexion sur la caractéristique (pH, V) , lequel se traduit par un maximum sur la courbe du gradient $\frac{dpH}{dv} = f(v)$. Ce point est appelé point d'équivalence.

1.8.1. Méthode manuelle

On utilise dans cette méthode les solutions colorées, qui sont des substances dont la couleur varie avec l'acidité du milieu, le changement de la coloration se produit dans un certain domaine de pH appelé zone de virage qui s'étend généralement sur 1.5 à 2 unité de pH.

Il y a deux types de graphes obtenus :

La courbe symétrique : cette méthode est très bien illustrée par la figure ci-dessous :

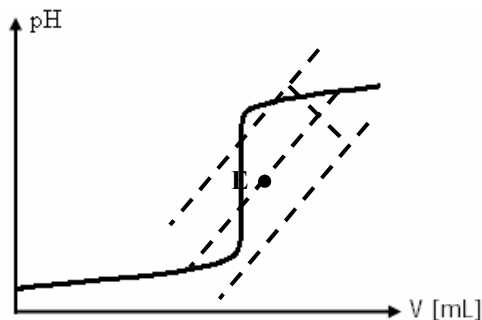


Figure 1.23 : Détermination du point d'équivalence pour une courbe symétrique

La courbe asymétrique : le point d'équivalence est déterminé par le tracé correspondant à l'intersection de la droite reliant les deux centres et la courbe de titrage (figure 1.24).

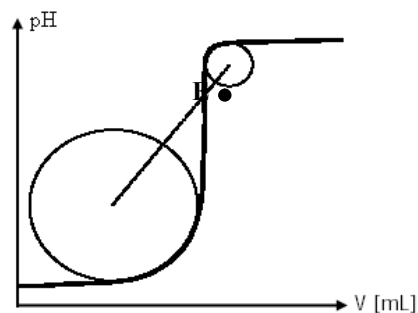


Figure 1.24 : Détermination du point d'équivalence pour une courbe asymétrique

1.8.2 Méthode numérique : pour une détection expérimentale convenable du point d'équivalence il est utile de prendre en considération deux aspects fondamentaux :

- Le volume impulsionnel.
- La cadence des impulsions.

En effet si le volume impulsionnel est trop important le point d'équivalence n'est pas détecté ceci est illustré par la figure ci-dessous :

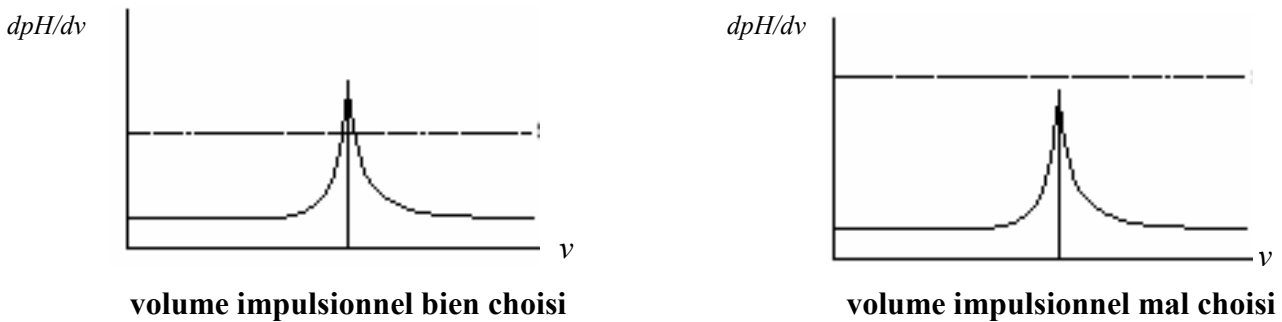


Figure 1.25 : Choix expérimental du volume impulsionnel

D'autre part si le volume impulsionnel est trop faible on voit apparaître plusieurs maximums sur la courbe du gradient (figure 1.26, le point recherché est impossible à localiser).

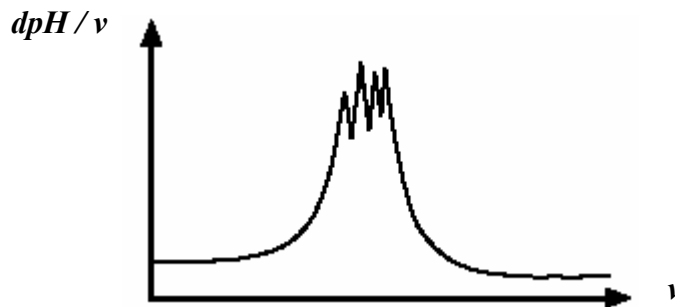


Figure 1.26 : Le gradient de la courbe de titrage pour un volume trop faible

Les conditions pratiques favorables qui tiennent compte de l'environnement expérimental (temps réponse du capteur, filtrage analytique du signal pH, préparation des solutions ...) ont conduit au choix des valeurs suivantes :

- Volume impulsionnel : 0,1 ml / impulsion.
- Cadence des impulsions : 0,6 ml / minute.

La courbe ci-dessous traduit la qualité de la mesure obtenue pour ce cas :

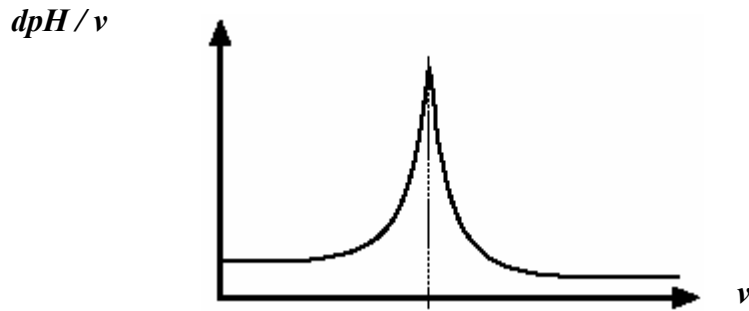


Figure 1.27 : Le gradient avec des conditions pratiques favorables.

1.8.3 METHODE DE GRAN (Rudan-Tasic, Klofutar, 1998)

Cette méthode consiste à linéariser les résultats expérimentaux. On définit une fonction f , appelée fonction de Gran, qui dépend des grandeurs connues (telle V_0) et des grandeurs mesurées (V , volume ajouté, pH) telle que f soit une fonction affine de v qui s'annule pour $V=V_{eq}$.

Bien entendu, une telle linéarisation n'est possible que dans certains domaines.

1.8.3.1 Dosage d'un acide fort par une base forte

Nous pouvons nous placer dans deux domaines :

- $V < V_{eq}$, c'est-à-dire avant l'équivalence : la réaction étant totale, nous avons :

$$h = \frac{V_0 C_0 - V C_1}{V + V_0} \quad (1.29)$$

$$f_1 = (V + V_0)h = (V + V_0)10^{-pH} = V_0 C_0 - V C_1 = C_1 (V_{eq} - V) \quad (1.30)$$

Soit $f_1 = (V + V_0)10^{-pH}$ pour la première fonction de Gran

Cette droite s'annule bien pour $V = V_{eq}$: deux points suffisent pour la construire.

- $V > V_{eq}$, c'est-à-dire après l'équivalence : il y a un excès d'ions OH^-

Soit

$$[OH^-] = \frac{V C_1 - V C_0}{V + V_0} = \frac{K_e}{h} \quad (1.31)$$

On déduit sans difficulté $f_2 = (V_0 + V)10^{14-pH} = C_1 (V - V_{eq})$

Cette deuxième droite, de pente positive et symétrique de la précédente, s'annule bien pour : $V=V_{eq}$.

1.8.3.2. Cas de dosage d'un acide faible par une base forte

Comme dans le cas précédent, cette méthode de linéarisation permet de déterminer le volume équivalent.

Avant l'équivalence, la réaction étant considérée comme quantitative, on a :

$$K_a = \frac{VC_1 h}{V_0 C_0 - VC_1} \quad (1.32)$$

$$f = hV = \frac{K_a}{C_1} (V_0 C_0 - VC_1), \quad (1.33)$$

Cette expression se transforme en :

$$f_1 = V10^{-pH} = K_a (V_e - V) \quad (1.34)$$

Le tracé $f_1 (v)$ permet non seulement de déterminer V_{eq} mais également la constante d'acidité K_a (la pente de la droite).

Après l'équivalence, il y a un excès de OH^- et nous sommes ramenés à la même situation que celle à l'acide fort et la base forte. On retrouve la même fonction f_2 .

1.9 MESURES EXPERIMENTALES

1.9.1 Analyse de l'acide acétique

a) Mode opératoire :

- Volume initial : $V_0 = 50$ ml.
- Réactif : base forte (KOH) de molarité $C_b = 0.5$ M.
- Objectif : neutraliser l'acide acétique par la base et localiser avec précision le point d'équivalence. Cette opération nous permet alors de déterminer le paramètre très important de la solution monoprotée qu'est le pKa.

b) Etude à partir de la courbe de gradient :

La caractéristique (pH, volume) est sujette à une augmentation très rapide dans la zone du point d'inflexion, ce qui se traduit par une grande imprécision sur l'évolution des coordonnées (pH_e , V_e) du point d'équivalence.

En revanche la courbe du gradient (dpH/dv , v) permet de palier cette difficulté.

c) Etude à partir de la fonction de Gran :

Cette méthode offre un moyen élégant et extrêmement simple pour la détermination du point d'équivalence.

En effet cette fonction qui a été définie plus haut s'annule exactement au point $V = V_e$. Ainsi on conçoit aisément la précision offerte par cette technique.

d) Comparaison des deux méthodes :

Le tableau ci-dessous fait ressortir la comparaison entre les méthodes sus-citées :

	Méthode du gradient	Méthode du Gran
V_e	11.20 ml	11.20 ml
pKa	4,74	4.79

Tableau 1.1 Résultats d'analyse de l'acide acétique par les méthodes de Gran et gradient

1.9.2 Dosage de l'acide éthanoïque contenu dans un vinaigre

Le vinaigre, condiment liquide au goût amer contient au moins 6 % d'acide acétique. Le vinaigre est le produit de deux fermentations. La première, assurée par la levure, transforme le sucre dissous en alcool et donne une liqueur de 6 à 9 degrés, appelée *brassin*. Le brassin fermente à son tour sous l'action de l'*Acétobacter*, une bactérie aérobie, qui transforme l'alcool en acide acétique. C'est Pasteur qui, le premier, a expliqué les processus microbiologiques de la fabrication du vinaigre.

a) Notre objectif :

Effectuer un dosage de l'acide éthanoïque contenu dans un vinaigre.

b) Confection de la solution à doser :

Afin que le vinaigre soit moins concentré, on réalise une dilution au $1/10^{\circ}$ du vinaigre commercial.

A l'aide d'une pipette jaugée, on prélève 10 ml de vinaigre que l'on introduit dans une fiole jaugée de 100 ml. On remplit au $\frac{3}{4}$ d'eau distillée. On homogénéise et on complète jusqu'au trait de jauge.

On effectue un dosage de 50 ml de la solution obtenue par la soude de concentration $C_b=0.5 \text{ M}$, au cours de l'opération de titration on a pris quelques valeurs (pH_i , V_i) qui sont données dans le tableau suivant :

V_b	0	0,5	1,0	1,5	2,0	2,5	3,0	3,5	4,5	5,0	6,5	7,0	7,1	7,2	7,3	8,0
pH	3,21	3,63	3,95	4,14	4,3	4,46	4,58	4,74	4,94	5,11	5,75	7,6	8,0	8,8	9,65	10,58

Tableau 1.2 Valeurs de titration de l'acide éthanóique contenu dans le vinaigre

c) Les résultats expérimentaux :

Point d'équivalence :

D'après la méthode de Gradient, on obtient le point d'équivalence de coordonnées (8.8, 7.2):

$$\text{pH}_{\text{eq}} = 8,8$$

$$V_{\text{eq}} = 7.2 \text{ ml}$$

Détermination du pKa du couple acide éthanóique/ion éthanóate :

A la demi équivalence nous avons directement $\text{pKa} = 4,74$. A ce niveau, la moitié de la quantité initiale des molécules $\text{CH}_3 \text{COOH}$ a été transformée en ions CH_3COO^- et l'on peut écrire :

$$\begin{aligned} n(\text{CH}_3\text{COO}^-) &= \frac{1}{2} n(\text{CH}_3\text{COOH}) \\ n(\text{CH}_3\text{COOH})_r &= \frac{1}{2} n(\text{CH}_3\text{COOH}) \end{aligned} \quad (1.35)$$

D'où $n(\text{CH}_3\text{COO}^-) = n(\text{CH}_3\text{COOH})$

Donc $[\text{CH}_3\text{COOH}] = [\text{CH}_3\text{COO}^-]$ et :

$$\text{pH} = \text{pKa} + \log \frac{[\text{CH}_3\text{COO}^-]}{[\text{CH}_3\text{COOH}]} = \text{pKa} + \log 1 = \text{pKa} \quad (1.36)$$

Donc à la demi équivalence : $\text{pH}_{1/2} = \text{pKa} = 4.74$

d) Vérification de résultats par la méthode de Gran :

D'après la méthode de Gran, on obtient :

Le volume à l'équivalence est 7.2009 mL, et le $pK_a = 4.7999$

On observe bien que les deux méthodes donnent des résultats très voisins.

e) Comparaison des deux méthodes expérimentales :

	Méthode du gradient	Méthode du Gran
V_e	7,20 ml	7,20 ml
pKa	4,76	4.79

Tableau 1.3 Résultats d'analyse de l'acide éthanique contenu dans le vinaigre par la méthode de Gran et du Gradient

f) Exploitation des résultats :

Vérification du degré d'acidité du vinaigre :

$$n_o(\text{CH}_3\text{COOH}) = n(\text{OH})_{\text{versé}}$$

$$C_a \times V_a = C_b \times V_b$$

$$C_a = C_b \times V_{\text{équivalent}} / V_a$$

$$C_a = 0,50 \times 7.2 \times 10^{-3} / 50 \times 10^{-3} = 0,072 \text{ mol} \times \text{L}^{-1}$$

Dans la solution dosée la concentration est de $0,072 \text{ mol.l}^{-1}$

Pour obtenir cette solution S2, on a réalisé une dilution à partir d'une solution S1 de vinaigre.

Lors d'une dilution, les quantités de matière restent inchangées :

$$n_1 = n_2$$

$$C_1 \times V_1 = C_2 \times V_2$$

$$C_2 = C_1 \times V_1 / V_2$$

Donc :

$$C_2 = 0,072 \times 100 / 10 = 0.72 \text{ mol.l}^{-1}$$

Calcul de la masse d'acide :

$$n^{\circ}_{(CH_3COOH)} = C \times V_p = 0.72 \times 10^{-3} = 0.72 \times 10^{-2} \quad \text{où } V_p : \text{ volume prélevé}$$

$$\text{Donc } n^{\circ}_{(CH_3COOH)} = M \times n = 0.72 \times 10^{-2} \times 60 = 0,432\text{g}$$

Calcul de la masse de vinaigre :

$$\text{On sait que : } \varphi = \frac{n_{\text{vinaigre}}}{V_p} = 1 \text{ g} \cdot \text{mL}^{-1} \quad \text{où } V_p : \text{ volume prélevé}$$

$$\text{Donc } n = \varphi \times V = 1 \times 10 = 10\text{g}$$

Calcul du degré d'acidité :

On sait que 1° correspond à 1g d'acide /100g de vinaigre.

Donc :

$$d = (0.432 / 10) \times 100 = 4,32^{\circ}$$

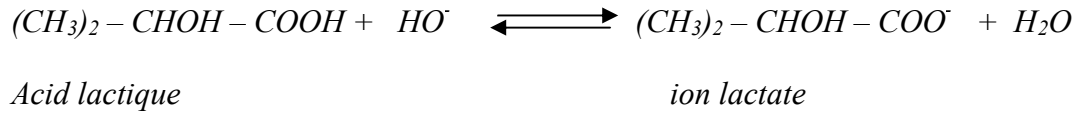
1.9.3 Dosage de l'acide lactique contenu dans le lait :

a) Principe :

La fabrication du yaourt se fait en ensemencant le lait avec des bactéries, comme *Lactobacillus bulgaricus* et *Streptococcus thermophilus*, qui fermentent, à partir du lactose (entre autres), et permettent d'acidifier le lait (production d'acide lactique par *fermentation homolactique*) et de développer les arômes. La conservation du lait est liée à ce phénomène, puisque le caillage correspond également à une fermentation par acidification, d'ailleurs utilisé pour certains yaourts.

Nous voulons étudier les conditions de cette fermentation et son activité (plus ou moins efficace) : nous pourrions le constater en mesurant l'acidification du lait. La technique de dosage par une solution d'hydroxyde de sodium est très utilisée dans les laiteries. De surcroît, elle est assez simple à réaliser. Détaillons les modalités de cette manipulation.

L'expérience consiste en une réaction acido-basique entre d'une part, les produits de la fermentation, et d'autre part, l'hydroxyde de sodium ($\text{Na}^+ + \text{HO}^-$) (ou soude). Dans le cas d'une fermentation homolactique, avec production d'acide lactique, l'équation chimique du dosage sera :



Les ions sodium sont ici spectateurs, c'est-à-dire qu'ils ne prennent pas part à la réaction. Comme pour toute réaction acido-basique, nous sommes en présence d'un couple forme acide/forme basique : l'ion lactate est alors une base. Par conséquent, lorsque l'on verse de la soude (des ions HO⁻), le pH du mélange augmente, tout en restant inférieur à 7 à cause de l'acide lactique encore présent. Pour neutraliser tout l'acide, il faut verser un volume de soude V, appelé volume à l'équivalence. Si l'on rajoute encore de la soude, le pH deviendra supérieur à 7.

b) Les résultats expérimentaux :

On effectue un dosage de 50 ml du lait de vache par la soude de concentration C_b=0.5 M, au cours de l'opération de titration on prend quelques valeurs (pH_i, V_i) qui sont données dans le tableau suivant :

V _b	0	0.5	1.0	1.5	2.0	2.5	2.8	2.9	3.0	3.3	3.4	3.5	4.0
pH	6.16	6.38	6.7	6.98	7.33	8.14	8.58	8.7	8.9	9.27	9.4	9.49	10.14

Tableau 1.4 Valeurs de titration de l'acide éthanique

Point d'équivalence :

D'après la méthode de Gradient, on obtient le point d'équivalence de coordonnées (8.9, 3.0) :

$$pH_e = 8,9$$

$$V_e = 3,0 \text{ ml}$$

c) Vérification de résultats par la méthode de Gran :

D'après la méthode de Gran, on obtient :

Le volume à l'équivalence est 3.0868 ml

On observe bien que les volumes obtenus par les deux méthodes sont très voisins.

d) Détermination de l'acidité du lait :

Cette opération est très simple si nous avons les coordonnées du point d'équivalence, et d'après la relation :

$$C_a \times V_a = C_b \times V_b$$

Donc : $C_a = C_b \times V_b / V_a = 3 \times 0.5 / 50 = 0.03 \text{ mol / L}$

Dans le présent travail un banc d'essai automatique pour le contrôle du pH, la dureté et le degré d'un produit a été réalisé. La chaîne d'acquisition est composée d'un capteur commercial TACUSSEL, d'une carte de traitement analogique du signal et d'un Convertisseur Analogique Numérique intégré au microcontrôleur.

Une pompe micrométrique à champ magnétique sert au titrage des produits à analyser. Le microcontrôleur mesure la valeur du pH, puis calcule les différents points de la courbe par la méthode du gradient. La valeur maximale correspond au point d'équivalence. Le banc d'essai est utilisé pour effectuer l'analyse du vinaigre commercial, du lait, de l'acide acétique et enfin la dureté de l'eau.

La comparaison des résultats obtenus par le banc d'essai et ceux fournis par la méthode de Gran montre l'efficacité, la précision de notre appareil.

2. RESEAUX DE NEURONES ARTIFICIELS POUR LA MODELISATION DES PROCESSUS CHIMIQUES

2.1 INTRODUCTION

Les avantages potentiels de la commande non linéaire sont traités par le biais de la modélisation linéaire et non linéaire et des techniques de commande. Les différentes technologies de commande du pH sont examinées et la motivation d'un système de commande basé sur le réseau de neurones est présentée.

2.1.1 Historique

Une grande partie de la théorie des systèmes de commande a été élaborée sur la base des systèmes linéaires, théorie qui est arrivée à maturité, bien assimilée et maîtrisée. Cette approche s'est avérée un succès parce que, bien que tous les systèmes physiques soient non linéaires dans une certaine mesure, la plupart des systèmes peuvent être estimés en utilisant les relations linéaires. Ceci est particulièrement vrai pour les systèmes électromécaniques où une performance satisfaisante est souvent obtenue en utilisant une commande linéaire. De nombreux systèmes chimiques, cependant, sont fortement non linéaires, en raison de leurs réactions cinétiques et leurs relations thermodynamiques. Dans ces circonstances, la conception d'une commande solide utilisant l'hypothèse de linéarité introduit un certain degré d'incertitude dans la procédure de conception. Cela, à son tour, exige généralement que la commande ne peut fonctionner efficacement qu'au sein d'une région limitée. La situation est encore compliquée par le fait que les processus chimiques sont souvent variables dans le temps et peuvent contenir des temporisations importantes. Les conséquences de la combinaison de ces facteurs signifient généralement que les processus industriels ne peuvent être efficacement exploités sous une commande linéaire.

Au cours des dernières décennies, plusieurs stratégies ont été mises au point pour améliorer les performances du système en incluant un modèle au sein de la structure de commande. Les performances de ces modèles sont dépendantes de plusieurs facteurs, et sont principalement tributaires de la validité du modèle. Le développement d'un grand nombre de ces modèles de commande (MBC) est fondé sur la théorie des systèmes linéaires, et par conséquent ils utilisent les modèles linéaires (DORF, 1992). Ainsi, la performance de ces modèles de

commande peut se détériorer pour les systèmes qui sont mal décrits par un modèle linéaire. Cela peut se manifester par une mauvaise performances, voire une instabilité, après un changement de valeurs des points ou une perturbation.

Compte tenu de la prémisse fondamentale de la stratégie MBC qui dit que si un système est bien modélisé, il peut être bien commandé, et de la discussion précédente sur le comportement non linéaire d'un grand nombre de procédés chimiques, la nécessité pour le développement d'un modèle non linéaire de processus est persuasif. Bien sûr, l'utilisation d'un modèle non linéaire ne se limite pas à l'intégration d'une structure MBC. Un modèle précis du processus est un précieux outil de simulation qui pourrait être utilisé pour valider les commandes candidates, pour prédire les conséquences sur les modifications proposées ou reconnaître, et ensuite prévenir les conditions d'exploitation cruciales des processus.

Manifestement, un modèle de processus joue un rôle central dans la conception du système de commande, que ce soit directement, comme un schéma MBC, ou indirectement, comme cela peut être le cas pour la mise au point d'un paramètre de la commande. La section suivante passe en revue les techniques de modélisation, leurs forces, leurs limites dans l'utilisation de processus de commande.

2.2 Modélisation de processus et commande

La conception de systèmes de commande utilise souvent des modèles de systèmes dynamiques. Un tel modèle décrit comment se comporte un système réel en fonction du temps et d'une ou plusieurs entrées externes. La distinction entre un modèle dynamique et un modèle statique peut être soulignée comme suit - un modèle dynamique décrit également le comportement statique d'un système, alors que l'inverse n'est pas vraie. Une simple forme d'un modèle dynamique est un modèle non paramétré, c'est une réponse d'un système physique et peut prendre la forme d'un échelon, d'une réponse à une impulsion ou d'une réponse en fréquence. Un autre type d'un modèle dynamique, qui est largement utilisé dans la conception du système de commande, est un modèle paramétré. Il s'agit d'une description mathématique qui relie les sorties du système aux entrées et au temps. Les modèles paramétrés peuvent être divisés en deux catégories distinctes – modèles théoriques (ou analytiques) et modèles expérimentaux (ou empiriques).

Les modèles théoriques sont obtenus par l'analyse du système à un niveau fondamental. Ils sont issus de relations physiques, telles que les lois de la conservation (par exemple de masse, de charge et de l'énergie), des équations d'état et des conditions aux limites (WELLSTEAD, 1979). Typiquement, un modèle théorique est composé d'équations différentielles et algébriques, et peut être décrit par une fonction de transfert pour le SISO (Single Input Single Output) linéaire. Le succès du développement d'un modèle théorique repose sur la disponibilité de bonnes informations sur le processus, et même pour les processus relativement simples et bien assimilés, la tâche peut être longue et difficile. En revanche, un modèle expérimental, qui comme son nom l'indique est dérivé d'expérimentation du système, peut être développé sans une compréhension approfondie de la science du système. Une forme commune d'un modèle expérimental est l'équation de différence. Après la spécification de la structure d'un modèle expérimental, le processus de données d'entrée/sortie (I/O) est utilisé pour calculer les paramètres de l'équation de différence en utilisant la méthode d'identification paramétrique.

Les paramètres d'un modèle théorique se rapportent directement au système physique, tandis que la relation entre les paramètres du modèle expérimental et du système physique peut être obscure. Par conséquent, les modèles expérimentaux sont communément appelés des modèles "boîte noire" car ils donnent peu de renseignements sur le fonctionnement du système physique. Un avantage du modèle expérimental est que la même technique d'identification peut être appliquée aux différents systèmes, alors que le développement d'un modèle théorique est spécifique à une application.

Il y a des situations où le choix entre un modèle théorique et expérimental est prédéterminé. Le succès de l'identification d'un modèle expérimental nécessite certaines conditions à remplir pour obtenir le processus de données d'E/S, qui peut ne pas être réalisable. Ces circonstances s'opposent à l'identification d'un modèle expérimental. En résumé, il n'y a pas de technique de modélisation qui fournit une solution idéale pour l'élaboration des modèles, mais une application individuelle souvent clarifie le choix.

Pour de nombreux processus chimiques, les dynamiques du système peuvent être complexes et comporter un degré d'incertitude, ce qui les rend plus difficile pour la modélisation théorique (LI et SLOTINE, 1989; WILLIS, 1991). Même si le coût de l'élaboration d'un modèle théorique est supportable, il est souvent inexact. En outre, les procédés chimiques

varient généralement dans le temps et adapter les paramètres d'un modèle théorique à de tels changements n'est pas évident. Pour ces raisons, et parce qu'il y a généralement un processus de données d'E/S pour la plupart des procédés chimiques, des modèles expérimentaux sont souvent une option plus réaliste.

Il faut souligner que les modèles sont toujours une approximation du comportement du système physique, et par conséquent les modèles parfaits ne sont pas réalisables. Un modèle théorique qui, à cause des hypothèses simplifiées, est une approximation grossière du véritable système, peut néanmoins être utile pour la conception du système de commande, à condition qu'il reproduise certaines caractéristiques du système physique. En effet, il peut ne pas être souhaitable d'améliorer la fidélité du modèle en affinant les hypothèses théoriques, parce que le modèle devient plus complexe (WELLSTEAD, 1979). Ainsi, la phase de validation de développement du modèle est importante dans la mesure où on décide si le modèle se rapproche du véritable système avec un degré acceptable de précision. Le concepteur ne doit plus chercher au delà d'un modèle linéaire, si celui-ci fournit une précision suffisante. En effet, la commande peut être alors conçue à l'aide de l'analyse de systèmes linéaires, qui est un puissant outil grâce au principe de la superposition qui peut être appliqué aux systèmes linéaires.

Il existe toutefois des systèmes physiques qui ne peuvent être approchés, avec un degré de précision acceptable sur la plage nécessaire de fonctionnement, par un modèle linéaire. Dans de tels cas, un système de commande élaboré à partir d'un modèle linéaire peut fonctionner d'une manière peu satisfaisante. Beaucoup de solutions possibles à ce problème ont été suggérées au cours des trois ou quatre dernières décennies, et la plupart de ces solutions ont rencontré un certain succès pour des circonstances particulières. Toutefois, une solution générique à ce problème s'est avérée plus difficile. La plupart des solutions proposées peuvent être réparties en deux approches - celles qui utilisent des modèles linéaires avec quelques modifications de la structure de commande classique, et celles qui refusent les modèles linéaires et tentent de mettre au point un système de commande non-linéaire, qui est généralement fondé sur un modèle non linéaire. Les sections suivantes examinent certaines des approches communes pour ces deux cas.

2.2.1 Techniques des systèmes linéaires

La discussion précédente a souligné l'attrait d'une conception linéaire et, par conséquent, il n'est pas surprenant d'avoir une multitude d'approches différentes pour la commande des systèmes non linéaires basée sur les modèles linéaires et/ou des commandes linéaires. Si la non linéarité du système est une fonction d'un processus de variables mesurables, et si cette fonction est connue, plusieurs commandes linéaires locales peuvent être conçues pour différentes valeurs de la variable mesurée. La commande du système est alors obtenue par la commutation entre les différentes commandes basées sur la valeur de la variable mesurée (ASTROM et WITTENMARK, 1989). Cette technique fonctionne bien si les conditions mentionnées ci-dessus sont réalisées et elle est largement utilisée dans les processus industriels. L'inconvénient de cette technique est que généralement il n'y a pas de feedback dans la logique qui choisit le modèle. Ainsi une telle conception est tributaire non seulement de la connaissance précise de la relation entre la variable mesurée et la non linéarité du système, mais aussi de la précision des modèles locaux et des variables mesurées précises - il n'y a pas de correction d'erreurs qui se produit. En outre, les commutations rapides entre les modèles peuvent provoquer une commande insuffisante.

D'autre part, l'approche largement utilisée est la commande adaptative, où les paramètres d'une commande linéaire sont adaptés pour compenser la non linéarité du système et/ou l'invariance du temps. Cet objectif peut être atteint directement, comme c'est le cas dans des projets tels que le modèle de référence de commande (ISERMANN., 1992), ou indirectement, telle l'approche de la commande auto-ajustable(WELLSTEAD et ZARROP, 1991), où les paramètres d'un modèle sont mis à jour et la commande est ensuite ajustée en utilisant un certain mécanisme d'adaptation qui repose sur le modèle. Il existe une grande variété de commandes linéaires qui peuvent être utilisés dans ces systèmes adaptatifs. L'objectif clé de tout système de commande adaptative est de suivre les modifications dans les dynamiques du système lorsque son point de fonctionnement change et compenser ces effets. Un des inconvénients de cette méthode de commande est que, lors du changement de son point de fonctionnement, l'information sur l'ancien modèle est perdue, et donc si le système revient à un point de fonctionnement antérieur le même modèle est identifié à nouveau. Les techniques de commande adaptative repose sur des données dynamiques riches de sorte que le modèle peut s'adapter rapidement lorsque le point de fonctionnement change.

La conception d'une commande robuste (FRIEDLAND, 1986) offre une alternative à l'approche adaptative. Ici les limites de l'incertitude sont associées à des paramètres d'un modèle linéaire et une commande linéaire est conçue pour stabiliser le système sur la plage des paramètres considérés. Un inconvénient de cette approche est la réduction de la bande passante du système de commande qui va de pair avec l'inclusion des paramètres d'incertitude.

Les techniques précédentes font usage des principes de la conception linéaire, et ont de nombreux domaines d'applications réussies. Chaque méthode a ses propres avantages et inconvénients. Toutefois, ces approches sont habituellement limitées de deux façons. Premièrement, la plupart de ces techniques ne sont pas adaptées aux applications fortement non linéaires. Pour les systèmes fortement non linéaires, un petit changement dans le point de fonctionnement peut entraîner une modification importante dans les dynamiques du système. Deuxièmement, la plupart de ces méthodes fonctionnent mal si les dynamiques du système sont sujet à des variations rapides, soit en raison d'un changement du point de fonctionnement, ou à cause de la variance de temps. Par exemple, avec les méthodes adaptatives, le paramètre d'estimation peut ne pas avoir le temps de converger dans de telles circonstances, Et, par conséquent, le modèle sera erroné, qui à son tour provoque une mauvaise commande.

En résumé, les techniques fondées sur la commande linéaire ne sont généralement possibles que si les dynamiques du processus ne sont pas fortement non linéaires, et si les conditions de processus ne changent pas rapidement (par rapport aux constantes de temps du système). Pour les systèmes en dehors de ce domaine, une approche non linéaire de la conception de la commande est nécessaire, et certaines approches antérieures à la présente sont examinées dans la section suivante.

2.2.2 Techniques des systèmes non linéaires

« Il n'y a pas de méthodes générales pour l'analyse et la synthèse des systèmes non linéaires » (GIBSON, 1963). Cette déclaration reste en grande partie vraie aujourd'hui, et par conséquent, la conception et l'analyse des systèmes de commande non linéaires, sont généralement beaucoup plus difficiles que celles des commandes linéaires. Diverses théories ont été élaborées pour certaines catégories de systèmes non linéaires. Ces méthodes basées sur une linéarisation ont été examinées en 2.1.3 D'autres techniques sont basées sur:

- i) la méthode de la fonction décrite
- ii) l'analyse d'état plan
- iii) les méthodes de Lyapunov

- iv) les modèles théoriques basées sur des équations différentielles non linéaires
- v) les modèles non linéaires expérimentaux
- vi) les modèles non linéaires basés sur les séries

(i)-(iii) sont des méthodes classiques, qui ont des applications limitées (par exemple OGATA, 1970; COOK, 1986; PHILLIPS et HARBOUR, 1988). En particulier, pour l'application de la méthode de la fonction décrite, le système doit être divisé en une composante linéaire et une composante non linéaire. La partie linéaire doit être suffisamment passe-bas pour atténuer toutes les harmoniques générées par un système à cycle limité. Cette méthode est généralement limitée à la détermination de la stabilité des cycles limités. L'analyse d'état plan donne la courbe de la trajectoire du système en fonction de ses états. Elle est donc limitée à établir le comportement d'un système d'ordre 2 ou 3 à partir d'un ensemble de conditions initiales. La première méthode (directe) de Lyapunov permet d'établir la stabilité locale des systèmes non linéaires. La deuxième méthode de Lyapunov (Indirectes) peut établir la stabilité globale, mais elle peut être très difficile à implémenter (FRANKLIN, 1990). En outre, les méthodes de Lyapunov donnent peu d'informations sur la performance du système. Les modèles théoriques Basés sur les équations différentielles non linéaires peuvent fournir une bonne solution non linéaire à la conception, mais ils sont difficiles à développer. Il existe plusieurs modèles dérivés de la théorie des séries fonctionnelles.

2.2.3 Motivation pour une approche de commande non linéaire en utilisant les réseaux de neurones

De la discussion précédente, il est clair que la modélisation et la commande des systèmes non linéaires n'est pas simple, et le succès a été limité à des classes restrictives de systèmes non linéaires. Le principal attrait de réseaux de neurones artificiels (RNA) dans les systèmes de commande est qu'ils offrent la possibilité d'une approche potentielle générique de

modélisation et de commande des systèmes non linéaires. Le terme de réseaux de neurones artificiels provient de la recherche qui a tenté de comprendre, et a proposé des modèles simples du fonctionnement du cerveau humain. En conséquence, les RNA possèdent les caractéristiques qui sont communes avec le système biologique - ils consistent en de nombreux éléments de traitement simples (neurones) reliés par des connexions variables (les synapses) pour former un système de traitement de l'information en parallèle (SIMPSON, 1990). Cela donne aux RNA plusieurs caractéristiques qui aident à la modélisation et à la commande des systèmes non linéaires.

2.3 Propriétés de la commande du pH :

La commande du pH est capitale à beaucoup de différents secteurs de la technologie, tels que la fermentation, la précipitation, l'oxydation, la flottaison, l'extraction par solvants, la neutralisation des eaux usagées et la fabrication des acides gras et des savons. Il est important d'apprécier la nature diverse de ces applications de commande du pH parce qu'ils changent considérablement selon le degré de difficulté, et par conséquent il est inutile de généraliser la commande du pH.

La commande du pH des eaux usagées pour quelques processus industriels peut être un problème très difficile à résoudre, particulièrement si l'usine est mal conçue.

Dans ce mémoire nous étudions la commande d'une neutralisation continue.

Le pH est le logarithme négatif de la concentration des ions hydronium $[H_3O^+]$. La courbe de titration a une forme en S et elle définit la caractéristique statique du pH d'un processus. En raison de la non-linéarité logarithmique, il est possible que le gain d'un processus de pH change d'un facteur de 10 par unité de pH.

Le schéma 2.1 illustre les différentes formes que la courbe de titration (acide simple/base simple) peut prendre. Le point où les concentrations de l'acide et de la base sont égales se nomme le point d'équivalence, et le gain de processus est à un maximum en ce point. Pour le système base forte/acide fort, le gain au point d'équivalence est extrêmement haut et il se produit à une valeur de pH = 7 (pH neutre), ce qui rend la commande difficile.

Clairement, il est plus facile de commander le système base faible/acide faible en raison du gain faible près de la neutralité. Noter que le point d'équivalence ne coïncide pas toujours avec la neutralité (par exemple figure 2.1 b et c).

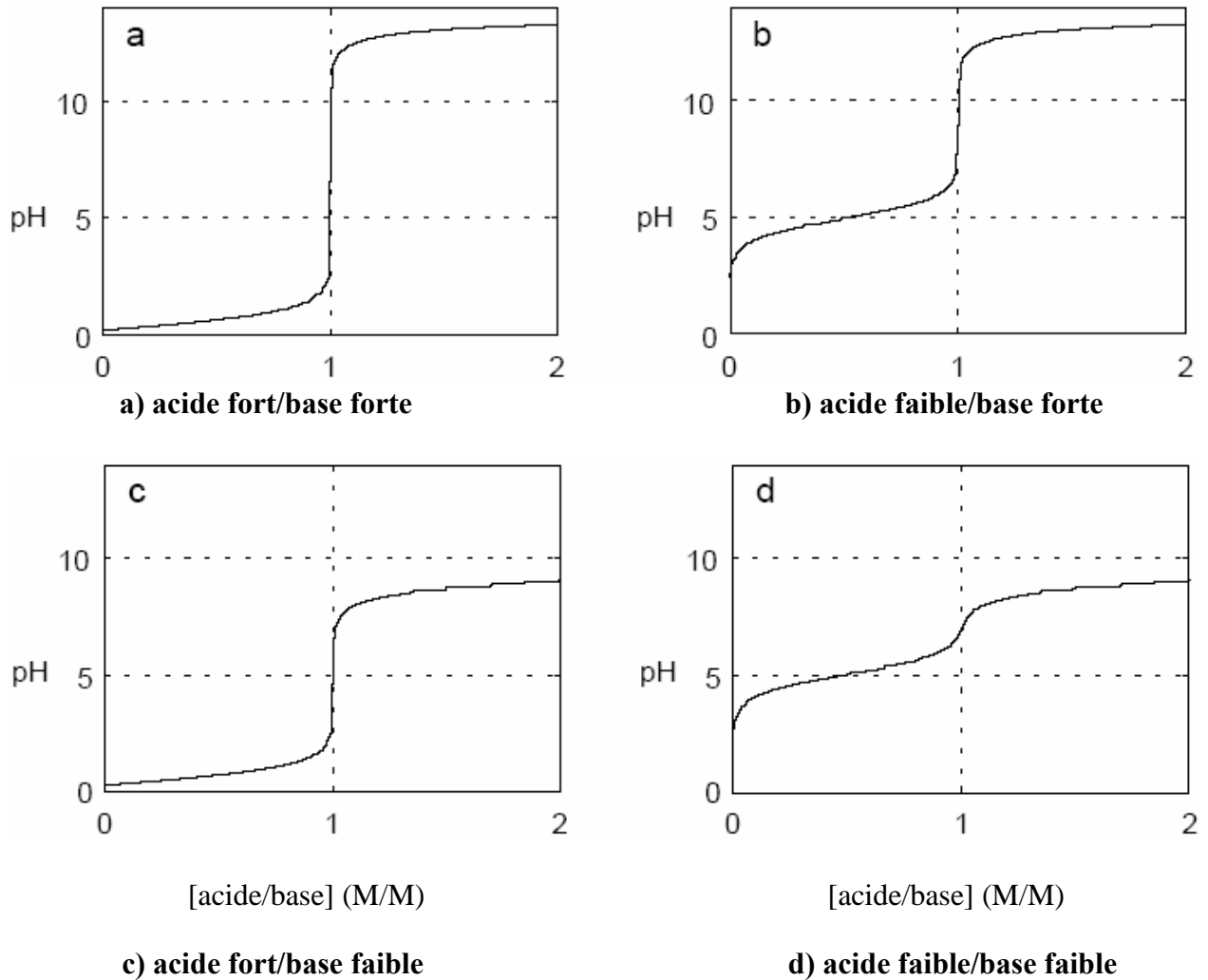


Figure 2.1 : Courbes de titration

Tandis que la non-linéarité logarithmique du pH peut être fortement prononcée, il devrait être possible de l'éliminer, puisque le rapport entre le pH et $[H^+]$ est défini. Le problème se réduirait alors à calculer le volume du réactif neutralisant à ajouter pour avoir $[H^+] = 10^{-7}$ (pH neutre). Malheureusement, pour n'importe quel système chimique réel, il n'y a pas une relation unique entre $[H^+]$ (ou pH) et le volume du réactif neutralisant.

La raison pour laquelle la relation entre le pH et le rapport "acide/base" n'est pas unique est due à l'effet tampon [4]. Le pH est seulement une mesure de la concentration des ions d'hydrogène dissociés présents dans une solution. Quand une solution contient un acide faible, la plupart des ions d'hydrogène ne sont pas dissociés (ce raisonnement peut être appliqué à une base faible en considérant les ions d'hydroxyde (OH^-) au lieu des ions d'hydrogène). Une fois titrés avec une base, les ions d'hydrogène non dissociés se dissocient et se combinent avec les ions d'hydroxyde. L'effet sur le pH de la solution n'apparaît que lorsque tous les ions d'hydrogène non dissociés sont consommés. Par conséquent, la solution ne réagit pas rapidement aux changements de pH. L'effet tampon (buffering) explique la contradiction apparente d'une solution contenant un acide faible exigeant un volume plus grand de la base pour changer son pH de 3 à 7 qu'une solution contenant seulement un acide fort (un acide fort est 100% dissocié et par conséquent n'a aucun pouvoir tampon).

Puisque le pH n'est pas une mesure interprétable de l'état d'une solution, il ne peut être utilisé seul pour la commande. C'est la courbe de titration qui définit uniquement la caractéristique statique d'un système de pH, et par conséquent la plupart des stratégies de commande de pH sont basées sur la courbe de la titration.

2.3.1 Une approche de commande du pH par RNA

Utiliser les RNA pour commander des systèmes non linéaires est justifié par leurs avantages par rapport à d'autres méthodes de modélisation des systèmes non linéaires. Un processus de pH peut avoir des caractéristiques, y compris la non linéarité, ce qui rend difficile la commande. En conséquence, une grande partie de l'effort de recherche a été appliquée à la commande du pH et de nombreuses différentes stratégies de commande ont été proposées. Récemment, plusieurs chercheurs ont appliqué le modèle de commande, qui emploie des modèles empiriques pour la commande du pH. PROLL et PR KARIM (1994) ont appliqué le modèle pour une commande prédictive en utilisant avec un modèle NARX à un processus de neutralisation des eaux usées. POTTMANN et SEBORG (1997) a utilisé un algorithme de commande prédictive basée sur un modèle de réseau à neurones à un processus de neutralisation de pH. Ces études ont confirmé la faisabilité de commander le processus du pH à l'aide de réseaux de neurones. De plus, certaines de ces études ont montré une amélioration dans les performances de commande par rapport aux commandes linéaires conventionnelles.

D'où la motivation pour étudier la commande à base de réseaux de neurones du processus pH possédant des caractéristiques qui le rendent difficile à commander.

2.4 RESEAUX DE NEURONES ARTIFICIELS

On introduit ici certains types de RNA couramment utilisés. Le perceptron à multicouches est passé en revue. La structure du modèle NARX est présentée comme une méthode souple d'utilisation d'un RNA non bouclé, comme un modèle dynamique non linéaire. Les paramètres d'entrée du neurone, la parcimonie des RNA et les techniques de validation du modèle sont discutées.

2.4.1 RNA non bouclés

Il existe de nombreux types de réseaux de neurones et ce travail a été limité au RNA non bouclé (feedforward), le perceptron multicouches (MLP) qui a été utilisé comme modèle des processus non linéaires. Cette section présente certaines caractéristiques du réseau et mentionne certains d'autres types de RNA.

Les RNA non bouclés sont composés de deux ou plusieurs couches d'éléments de traitement reliés par des connexions pondérées (Figure 4.1). Le réseau opère en deux phases appelées apprentissage et validation. Pendant la phase l'apprentissage, un ensemble de données (apprentissage) est continuellement présenté au RNA, qui traite les données de chaque vecteur en fonction de son architecture et ajuste ses poids, et dans certains cas il ajuste également certains paramètres de traitement selon la règle d'apprentissage. Le but de la règle d'apprentissage est d'adapter les poids du réseau afin que celui-ci traite les données d'apprentissage pour donner un résultat spécifique, qui est généralement la minimisation de la fonction des données d'apprentissage.

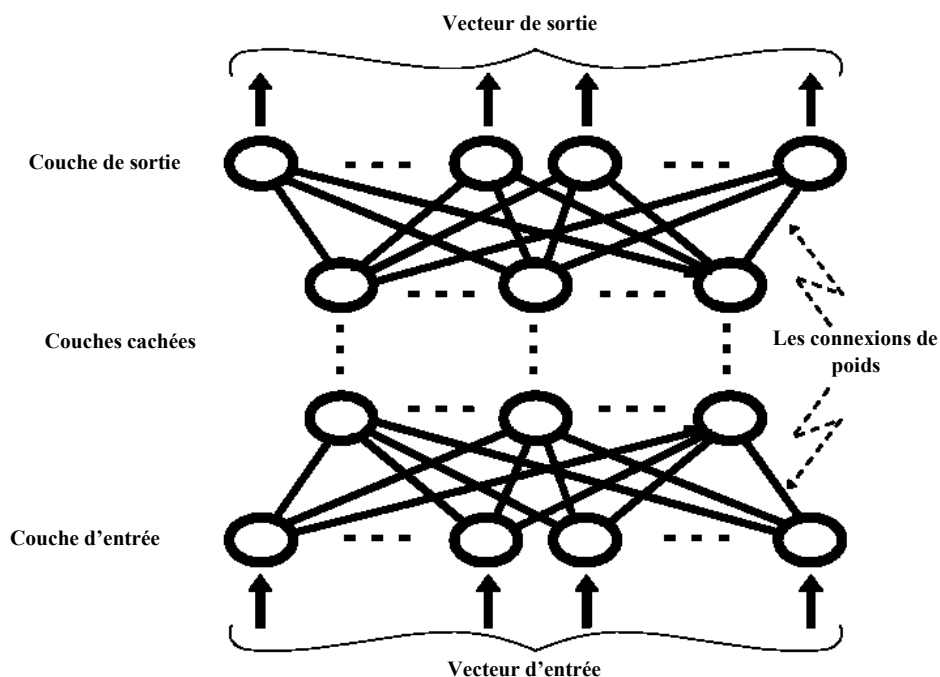


Figure 2.2 Les réseaux de neurones non bouclés

Après l'apprentissage le réseau peut être utilisé dans le mode de validation où les poids du réseau sont fixes et il est testé avec un seul ensemble de données. Cette vérification ou validation du réseau est un pas très important dans le cycle de développement d'un modèle RNA non linéaire. L'objectif principal est de développer un réseau qui peut émuler le système qui a fourni les données d'apprentissage.

Les réseaux MLP sont des réseaux non bouclés, ce qui signifie que, lorsqu'ils opèrent en mode de validation, le flux d'informations à travers le réseau est entièrement dirigé vers l'avant, des entrées vers les sorties. (Figure 2.2). La classe inverse est dite récurrente ou réseaux de neurones à rétroaction (par exemple, le réseau Elman - ELMAN, 1990), où, au cours de la phase de validation, la sortie d'au moins un élément du réseau est ramenée à l'entrée, généralement après un certain traitement, en tant qu'entrée à un autre élément du réseau. Même s'il y a eu quelques applications de commande employant les RNA récurrents, la majorité ont impliqué les réseaux non bouclés. Ceci est probablement dû à la capacité prouvée d'approximation de certains réseaux non bouclés et le comportement dynamique complexe des réseaux récurrents (BROWN et HARRIS, 1994).

2.4.2 Le perceptron multicouches (MLP)

Le perceptron multicouches est un réseau non bouclé, qui est formé au moyen d'une procédure d'apprentissage supervisé.

Les RNA MLP ont été amplement étudiés et utilisés avec succès dans diverses applications, telles que la reconnaissance des formes (MAH et CHAKRAVARTHY, 1992), la détection des anomalies (HOSKINS et HIMMELBLAU, 1988), la prévision (FOSTER et coll., 1990) et l'interprétation du langage naturel (SELMAN, 1989). Le succès du RNA MLP est dû à sa capacité multidimensionnelle et non linéaire d'approximation. Il a été prouvé (CYBENKO, 1989; FUNAHASHI, 1989) qu'avec un nombre de neurones cachés suffisant, un RNA MLP incorporant une couche cachée peut approcher toute fonction continue non linéaire avec un degré de précision arbitraire.

2.4.2.1 Le Perceptron

L'unité élémentaire de traitement pour un RNA non bouclé est dénommée neurone. Le MLP, à l'instar de nombreux réseaux de neurones, est fondé sur le modèle de neurone McCULLOCH - PITTS (1943). L'original neurone de McCulloch et Pitts consiste en une somme linéaire de ses entrées activées par une fonction à seuil binaire. Ce modèle (Figure 2.3) constitue la base de la plupart des architectures de réseaux de neurones, bien que la fonction d'activation soit souvent continue au lieu d'être binaire.

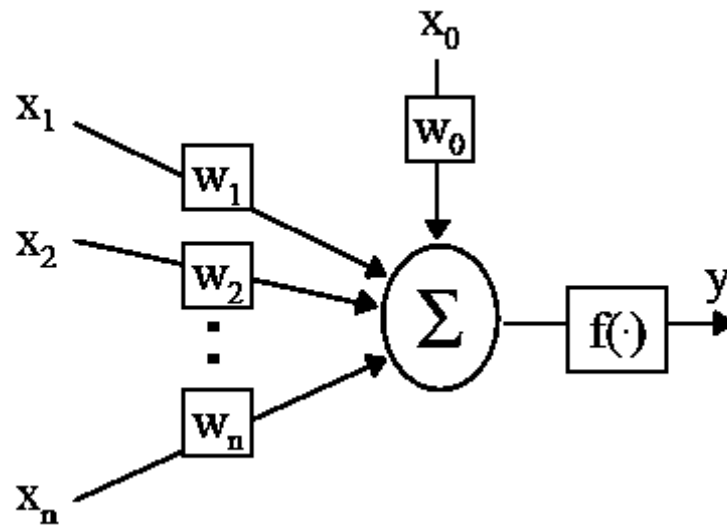


Figure 2.3 Le neurone de McCulloch et Pitts

Le neurone de McCulloch et Pitts calcule le produit scalaire d'un vecteur d'entrée $\mathbf{x} = [x_0 \ x_1 \dots \ x_n]$ et un vecteur de poids $\mathbf{w} = [w_0 \ w_1 \dots \ w_n]$, où w_0 est un biais d'un poids fixe d'entrée ($x_0 = 1$), et transmet le résultat à travers une fonction d'activation $f(\cdot)$...

$$y = f(\mathbf{x} \cdot \mathbf{w}) = f\left(\sum_{i=0}^n w_i x_i\right) \quad (2.1)$$

La fonction $f(\cdot)$ doit être non linéaire, si le RNA MLP doit avoir une capacité d'approximation non linéaire et elle doit être une fonction continue dérivable pour que le réseau soit un approximateur universel (CYBENKO, 1989). Le choix de $f(\cdot)$ doit inclure et les fonctions Gaussienne et sinus, mais la sigmoïde et la tangente hyperbolique sont plus largement utilisées. La figure 2.4 illustre le fait que la sigmoïde et la tangente hyperbolique sont étroitement liées. Elles ont toutes les deux un seuil monotone croissant. Seulement la sigmoïde est une fonction positive alors que la tangente hyperbolique a une moyenne nulle.

Pour les deux fonctions la nature de leur monotonie signifie que l'effet d'une entrée pondéré est facilement interprétable (c'est-à-dire $w_i > 0$ est excitatrice, $w_i < 0$ est inhibitrice, $w_i = 0$ ne donne aucune contribution). Du point de vu calcul la sigmoïde est plus séduisante parce que sa dérivés a une forme simple.

Enfin, plusieurs types de fonction d'activation peuvent être combinés dans un RNA MLP. L'exemple le plus courant est l'utilisation d'une fonction d'activation linéaire dans les neurones de sortie (BISHOP, 1995).

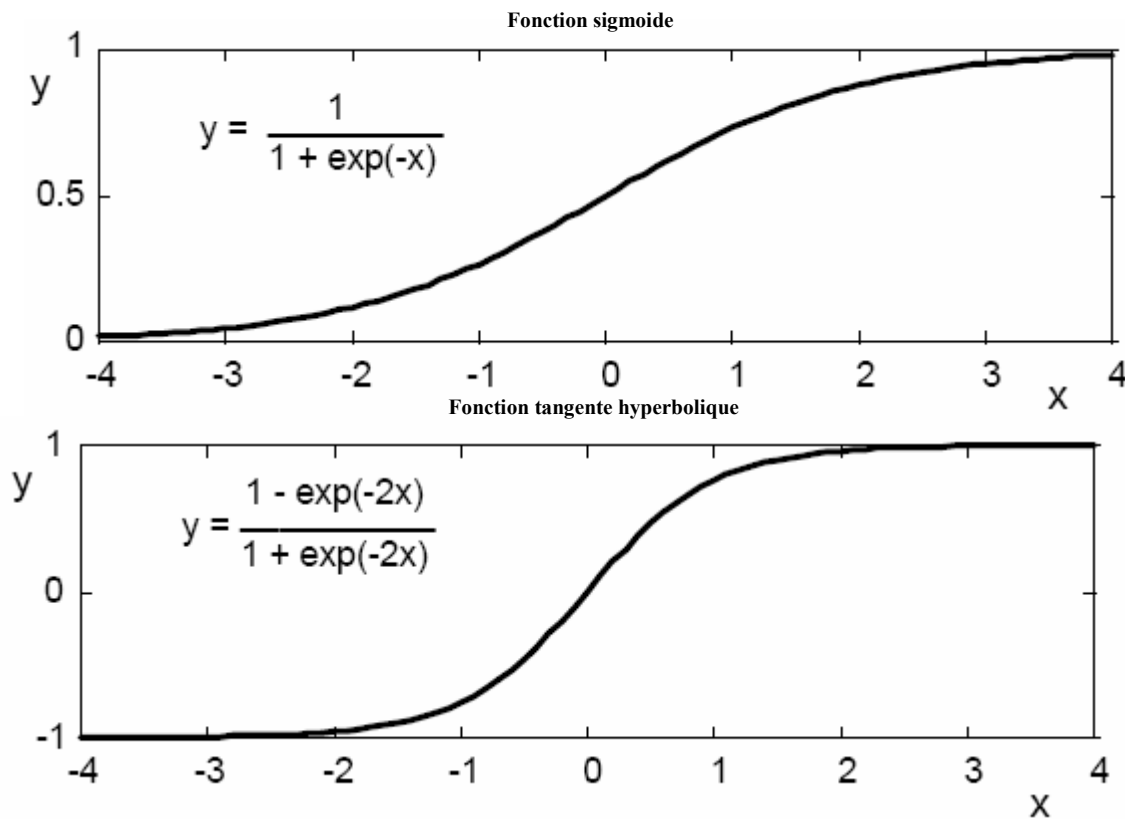


Figure 2.4 Les fonctions d'activation sigmoïde et tangente hyperbolique

2.4.2.2 Apprentissage d'un MLP RNA

Les MLP sont formés en utilisant une règle d'apprentissage supervisée, qui tente de minimiser l'erreur entre les sorties du réseau et la cible. Si les sorties de la cible ne sont pas nécessaires, alors la règle d'apprentissage n'est pas supervisée et le réseau extrait ses propres caractéristiques de l'ensemble d'apprentissage. Pour les applications en commande basées sur un modèle identifié, le réseau de neurones est généralement formé en utilisant une procédure d'apprentissage supervisée.

Le but de l'algorithme d'apprentissage est de permettre au RNA à estimer les paramètres des neurones du réseau afin que celui-ci remplisse au mieux la tâche qui lui est affectée. Pour atteindre cet objectif, l'algorithme tente de minimiser la fonction de coût des moindres carrés (somme des carrés des différences entre les valeurs calculées par le réseau de neurones et les valeurs mesurées) en ajustant les paramètres de poids du RNA. La fonction de coût permet de dire comment le RNA assimile les données d'apprentissage que le système a produites.

Par souci de simplicité, l'algorithme de rétro propagation (BP) décrit ici est pour un RNA de trois couches, puisque la généralisation de l'algorithme à des couches supplémentaires est simple. La figure 2.4 montre un tel réseau où x_i est la $i^{\text{ème}}$ entrée du réseau, h_j est la sortie, ou l'activation du $j^{\text{ème}}$ neurone caché, \hat{y}_k est la $k^{\text{ème}}$ sortie du réseau, w_{ij}^1 est le poids reliant le $i^{\text{ème}}$ neurone d'entrée au $j^{\text{ème}}$ neurone caché, w_{jk}^2 est le poids reliant le $j^{\text{ème}}$ neurone caché au $k^{\text{ème}}$ neurone de sortie, n_i est le nombre d'entrées du réseau, n_h est le nombre de neurones cachés, n_0 est le nombre de sorties du réseau.

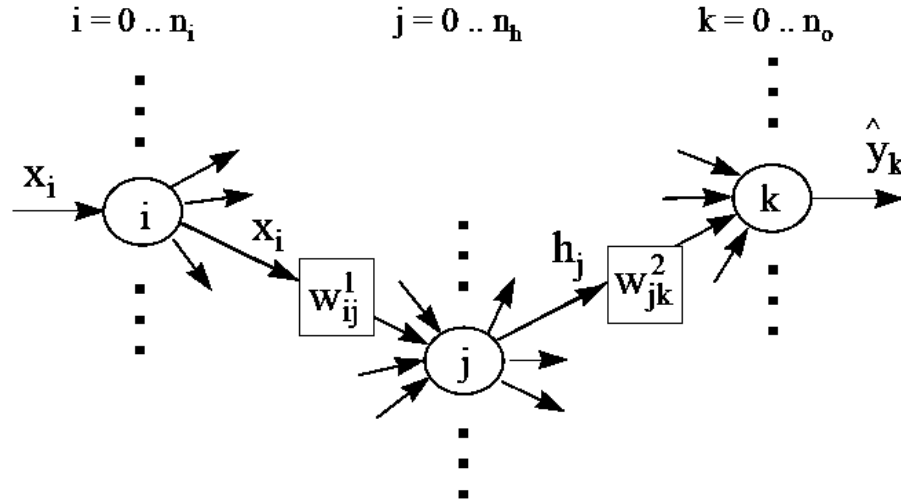


Figure 2.5 Réseau perceptron à trois couches

Les sorties des neurones cachés et des neurones de sortie sont dérivées à partir de l'équation

2.1

$$h_j = f\left(\sum_{i=0}^{n_i} w_{ij}^1 x_i\right) \quad (2.2)$$

$$\hat{y}_k = f\left(\sum_{j=0}^{n_h} w_{jk}^2 h_j\right) \quad (2.3)$$

En phase d'apprentissage, le réseau est présenté avec N vecteurs $[x(p) \ y(p)]$, où $x(p) = [x_1(p) \ x_2(p) \dots \ x_{n_i}(p)]$ est le $p^{\text{ième}}$ vecteur d'entrée et $y(p) = [y_1(p) \ y_2(p) \dots \ y_{n_o}(p)]$ est le $p^{\text{ième}}$ vecteur cible de sortie, et l'algorithme d'apprentissage tente de minimiser la fonction de coût...

$$J = \frac{1}{2} \sum_{p=1}^N \sum_{k=1}^{n_o} \left[y_k(p) - \hat{y}_k(p) \right]^2 \quad (2.4)$$

L'algorithme batch d'apprentissage ajuste les poids du réseau après que tous les vecteurs N E/S sont introduits au réseau et J est calculé, tandis qu'un algorithme continu d'apprentissage ajuste la pondération du réseau après introduction de chaque ensemble d'apprentissage tout en évaluant la valeur instantanée de la fonction de coût J_p .

$$J_p = \frac{1}{2} \sum_{k=1}^{n_o} \left[y_k(p) - \hat{y}_k(p) \right]^2 \quad (2.5)$$

BP est un algorithme de gradient descendant qui utilise le gradient négatif de la fonction de coût pour ajuster les poids du réseau. Pour la version de l'algorithme continu ces gradients sont donnés par

$$\frac{\partial J_p}{\partial w_{jk}^2} = -f'(I_k^2(p)) \left[y_k(p) - \hat{y}_k(p) \right] h_j(p) \quad (2.6)$$

$$\frac{\partial J_p}{\partial w_{ij}^1} = -f'(I_j^1(p)) \sum_{k=1}^{n_o} \left[f'(I_k^2(p)) \left[y_k(p) - \hat{y}_k(p) \right] w_{jk}^2 \right] x_i(p) \quad (2.7)$$

Où f' est la dérivée première de la fonction d'activation, $I_j^1(p)$ est l'entrée du $j^{\text{ième}}$ neurone de la couche cachée, et $I_k^2(p)$ est l'entrée de $k^{\text{ième}}$ neurone de la couche de sortie.

$$I_j^1(p) = \sum_{i=0}^{n_i} w_{ij}^1 x_i(p) \quad (2.8)$$

$$I_k^2(p) = \sum_{j=0}^{n_j} w_{jk}^2 h_j(p) \quad (2.9)$$

L'utilisation de la fonction d'activation sigmoïde est un avantage à cause de la facilité de calcul de sa première dérivée

$$f'(z) = [1 - f(z)] f(z) \quad (2.10)$$

Une fois que le gradient est trouvé, il faut descendre la surface de la fonction dans la direction où la pente est la plus forte. A la $n^{\text{ième}}$ itération de l'algorithme, l'ajustement apporté a un poids ...

$$\Delta w(n) = w(n) - w(n-1) = -\eta \frac{\partial J_p}{\partial w(n)} \quad (2.11)$$

Où w est un poids dans les deux couches et le gradient est calculé en utilisant l'équation 2.6 ou 2.7. Le coefficient η détermine l'ampleur de l'ajustement du poids dans la direction du gradient négatif. Il est appelé le taux d'apprentissage. Il est habituellement choisi dans l'intervalle $0 < \eta < 1$. Le choix du taux d'apprentissage est un compromis, puisque les valeurs

faibles donnent une convergence lente alors que les grandes valeurs permettent une convergence plus rapide avec une oscillation autour du coût minimum.

Un taux d'apprentissage infinitésimal est optimal, mais il n'est pas pratique en raison des temps d'apprentissage excessifs qui en résultent, mais ce qui est plus grave est le fait que la recherche peut aboutir à un minimum local au lieu d'un minimum global. RUMELHART et coll. (1986) a suggéré de modifier la mise à jour des poids (équation 2.11) en ajoutant un terme momentum, qui est une fraction de la dernière mise à jour du poids

$$\Delta w(n) = -\eta \frac{\partial J_p}{\partial w(n)} + \alpha \Delta w(n-1) \quad (2.12)$$

Le coefficient momentum α est habituellement choisi dans l'intervalle $0 < \alpha < 1$

2.4.2.3 RNA MLP et rétro propagation

La version continue de BP avec momentum (équation 2.12) est largement utilisée. Trois couches ont été utilisées pour former le RNA MLP de cette recherche.

Les valeurs initiales des poids du réseau ont été choisies de manière aléatoire à partir d'une distribution uniforme dans l'intervalle $[-0.1, 0.1]$, (par exemple FAUSETT, 1994). Au cours de l'apprentissage un ensemble de valeurs E/S, choisi au hasard parmi le jeu d'ensembles d'apprentissage, est introduit.

L'ensemble des vecteurs d'apprentissage pris de manière séquentielle peuvent être similaires, comme c'est généralement le cas pour les données échantillonnées d'un processus réel.

Dans ce cas, il est préférable d'introduire des vecteurs successifs pris dans différentes régions de la zone non linéaire de sorte que leur séquence contienne plus d'information globale.

On réduit linéairement l'échelle de chaque élément d'un vecteur E / S de données

$v = [x(p) \ y(p)]$ avant l'apprentissage.

$$v_s = \left(\frac{0.9 - 0.1}{v_{\max} - v_{\min}} \right) v + \left(\frac{0.1 v_{\max} - 0.9 v_{\min}}{v_{\max} - v_{\min}} \right) \quad (2.13)$$

Où v_{\max} et v_{\min} sont le maximum et le minimum de tous les vecteurs d'apprentissage ou d'une

validation anticipée. Après avoir validé le réseau, la sortie réelle est ensuite obtenue en utilisant l'inverse de l'équation 2.13.

Dans ce travail, on a pris $\eta = 0.9$ et $\alpha = 0.6$ pour accélérer la phase d'apprentissage. Pendant l'apprentissage, on diminue exponentiellement vers zéro les deux coefficients en utilisant une constante exponentielle de temps de 200 époques.

BP tente de minimiser la fonction de coût de l'équation 2.4, et, par conséquent, au fur et à mesure que l'apprentissage progresse il y a une continue réduction de la fonction de coût.

$$MSE = \frac{1}{N \times n_0} \sum_{p=1}^N \sum_{k=1}^{n_0} \left[y_k(p) - \hat{y}_k(p) \right]^2 \quad (2.14)$$

Pour tester la généralisation de la capacité du réseau l'erreur MSE pour la validation d'un ensemble devrait être évaluée à intervalles réguliers au cours de l'apprentissage (par exemple, chaque époque). Cette MSE de test, contrairement à MSE d'apprentissage, a généralement un point de réglage minimum (Figure 2.5) où on doit arrêter l'apprentissage. Si l'apprentissage est poursuivi après ce point, le RNA rate les données l'apprentissage ce qui limite sa capacité de généralisation.

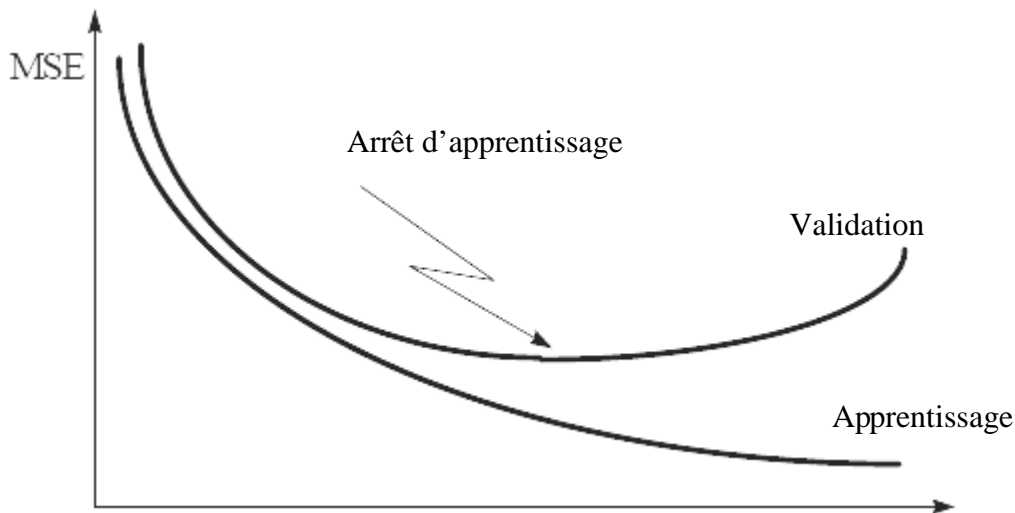


Figure 2.6 Arrêt de l'apprentissage d'un réseau MLP

2.5 MODELISATION DE PROCESSUS UTILISANT LES RNA NON BOUCLES

Les réseaux RNA non bouclés évoqués ici effectuent un mapping statique non linéaire d'un espace d'entrée à un espace de sortie. Si le réseau doit modéliser un processus dynamique, il doit y avoir un mécanisme qui lui permet de représenter les signaux temporels

et il y a deux principales approches à cet égard. La première méthode utilise un réseau neuronal récurrent qui introduit des dynamiques dans l'architecture du réseau en créant un feedback des sorties de tous les neurones ou une partie, via une connexion pondérée, à l'entrée de tous les neurones ou une partie. Ainsi, l'architecture du réseau est modifiée de sorte qu'il n'est plus un simple réseau non bouclé. La deuxième approche, appelée la méthode de la ligne à retard, utilise les valeurs actuelles et les valeurs précédentes des entrées et sorties du système comme entrées pour un réseau non bouclé standard, et le réseau est formé pour prédire la prochaine valeur de la sortie du système. Ainsi, le problème de modélisation temporelle, qui est l'apprentissage du comportement dynamique du système dans le domaine temps, se transforme en un problème de modélisation spatiale, qui est l'apprentissage du mapping à partir des entrées sorties retardées du système à la sortie suivante du système.

L'approche de la ligne à retard est la plus largement adoptée, principalement en raison de sa simplicité d'implémentation en utilisant, les architectures de réseau et les algorithmes d'apprentissage. Par conséquent, cette méthode a été appliquée à cette recherche en utilisant le modèle de structure NARX décrit dans la section suivante.

2.5.1 Le modèle de structure NARX

Beaucoup de systèmes non linéaires stochastiques peuvent être décrits en temps discret par le modèle non linéaire AutoRegressive Moving Average avec des entrées eXogenous (NARMAX) qui a été introduit pour la première fois par BILLINGS et LEONTARITIS (1981) et rigoureusement déduit par LEONTARITIS et BILLINGS (1985). Le NARMAX modèle est une généralisation de l'ARMAX linéaire et le cas SISO est...

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-k), \dots, u(t-k-n_u+1), e(t-1), \dots, e(t-n_e)) + e(t) \quad (2.15)$$

Où $y(t)$ et $u(t)$ sont l'entrée et la sortie échantillonnées au temps t du système, $f(\cdot)$ est une certaine fonction non linéaire, $e(t)$ est l'erreur de prédiction, k est le deadtime du système ($k \geq 1$) et n_u , n_y et n_e sont le nombre d'entrées retardées du système, les sorties et les erreurs de prédiction incluses dans le modèle. Le modèle NARX (LEONTARITIS et BILLINGS, 1985) est un cas spécial du modèle NARMAX avec $n_e=0$.

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-k), \dots, u(t-k-n_u+1)) + e(t) \quad (2.16)$$

La différence entre la structure des modèles NARMAX et NARX est que NARMAX contient un modèle de bruit, alors que la structure NARX suppose que le bruit (c'est-à-dire

$e(t)$ est additif à la sortie du processus. Les données d'un processus réel contiennent toujours du bruit et, si le processus est non linéaire, le bruit pénètre dans le système d'une manière non linéaire et, par conséquent, les entrées et sorties du système sont affectées. En conséquence, la structure du modèle NARMAX est plus adaptée que celle du modèle NARX pour tenir compte de ces effets non linéaires du bruit sur la sortie du processus (LEONTARITIS et BILLINGS, 1985). D'autre part, la structure du modèle NARX peut souvent mener à un modèle parcimonieux.

Il est plus simple à implémenter que le NARMAX. Et il a été utilisé avec succès pour la modélisation des processus non linéaires utilisant les RNA (par exemple, et CHEN BILLINGS, 1992; POTTMANN et SEBORG, 1992a, b; BHAT et McAVOY, 1992).

2.5.2 Implémentation d'un modèle de processus RNA de structure NARX

Il y a plusieurs phases dans la conception d'un modèle de processus RNA de structure NARX. Chaque phase joue un rôle dans la manière dont le modèle se rapproche des dynamiques du système:

A. Collecte de données. Les entrées du processus subissent des perturbations. Le signal de perturbation ou d'excitation doit être conçu de telle sorte que les données d'E/S contiennent des informations sur toutes les dynamiques du processus et soient échantillonnées à un taux approprié à l'identification du modèle RNA.

B. Structure du réseau. Le type de RNA doit être choisi (MLP) et sa topologie spécifiée. La taille des couches d'entrée et de sortie du réseau sera influencée par le nombre d'entrées et sorties du processus, et par l'ordre des dynamiques du processus.

C. Identification. L'apprentissage du modèle RNA est fait à l'aide d'un algorithme bien choisi. On doit spécifier la longueur des données d'apprentissage et les paramètres de l'algorithme.

D. Validation. La capacité du modèle RNA d'émuler la dynamique du processus est constatée et si elle est jugée insuffisante certaines de ces étapes sont répétées.

2.5.3 Configuration d'un RNA non bouclé dans une structure NARX

La Figure 2.7 montre comment un RNA non bouclé peut être configuré comme un modèle de structure NARX, où n_u sont les entrées retardées et n_y les sorties retardées utilisées comme entrées du réseau et la sortie du réseau est la sortie prédite du processus pour le prochain

échantillon de temps. La figure 2.7 montre aussi que deux différentes configurations de réseau peuvent être formées en changeant la position de l'interrupteur. Pour les deux configurations une fenêtre des entrées retardées du processus est utilisée comme entrées pour le réseau. Dans la position A, une fenêtre des sorties du processus est aussi utilisée comme entrées pour le réseau, et cette configuration est dénommée prédicteur...

$$\hat{y}(t) = f(y(t-1), \dots, y(t-n_y), u(t-k), \dots, u(t-k-n_u+1)) \quad (2.17)$$

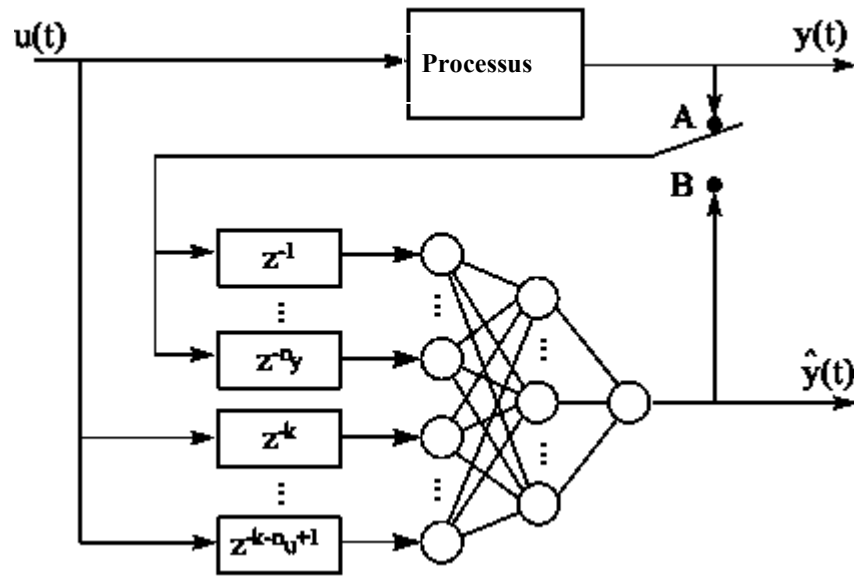


Figure 2.7 le réseau de neurones configuré comme un NARX modèle

Avec le commutateur en position B le réseau utilise ses propres sorties, plutôt que celles du processus, comme entrées et cette configuration est dénommée modèle...

$$\hat{y}(t) = f(\hat{y}(t-1), \dots, \hat{y}(t-n_y), u(t-k), \dots, u(t-k-n_u+1)) \quad (2.18)$$

NARENDRA et PARTHASARATHY (1990) a étudié la structure du prédicteur, et ils l'ont appelé le modèle série parallèle, et la structure du modèle a été appelée le modèle parallèle. La différence entre les structures prédicteur et modèle est l'utilisation du feedback. Lorsque le réseau est configuré comme prédicteur et validé l'information va dans le sens entrées vers sorties du réseau et, partant, le réseau est vraiment non bouclé. Pour valider un réseau configuré en modèle, il y a un feedback de la couche de sortie vers la couche d'entrée

et par conséquent, le réseau est récurrent externe. Un apprentissage récurrent doit être appliqué à la configuration modèle, ce qui pose problème. Généralement l'apprentissage est effectué sur une configuration prédictive et validé avec une configuration modèle.

La motivation pour valider un réseau configuré en modèle est d'obtenir une longue série de prédictions pour la sortie du processus de production, qui peut être nécessaire dans certaines circonstances. Par exemple, il peut être souhaitable d'utiliser le modèle indépendamment du processus de façon à ce que le contrôleur pourrait être ajusté en mode simulation. Plusieurs étapes en avant de la prédiction peuvent aussi être nécessaires pour implémenter des schémas de commande neuronale prédictive. Compte tenu d'une condition initiale où les données E/S courantes et passées du processus et les données futures du processus sont connues, le prédictive peut prédire seulement la prochaine valeur de sortie du processus, alors qu'un modèle peut prédire la trajectoire des sorties du processus, plusieurs pas dans le futur. Le réseau de neurones réalise P pas de prédiction consécutifs en avant, où la i-ème prédiction est de la forme

$$\hat{y}(t+i) = f\left(\hat{y}(t+i-1), \dots, \hat{y}(t+i-n_y), u(t+i-k), \dots, u(t+i-k-n_u+1)\right) \quad (2.19)$$

Parce que chacun de ces pas d'avance de prédiction contient une erreur de prédiction, cette opération récursive peut entraîner une accumulation de ces erreurs et par conséquent une dégradation de la précision du modèle. Par conséquent, il est possible qu'un réseau puisse bien fonctionner quand il est validé comme prédictive, mais lorsqu'il est validé comme modèle, les multiples étapes de prédiction rendent le modèle pratiquement inutilisable.

Il est donc important, si un RNA est utilisé comme modèle, d'effectuer P prédictions en avant, de valider le modèle RNA, en utilisant un ensemble de données de test sur un horizon de prédiction de p étapes. Ce problème devrait être examiné pendant la phase d'apprentissage. Ainsi, lorsque le RNA est destiné à être utilisé pour prédire sur un horizon de p étapes, l'apprentissage devrait être terminé lorsque la fonction de coût a un minimum pour p étapes.

3. MODELISATION DU pH D'UN CSTR A L'AIDE DE RESEAUX DE NEURONES

3.1 INTRODUCTION

Dans la conception de modélisation d'un processus non linéaire à l'aide des réseaux de neurones on doit prendre en considération ce qui suit :

1. Sélectionner le signal d'excitation du processus
2. Choisir les échantillons de temps
3. Choisir la structure du modèle
4. Valider le modèle résultant

3.2 DESCRIPTION DU PROCESSUS

Le CSTR considéré, illustré à la figure 3.1, dispose de deux entrées, l'une contenant l'hydroxyde de sodium (NaOH) et l'autre de l'acide acétique (CH_3COOH). La figure 3.2 montre que le processus a un comportement fortement non linéaire à l'état statique. Il s'agit d'un système acide faible / base forte qui présente de grandes variations au point 4 (le point d'équivalence), 150 fois celle au point 7. Ce processus a été largement étudié (par exemple, BHAT et McAVOY; 1990) pour une modélisation non linéaire utilisant une simulation analytique.

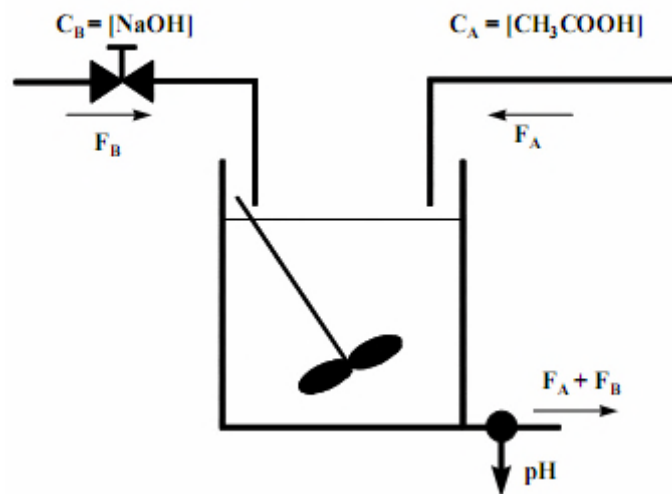


Figure 3.1 : Le processus CSTR du pH

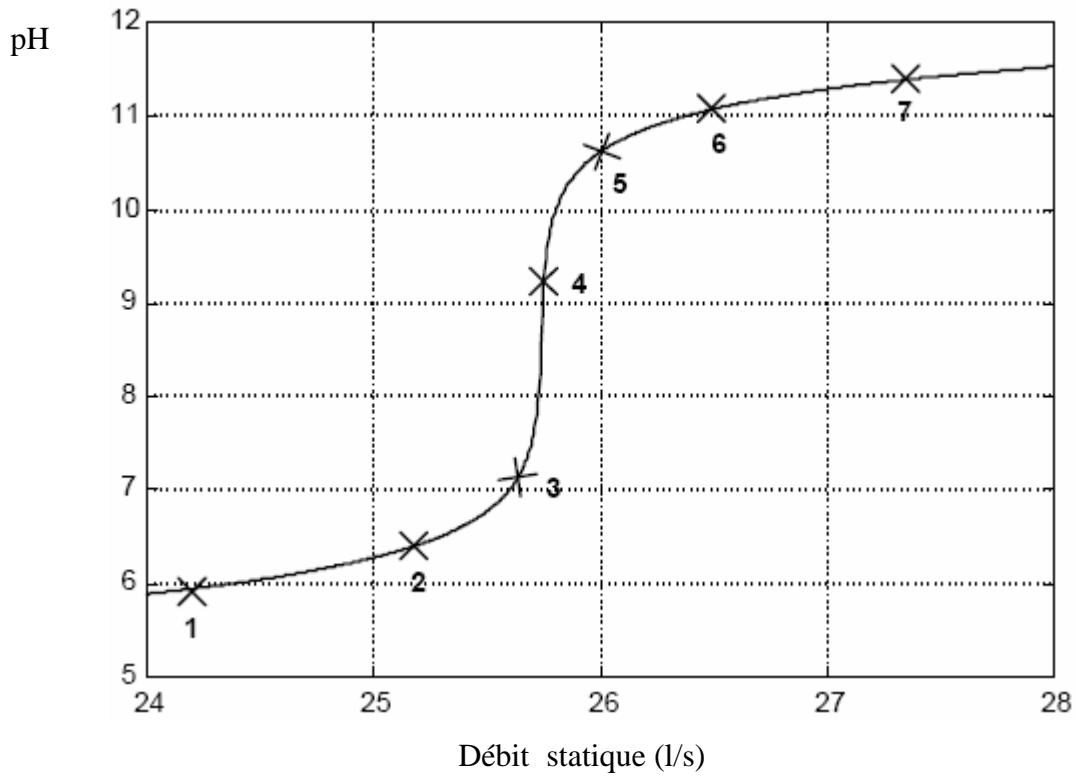


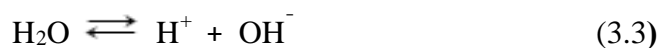
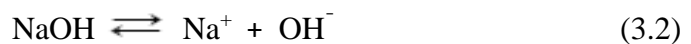
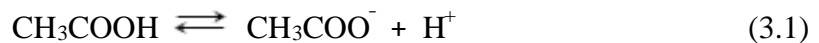
Figure 3.2 : Courbe de titration du processus CSTR

3.2.1 Le modèle mathématique du processus

Un modèle théorique de cet acide faible /base forte a été présenté par McAVOY , (1972) et a été généralisé à des systèmes avec un nombre arbitraire d'acides et de bases par GUSTAFSSON et WALLER (1983).

Le modèle est dérivé des principes élémentaires, de l'équilibre de la masse et des réactions chimiques, et il est généralement accepté dans la littérature. La méthode met en oeuvre les bilans massiques sur les composants, ou des combinaisons de composants, appelés réaction des invariants par GUSTAFSSON et WALLER (1983) des espèces ioniques de la solution CSTR. Ces invariants de réaction sont adaptés à l'équilibre de masse parce que, contrairement à la concentration des ions hydrogène, ils ne changent pas durant la réaction d'équilibre.

L'équilibre chimique du système acide faible / base forte sont :



Et les relations d'équilibre sont :

$$K_A \cdot [\text{CH}_3\text{COOH}] = [\text{CH}_3\text{COO}^-] \cdot [\text{H}^+] \quad (3.4)$$

$$K_B \cdot [\text{NaOH}] = [\text{Na}^+] \cdot [\text{OH}^-] \quad (3.5)$$

$$K_W = [\text{H}^+] \cdot [\text{OH}^-] \quad (3.6)$$

Où $[X]$ est la concentration de X dans le CSTR, K_A est la constante de dissociation du CH_3COOH , K_B est la constante de dissociation de NaOH et K_W est la constante de dissociation de l'eau. NaOH est une base forte qui se dissocie complètement (c'est-à-dire $1/K_B = 0$) et, par conséquent, l'équation 3.5 donne

$$[\text{NaOH}] = 0 \quad (3.7)$$

Les deux invariants pour les réactions de l'équation 3.1 sont la concentration totale ionique de l'acide et de la concentration totale ionique de la base ...

$$\xi = [\text{CH}_3\text{COOH}] + [\text{CH}_3\text{COO}^-] \quad (3.8)$$

$$\zeta = [\text{NaOH}] + [\text{Na}^+] \quad (3.9)$$

En plus de l'équilibre chimique de l'acide et de la base, la solution doit rester électriquement neutre, à tout moment, en donnant...

$$[\text{CH}_3\text{COO}^-] + [\text{OH}^-] = [\text{Na}^+] + [\text{H}^+] \quad (3.10)$$

Les équations algébriques 3.2... 3.8 avec 8 inconnues peuvent être réarrangées pour donner un polynôme de degré 3 en $[\text{H}^+]$ avec des inconnues ξ et ζ

$$[\text{H}^+]^3 + (K_A + \zeta) [\text{H}^+]^2 + (K_A (\zeta - \xi) - K_W) [\text{H}^+] - K_A K_W = 0 \quad (3.11)$$

Les invariants de la réaction, ζ et ξ sont calculés à partir de l'équilibre des masses :

$$F_A C_A - (F_A + F_B) \xi = V \frac{d\xi}{dt} \quad (3.12)$$

$$F_B C_B - (F_A + F_B) \zeta = V \frac{d\zeta}{dt} \quad (3.13)$$

F_A est le débit de l'acide, F_B est le débit de la base, C_A est la concentration de l'acide, C_B est la concentration de la base et V est le volume de liquide CSTR. Enfin, le pH est calculé en utilisant...

$$\text{pH} = -\log_{10}([\text{H}^+]) \quad (3.14)$$

Le modèle de simulation comprend les équations 3.11... 3.14. Les équations 3.12 et 3.13 sont des équations différentielles bilinéaires. Elles sont essentiellement linéaires pour de petites perturbations de F_B avec F_A constante, tandis que les équations 3.11 et 3.14 sont des équations statiques fortement non linéaires. Ainsi, le modèle peut être considéré comme ayant des caractéristiques dynamiques linéaires et des caractéristiques statiques fortement non linéaires. Le modèle néglige certains effets, dont la plupart sont généralement mineurs, mais dont certains peuvent être importants pour certains processus. Par exemple, le modèle suppose :

1. Température constante de 25° C. Les constantes de dissociation et les électrodes de pH dépendent de la température et ceci devrait être pris en compte pour les applications à haute température
2. Le mélange parfait. De nombreux CSTR industriels sont mal mélangés et il existe toute une gamme de modèles de réacteurs qui présentent un mélange non parfait.

Le modèle de base a été jugé satisfaisant pour cette étude de cas, car il présente des caractéristiques non linéaires dominantes pour ce processus acide / base.

3.3 APPRENTISSAGE ET VALIDATION DES MODELES RNA

Une simulation du processus a été réalisée à l'aide des valeurs des paramètres données dans le tableau 3.1. F_A est maintenu constant et un signal aléatoire d'amplitude de 10% a été superposé à un état statique, F_B , pour fournir des données E/ S dynamiques (F_B et pH) pour un apprentissage du réseau de neurones et la validation du modèle (DOHERTY, 1999).

Paramètre	Symbole	Valeur
Volume liquide CSTR	V	3000 l
Débit acide	F_A	4.05 l/s
Débit de la base	F_B	25.75 l/s
Concentration de l'acide	C_A	0.3178 M
Concentration de la base	C_B	0.05 M
Constante dissociation acide	K_A	1.8×10^{-5}
Constante dissociation base	K_B	1.0×10^{-14}
Effluent pH	pH	9.18

Tableau 3.1 Paramètres du Processus

3.3.1 Procédure expérimentale

L'objectif de cette étude est d'établir quels sont les effets de chacune des étapes dans la conception d'un modèle RNA quant à la validité du modèle.

En évaluant l'effet de structure du modèle ANN NARX sur la validité du modèle, il n'y a pas de tentative pour optimiser la taille de la couche cachée. Lors d'expériences préliminaires, il a été noté que, sous réserve qu'un réseau MLP inclue suffisamment de neurones dans la couche cachée, la précision du RNA était relativement insensible au nombre de neurones des couches cachées (POLLARD, 1992).

Par conséquent, la taille de la couche cachée a été fixée expérimentalement à 15, ce qui est suffisant pour permettre la convergence de tous les réseaux testés. La fixation de la taille de la couche cachée permet d'observer clairement les effets dus au changement de l'ordre du modèle NARX. Comme tout changement de précision de la prédiction du réseau peut être plus directement attribuable aux changements de la seule couche d'entrée, plutôt que des changements à la fois dans les couches d'entrées et les couches cachées.

Une approche uniforme a toujours été prise pour l'apprentissage et la validation des modèles RNA. Les réseaux MLP ont été formés pour utiliser la version continue de l'algorithme de la rétro-propagation.

Afin de rendre les conditions d'apprentissage aussi équitables que possible, la longueur des données a été fixée à 1000 E / S vecteurs d'apprentissage et l'apprentissage de réseaux de neurones a été terminé après 1000 époques, puisque après 1000 époques le taux d'apprentissage et le momentum tendaient vers zéro. Dans tous les cas, on a observé que ce n'était pas suffisant pour obtenir la convergence des poids du réseau vers un minimum de la fonction du coût d'apprentissage. De plus, un apprentissage ultérieur n'a pas réussi à diminuer la fonction du coût.

Un jeu de cinq ensembles de données a servi à tester la capacité de tous les modèles neuronaux. Ces signaux de validation croisée sont générés par trois différents signaux d'amplitude aléatoire avec une horloge de périodes de 72, 120 et 240 secondes, un signal d'amplitude aléatoire dont l'amplitude a été fortement réduite. Par conséquent, les seules prédictions du modèle sont testées dans la région du gain maximum du processus (point 4 de la figure 3.2), avec des données statiques. Ces différents signaux d'essai ont été sélectionnés pour tester les capacités de la généralisation des réseaux de neurones dans des conditions différentes, et leur précision de généralisation a été évaluée à l'aide de l'indice de performance (IP) et l'indice de performance normalisé (NPI).

La validité des modèles RNA a aussi été évaluée en utilisant les tests de corrélation. Pour appliquer ces tests à un NARX structuré modèle de bruit, ce qui représente une erreur de mesure, doit être ajoutée à la sortie du processus. Une séquence aléatoire d'un bruit gaussien distribué de moyenne zéro a été ajoutée à la sortie du simulateur. La variance du bruit a été fixé pour donner un rapport signal / bruit de 20dB. Les tests de corrélation ont été utilisés à titre d'indication supplémentaire.

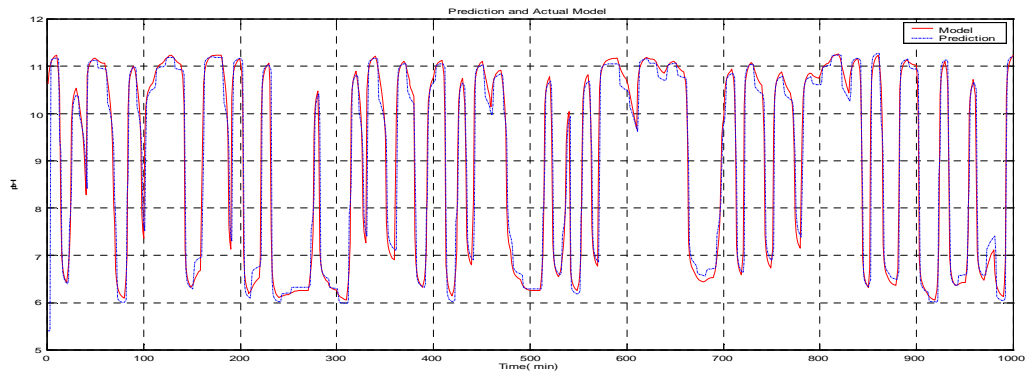
3.3.2 Excitation du processus à l'aide d'un signal à amplitude aléatoire

La première étape de la procédure de conception du modèle RNA est la génération d'un ensemble de données d'apprentissage, et ceci exige la conception d'un signal d'entrée du processus. La conception du signal d'entrée pour l'identification des systèmes non linéaires est plus compliquée qu'elle ne l'est pour l'identification des systèmes linéaires. Le signal d'entrée devrait exciter toutes les fréquences d'intérêt et devrait également exciter le processus sur l'ensemble de la région d'exploitation requise (LEONTARITIS et BILLINGS, 1987). Un signal à amplitude aléatoire est couramment utilisé comme signal d'excitation du processus pour générer les signaux de données en boucle ouverte pour l'apprentissage des réseaux de neurones. Ce signal se compose d'une variable aléatoire uniformément répartie appliquée à l'entrée du processus à chaque période d'horloge (LEONTARITIS et BILLINGS, 1987; POTTMANN et SEBORG, 1992a). Par exemple, une séquence pseudo aléatoire binaire (PRBS), est largement utilisée pour l'identification d'un système linéaire. Un signal à amplitude aléatoire est spécifié par sa période d'horloge, ce qui devrait être un multiple de l'échantillon de temps afin que l'entrée du processus soit constante entre deux échantillons consécutifs, et sa gamme d'amplitude, qui peut être exprimée en pourcentage de la déviation maximale de valeur de l'état statique.

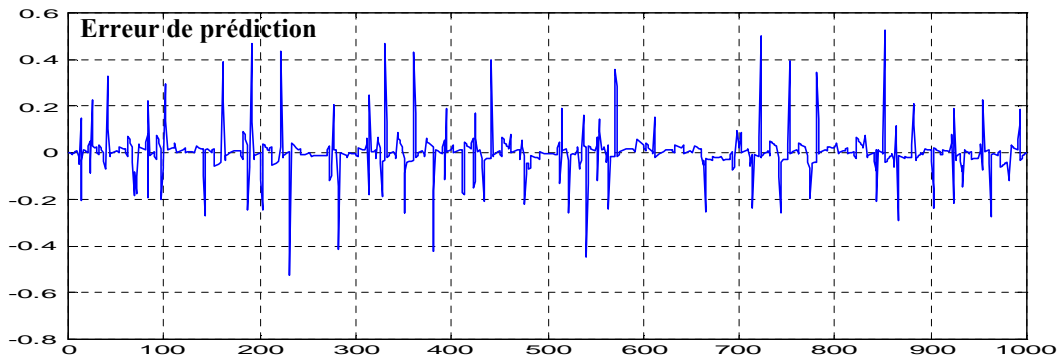
3.4 UN MODELE INITIAL RNA

Les valeurs initiales des paramètres du signal d'excitation et de l'échantillon temps sont pris à partir des travaux antérieurs d'identification par réseaux de neurones pour le processus de pH CSTR (et BHAT McAVOY, 1990). On prend 10% du signal RAS avec une période d'horloge de 240 secondes et un échantillon de temps de 24 secondes. L'apprentissage est fait avec des données provenant de la simulation CSTR à l'aide de ces paramètres, sauf mention contraire. Le modèle initial ANN, modèle 0, avait une structure NARX d'ordre 5 et il était utilisé par BHAT et McAVOY (1990), et le modèle de prédiction d'un pas en avant sur un ensemble de données d'essai est illustré à la figure 3.3. Bien que le modèle semble suivre le processus relativement bien, il y a quelques erreurs de prédiction de l'ordre de 0.6 unités pH,

qui est d'environ la moitié de la plage de fonctionnement. Lors d'un essai avec les cinq signaux d'essai, le modèle 0 avait un IP de 1.52 et un NPI de 4.7.



line discrète : Processus - line continue :RNA



Nombre d'échantillons

Figure 3.3 : La prédiction un pas en avant du modèle 0 par un RNA sur des données de test

On pensait que la validation croisée des résultats suggère que le modèle 0 n'était pas une exacte représentation du processus de pH, et cette position a été renforcée par les résultats du test de corrélation (Figure 3.4), où deux des cinq tests de corrélation croisée dépassent la confiance limite.

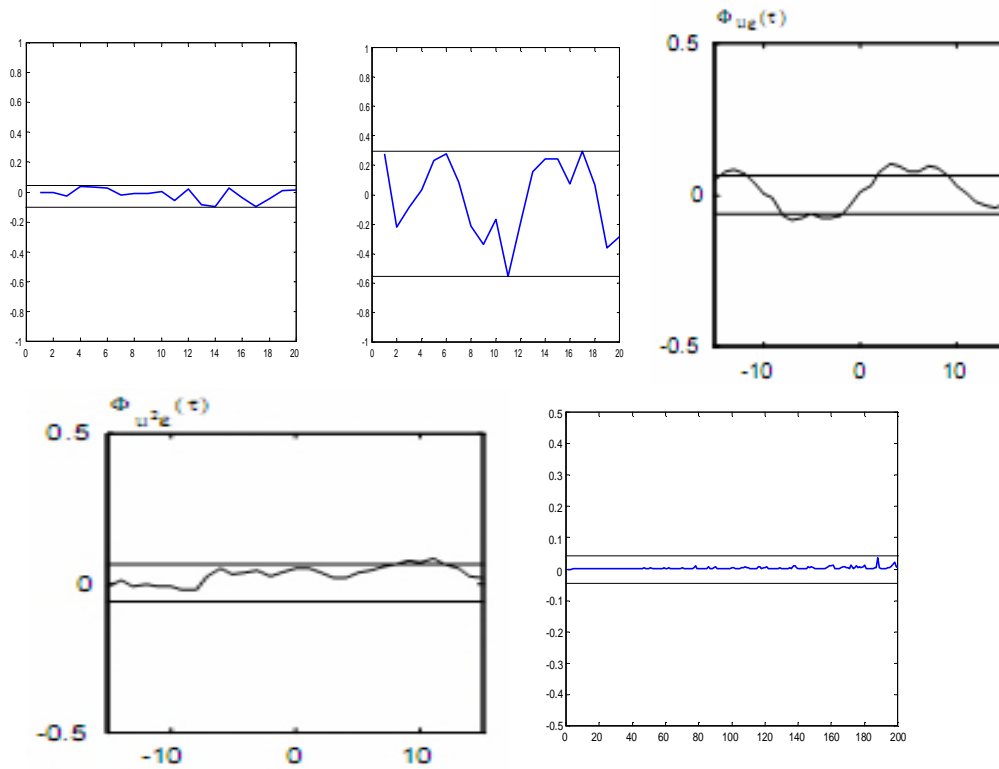


Figure 3.4 : Résultats du test de corrélation pour le modèle 0

La validation croisée et les résultats des tests de corrélation pour le modèle 0 fournissent un benchmark RNA auquel d'autres modèles pourraient être comparés. Les techniques de conception RNA sont désormais appliquées pour tenter d'élaborer une représentation plus précise du processus de simulation.

3.5 ETUDE DU MODELE LINEAIRE LOCAL

Le but ici était d'établir une structure de modèle convenable et des données pour l'échantillon temps pour des modèles linéaires du processus. Les modèles ARX ont été identifiés sur les sept points de fonctionnement indiqué dans la Figure 3.2 à l'aide d'un signal d'excitation RAS. L'amplitude du RAS a été choisie pour donner une variation maximum de sortie de +0,1 unité de pH sur le point de fonctionnement. Le processus peut être représenté par un modèle linéaire sur une petite plage de fonctionnement et d'un jeu typique de données d'E / S comme illustré à la figure 3.5.

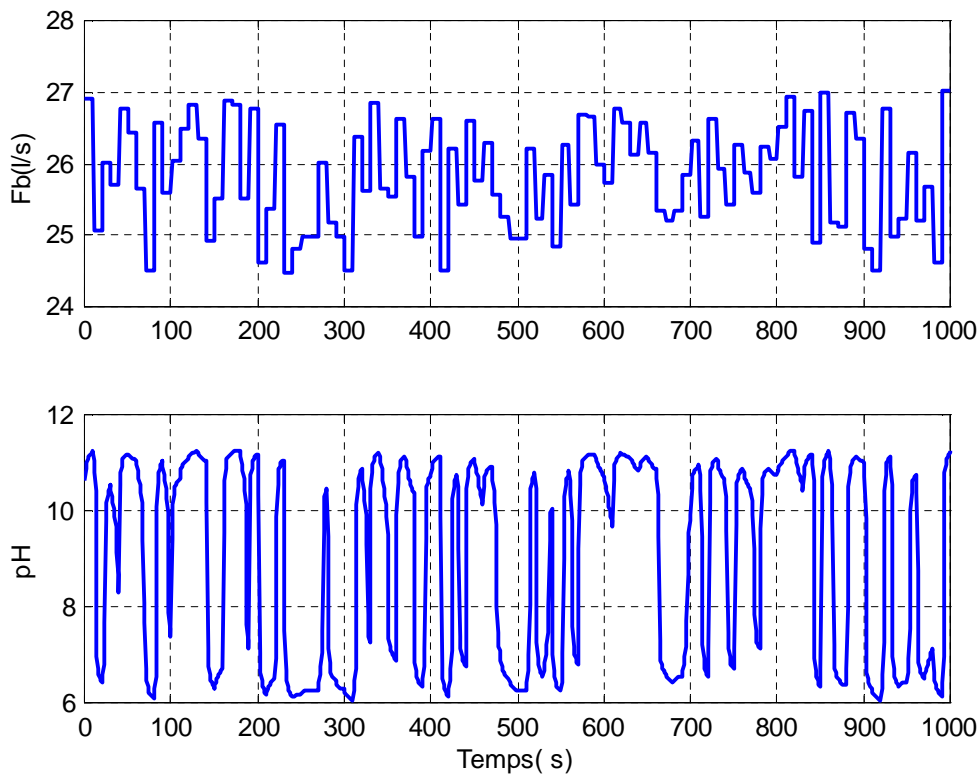


Figure 3.5 : Modèle typique ARX des données d'identification. Point 4 de la courbe de titration ($F_B = 25,75$ l / min. $\text{pH} = 9,18$)

3.5.1 Structure du modèle ARX

Initialement, les modèles ARX locaux ont été identifiés à l'aide d'une période d'horloge de 240 secondes et un temps d'échantillonnage de 24 secondes, puis une structure convenable du modèle ARX a été déterminée. Par la suite, cet échantillon du temps et la période d'horloge ont été étudiés. Pour choisir un modèle temps mort convenable, les modèles ARX de différents temps morts ont été identifiés pour les différents points de fonctionnement. Pour chaque temps mort les modèles de différents ordres ont été identifiés, et la Figure 3.6 montre l'erreur finale de prédiction (AFPE) pour les modèles identifiés au point de fonctionnement 4. Ces modèles représentent les résultats obtenus aux autres points de fonctionnement. Dans tous les cas, un modèle temps mort de $k = 1$ donne la valeur minimale de l'AFPE et cela a également été le cas pour l'autre modèle à fonctions de parcimonie, y compris la fonction de perte. Ce résultat était attendu puisque le simulateur suppose un parfait mélange dans le CSTR et, par conséquent, il n'y a pas de temporisation.

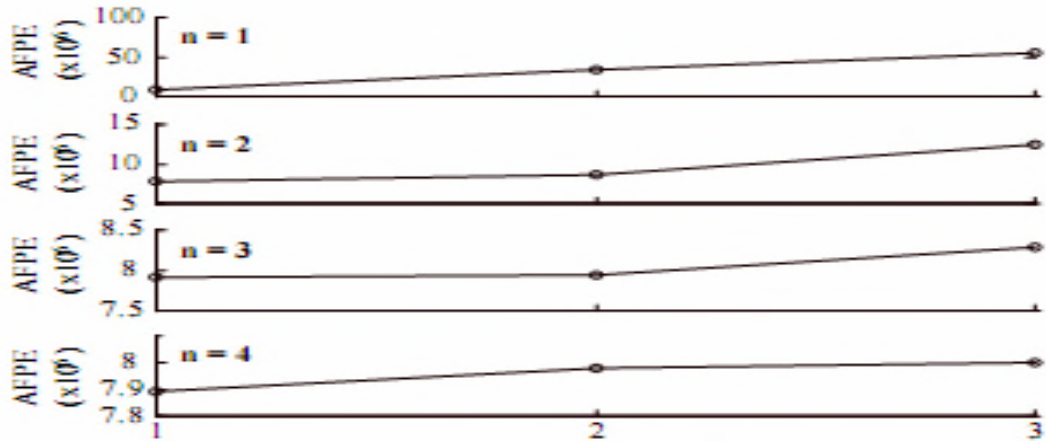


Figure 3.6 : AFPE pour les modèles ARX avec deadtime, k , pour différents ordres du modèle. Les modèles ARX ont été identifiés au point exploitation 4 de la courbe de titration.

Le modèle deadtime était désormais fixé à $k = 1$ et le modèle local de procédure de recherche est répété pour établir l'ordre optimum du modèle ARX. La Figure 3.7 montre l'AFPE pour les modèles ARX d'ordres 1 à 6 identifiés à différents points de fonctionnement. La tendance générale de réduction négligeable pour les modèles AFPE d'ordre supérieur à 2 a été observée pour tous les points d'exploitation. Des résultats analogues ont été obtenus pour l'autre modèle parcimonieux et pour la fonction de perte. Par conséquent, la structure optimale du modèle ARX pour un modèle local de ce processus avait un modèle deadtime de 1 et un modèle d'ordre 2, ce qui a été le cas pour chacune des sept régions distinctes de fonctionnement.)

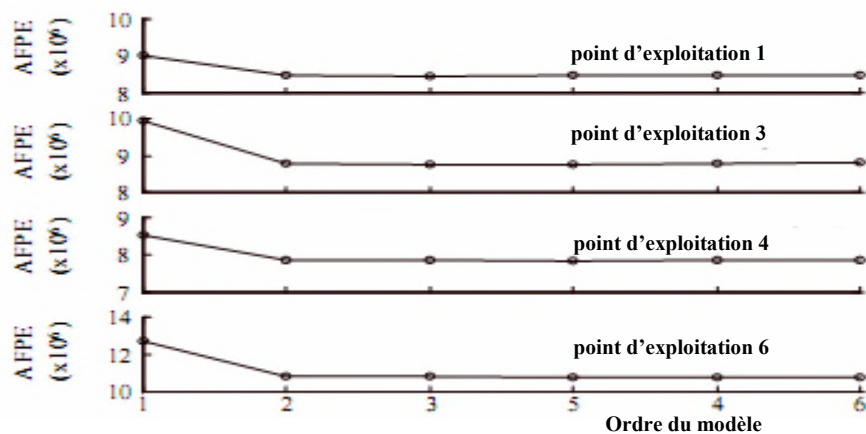


Figure 3.7 : AFPE pour modèles ARX d'ordre n identifiés aux différents points d'exploitations

3.5.2 Echantillons du temps

Il est important d'utiliser une donnée appropriée pour l'échantillon de temps pour l'identification d'un modèle linéaire. Si l'intervalle de l'échantillon est trop important alors les données d'identification ne décrivent pas correctement le comportement dynamique des systèmes. D'autre part, un intervalle d'échantillon trop faible peut entraîner des problèmes dans l'algorithme d'identification et peut augmenter l'effet du bruit du processus. Toutefois, pour la plupart des systèmes linéaires, la plage possible entre les intervalles d'échantillon trop larges et trop petits est relativement importante (ISERMANN, 1980) et certaines directives pour sélectionner un intervalle d'échantillon approprié sont de - 15 à 115 fois les 95% du temps de rétablissement (ISERMANN, 1980), inférieure à 0,38 fois la plus petite constante de temps d'intérêt (ASTROM et WITTENMARK, 1989), 1/5 à 1/10 fois la constante de temps dominante (SEBORG., 1986) et une fréquence d'échantillonnage de 10 fois la largeur de la bande passante du système(LJUNG, 1987). L'application de ces recommandations à la constante de temps (τ) de la fonction de transfert d'ordre 1 permettra un large éventail d'échantillons des intervalles de $0,1\tau$ à $0,6\tau$.

Les réponses à un échelon de petite amplitude ont montré que le processus avait une constante de temps d'environ 80 secondes et, par conséquent, les directives précédentes ont montré que l'intervalle échantillon de 24 secondes était trop grand. La figure 3.8 montre la fonction de perte pour les modèles identifiés ARX d'ordre 2 au point de fonctionnement 3 en utilisant différents échantillons, et la diminution relative de la fonction de perte pour de petits intervalles d'échantillon est représentative des résultats pour les modèles identifiés à d'autres points de fonctionnement.

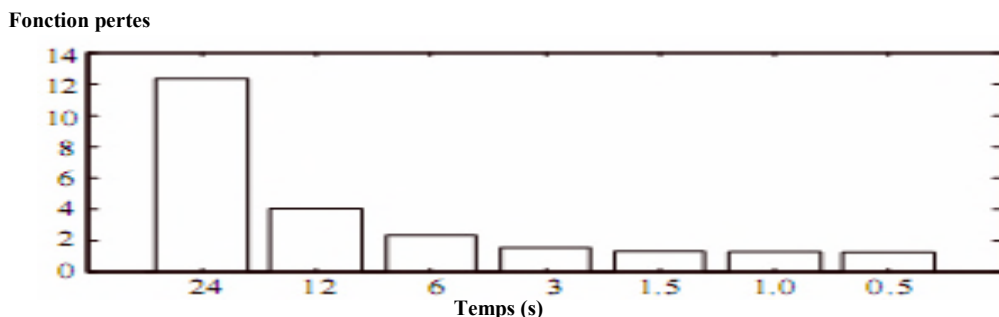


Figure 3.8 : La fonction de perte pour les modèles ARX identifiés avec des échantillons de temps différents

Si la fonction de perte a été le seul critère de sélection pour l'échantillon de temps, alors 6 ou 3 secondes sera probablement retenu comme un bon compromis. Toutefois, pour un

modèle linéaire robuste, la position du pôle dominant du modèle dans le plan z ne devrait pas être trop proche du cercle unité de la limite de stabilité ($Z = 1$) et c'est particulièrement le cas pour le modèle ARX (LJUNG, 1987). Le pôle dominant du modèle migre vers la limite de stabilité lorsque l'intervalle de l'échantillon est réduit, comme le montre le tableau 3.2. Généralement un modèle ARX avec un pôle de magnitude supérieure à 0,9 serait rejeté parce que le modèle laisse passer des bandes de fréquences très élevées, y compris le bruit du processus. Ainsi, à titre de compromis entre la fonction de perte et l'amplitude du pôle dominant, un échantillon de temps d'environ 12 secondes serait préférentiel pour un modèle ARX local de ce processus.

Echantillon de temps (s)	Amplitude du pôle dominant
24	0.782
12	0.888
6	0.944
3	0.971
1.5	0.986
1.0	0.990
0.5	0.995

Tableau 3.2 Amplitudes du pôle dominant des modèles ARX identifiés avec différents échantillons de temps. Point d'exploitation 3.

Les expériences sur l'ordre du modèle ARX ont été répétées en utilisant un intervalle d'échantillon de 12 secondes. Bien que l'amplitude des AFPEs était plus petite (comme indiqué par la diminution de la fonction de perte de la figure 3.8) que pour les précédentes expériences qui employaient un échantillon de 24 secondes, la tendance de l'AFPE montre encore que la structure de degré 2 est plus appropriée pour le modèle ARX de ce processus. Enfin, le choix initial de 240 secondes de la période d'horloge a été étudié. Pour l'identification des systèmes linéaires, la période d'horloge d'un signal PRBS est souvent choisie égale au temps de rétablissement du système (GODFREY, 1980). Les 95% du temps de rétablissement pour les faibles variations à l'entrée de ce processus est d'environ 240 secondes, et, par conséquent, 240 secondes semblent être un choix raisonnable de la période d'horloge RAS. Cela a été confirmé par l'identification de modèles ARX de second ordre employant un échantillon de temps de 12 secondes avec différentes périodes d'horloge. Il a été

constaté que des périodes d'horloge allant de 4 à 8 minutes donnent des modèles ayant des fonctions de perte de même amplitude. Par contre, en réduisant la période d'horloge à 2 minutes la fonction de perte est multipliée par 2.

3.6 ETUDE DU MODELE RNA NON LINEAIRE

Pour établir un ordre approprié au modèle NARX pour le processus CSTR, les réseaux de neurones d'un modèle d'ordre croissant ont été formés et les performances des données d'apprentissage ont été comparées en utilisant la fonction de perte. La figure 3.9a. montre la fonction de perte pour les modèles de réseaux de neurones avec des modèles de différents ordres $N = 1, \dots, 10$ et un retard de $k = 1$. Le retard d'un échantillon entre l'entrée et la sortie du processus du CSTR a été choisi en fonction du fait que le processus était parfaitement mélangé, et que l'étude du modèle linéaire a clairement indiqué que le retard d'un échantillon était approprié pour une représentation ARX locale du processus. La figure 3.9a montre que le modèle d'ordre 10 présente la plus faible fonction de perte. Toutefois, la fonction de pertes, par elle-même, ne constitue pas une bonne base pour le choix de l'ordre du modèle. L'utilité de la fonction de pertes (Figure 3.9b) est qu'elle montre la variation relative de la fonction de pertes pour chaque augmentation de l'ordre du modèle. Cela met en évidence un modèle de second ordre qui provoque une forte réduction de la fonction de pertes par rapport à un modèle de premier ordre, mais les diminutions de la fonction de pertes dues aux ordres croissants du modèle ne sont pas comparables.

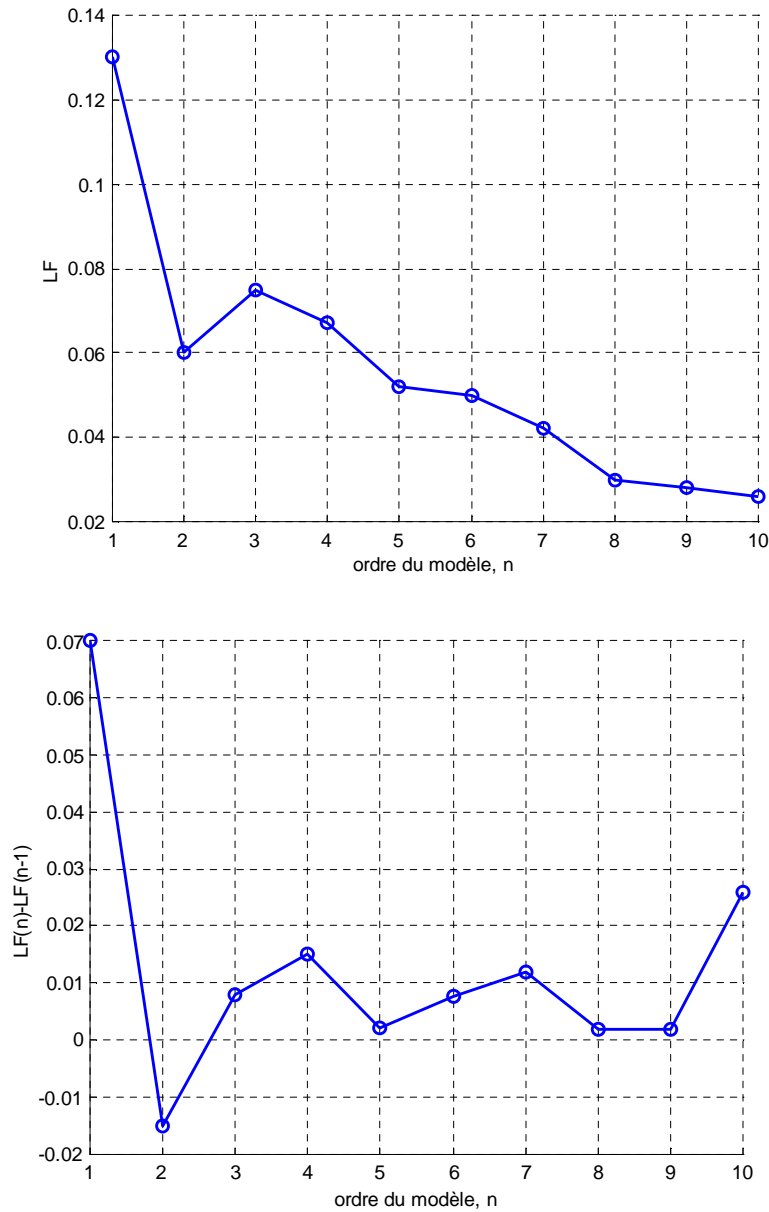


Figure 3.9 : Comparaison des RNA avec différents ordres du modèle NARX. (a) fonction de pertes, (b) tendance de la fonction de pertes

Le difficile équilibre entre la précision et la complexité du modèle peut être résolu par l'application des indices de parcimonie du modèle, comme indiqué sur la figure 3.10. Ces indices de parcimonie sont des fonctions pondérées de la fonction de pertes qui pénalisent les réductions au détriment d'une augmentation de la taille du modèle. Les fonctions sont différentes dans la façon de pénaliser l'augmentation du nombre de paramètres du modèle et, par conséquent, ils ne donnent pas des informations identiques. Toutefois, ils fournissent une information commune, puisque pour l'ensemble des quatre indices la valeur du 2e ordre montre une nette amélioration par rapport à la valeur de l'ordre 1, et sont minimum dans

l'intervalle du 1er au 5e. L'interprétation rigide du critère donne le 10e, le 2e et le 8e ordre pour le AFPE, AIC, BIC et KLILC respectivement, parce qu'ils donnent des valeurs minimales. Toutefois, les faibles diminutions enregistrées dans les AFPE et AIC pour les structures de modèles d'ordre supérieur ne sont pas convaincantes pour la structure d'ordre 10. De la même façon la valeur de la KLILC d'ordre 8 n'est que marginalement inférieure à celle d'ordre 2. Par conséquent, le jugement nous pousse à choisir l'ordre 2, ce qui a été indiqué par la fonction de pertes.

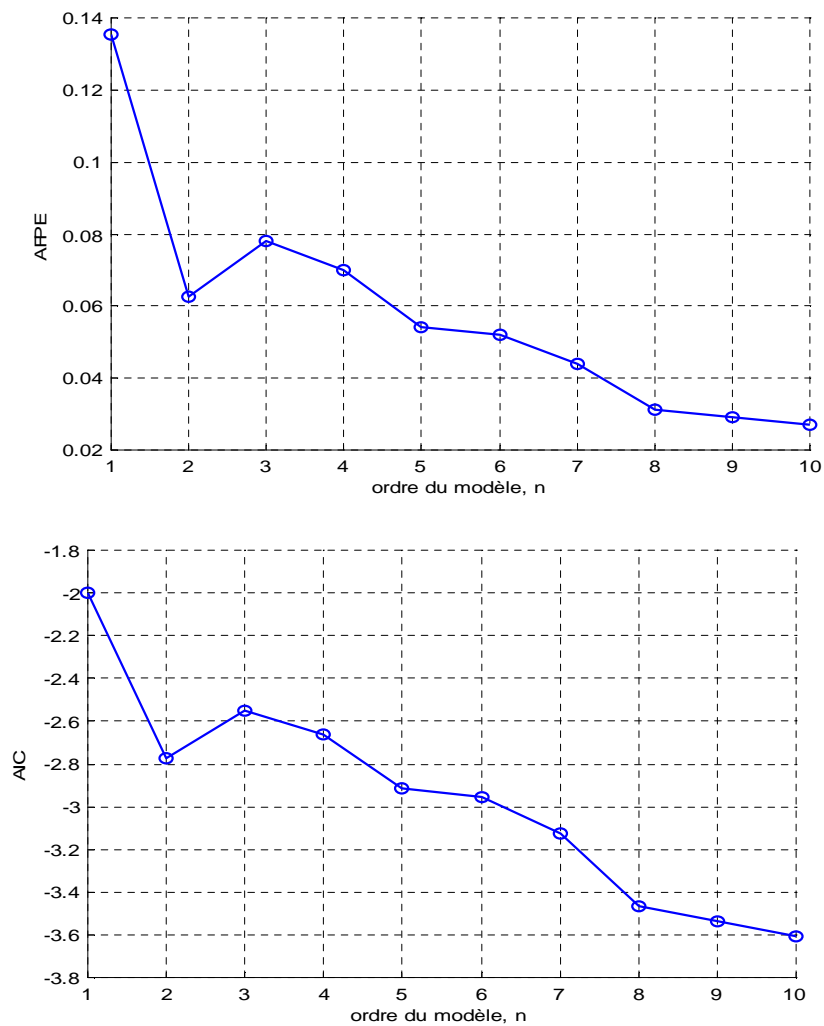


Figure 3.10 : Comparaison des RNA avec différents ordres du modèle NARX.

(a) AFPE, (b) AIC.

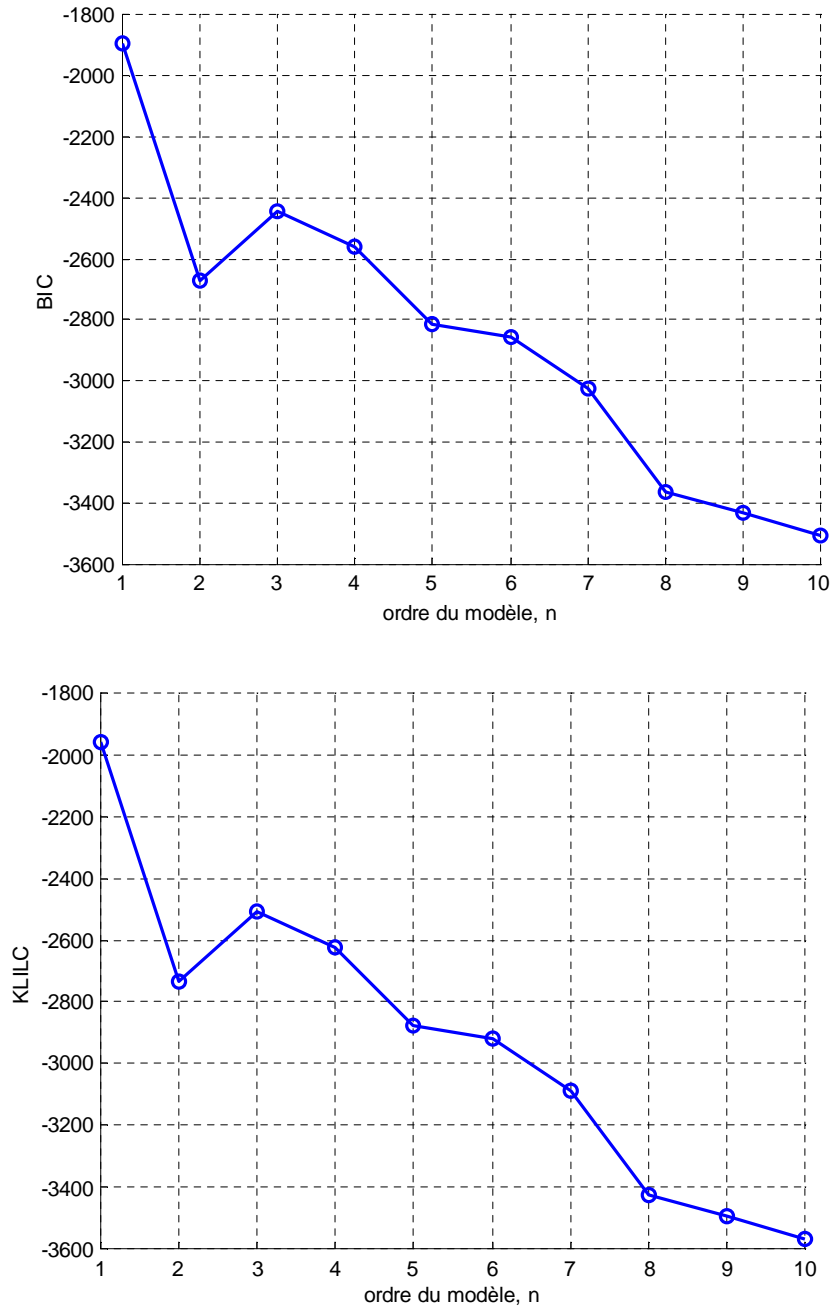


Figure 3.11 : Comparaison des RNA avec différents ordres du modèle NARX.

(a) BIC, (b) KLILC

Tandis qu'une interprétation stricte des indices de parcimonie du modèle précédent ne donne pas une structure optimale évidente du modèle, leur interprétation donne une structure NARX de degré 2 comme un bon compromis. Bien entendu, ces indices de parcimonie sont évalués sur l'apprentissage des données, car elles sont fondées sur l'hypothèse que la fonction de pertes (des données d'apprentissage) diminue avec l'augmentation de l'ordre du modèle, et par conséquent, ils ne sont pas un indicateur de précision. Pour évaluer la précision, les

modèles de réseaux de neurones sont testés à l'aide de 5 des signaux de test et les indices de performance et de la performance normalisée sont évalués (Figure 3.11). L'indice de performance indique clairement une structure de modèle de second ordre comme présentant les meilleures performances. Les résultats sont démontrés aussi bien par les signaux d'essai utilisés que par l'indice de performance normalisé.

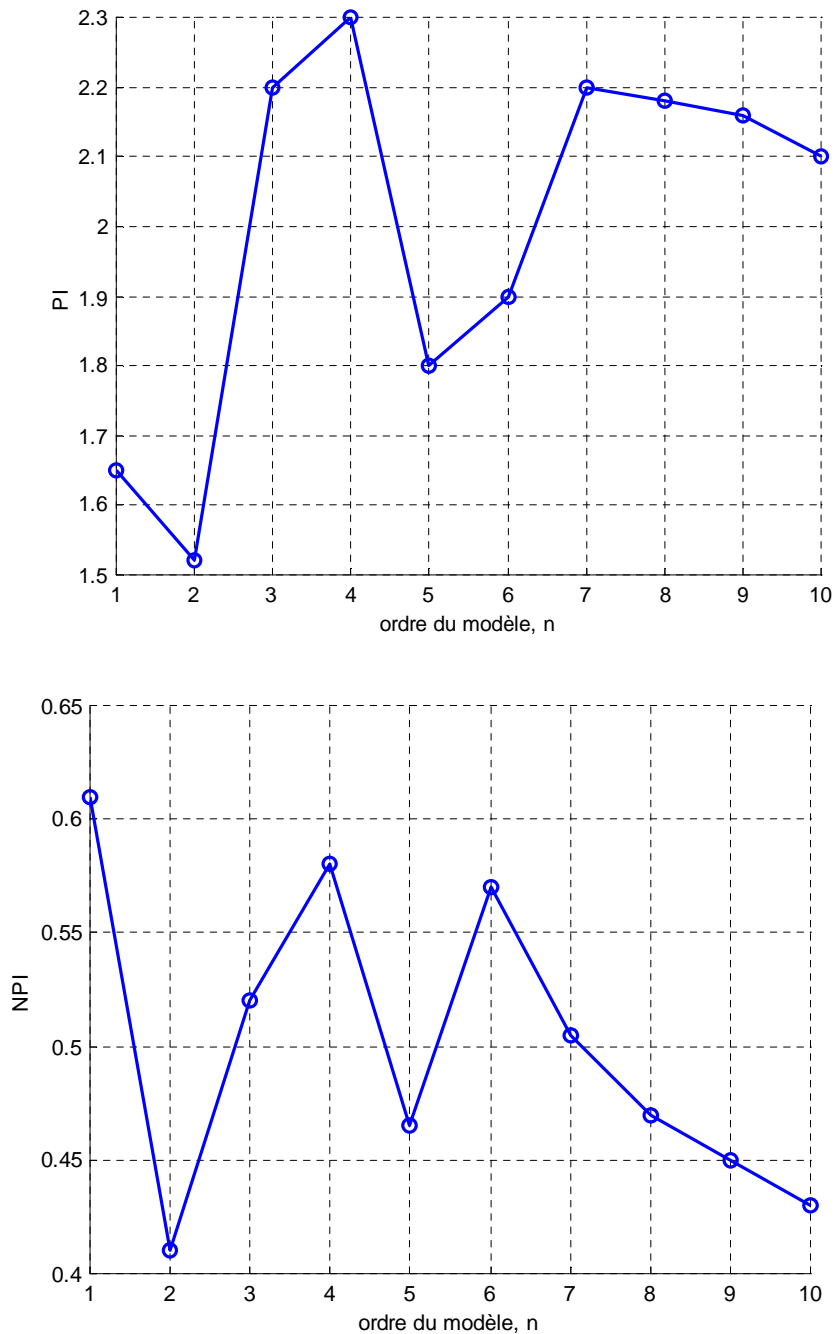


Figure 3.12 : Comparaison des RNA avec différents ordres du modèle NARX.

(a) Indice de Performance (PI), (b) Indice de Performance Normalisé NPI.

3.6.1 Sélection de l'échantillon du temps

Le choix de l'échantillon des données du temps d'après le théorème de l'échantillonnage, correspondent à une fréquence d'échantillonnage d'au moins le double de celui de la plus haute fréquence du processus à identifier. Autre que cette exigence fondamentale de l'échantillon, on sait peu de choses à l'heure actuelle comment l'échantillon du temps agit sur les performances d'un modèle de réseaux de neurones. Cette section analyse les effets de l'échantillon des données temps sur les performances du modèle pour le processus CSTR du pH. Plusieurs RNA identiques ont été formés en utilisant les mêmes données d'apprentissage échantillonnées à des intervalles allant de 24 à 0,5 secondes. Les réseaux sont de 2ème ordre, et on comptait 15 neurones cachés. L'évolution de la fonction du coût du réseau d'apprentissage (Figure 3.13) montre que la diminution de l'intervalle d'échantillon fait que la fonction du coût du réseau d'apprentissage converge plus rapidement et à une moindre valeur, et dans ce cas, le réseau apprend les données d'apprentissage avec plus de précision avec un échantillon plus rapide. Toutefois, du fait que les réseaux ont été formés en utilisant le même nombre de vecteurs d'apprentissage, la longueur en temps réel des données d'apprentissage par époque est directement proportionnelle à l'intervalle de l'échantillon utilisé. Par conséquent, la réduction de l'intervalle échantillon donne moins d'impulsions d'horloge RAS par époque et réduit aussi les changements à la sortie du processus entre les échantillons adjacents. Par conséquent, cela permet au RNA de s'adapter aux données d'apprentissage avec plus de précision.

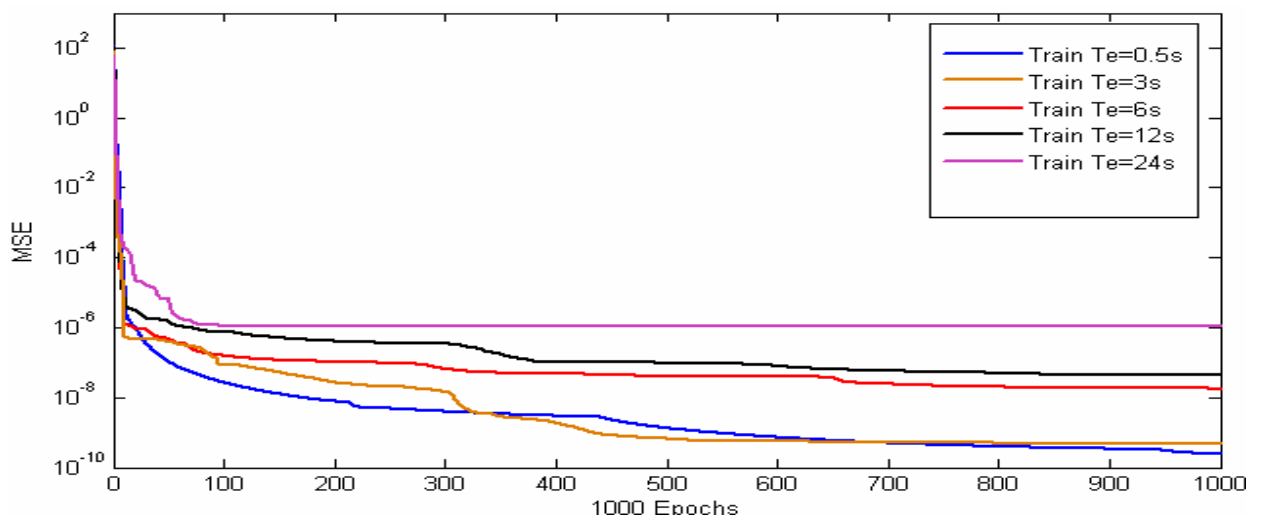


Figure 3.13 : Apprentissage des fonctions d'erreurs pour les RNA d'ordre 2 avec différents intervalles de temps.

La figure 3.14 montre l'indice de performance et l'indice de performance normalisé pour ces modèles de réseaux de neurones lors d'essais avec la validation croisée des signaux. Pour le calcul des indices tous les modèles ont été testés sur la même fenêtre de temps réel et, par conséquent, le nombre de vecteurs de test pour chacun des signaux d'essai est judicieusement augmenté pour une réduction de l'intervalle échantillon.

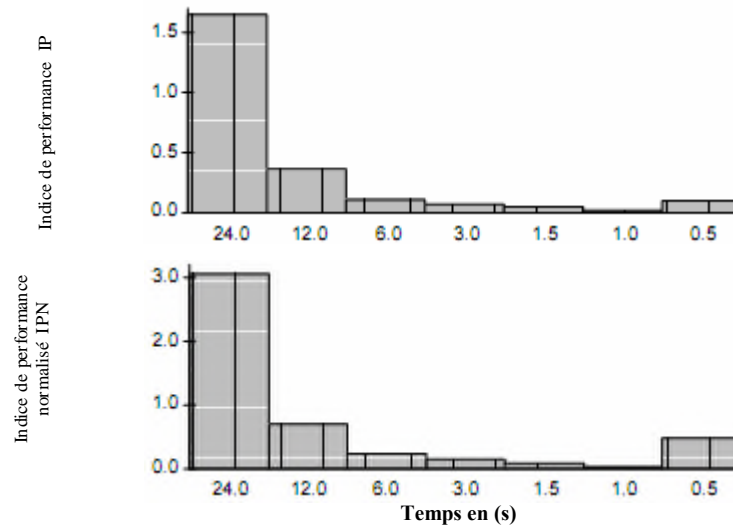


Figure 3.14 : Modèles ANN 2ème ordre employant différents échantillons du temps.

Contrairement à la fonction du coût du réseau d'apprentissage pour un échantillon plus rapide, illustrée à la figure 3.8, l'indice de performance et sa forme normalisée (Figure 3.14) montrent qu'il y a une limite inférieure de la sélection de l'échantillon. Un échantillon de temps de 1 seconde donne les valeurs les plus faibles des indices de performance. La figure 3.14 montre aussi qu'une performance satisfaisante du réseau peut être obtenue avec un ensemble d'échantillons, dans ce cas, entre 0,5 et 12 secondes.

Alors qu'il n'y a pas de restrictions sur le choix de l'échantillon dans l'exercice de simulation, la présence de bruit du processus peut imposer des limites sur le choix de l'échantillon temps pour la modélisation d'un processus réel. Réduire les données de l'échantillon augmente les effets du bruit du processus aux hautes fréquences, ce qui peut se traduire par un mauvais modèle. Les contraintes de calcul sont un autre facteur d'application spécifique qui peut limiter le choix de l'échantillon temps dans la pratique. Car, si le modèle identifié doit être exécuté temps réel, les calculs doivent être effectués en moins de temps que l'intervalle échantillon. Souvent, il est nécessaire que les calculs soient achevés en une petite fraction de l'intervalle échantillon, comme dans un modèle de contrôle basé sur un

algorithme. En Comparant les Figure 3.12 et Figure 3.14, on voit que pour ce processus, la relative amélioration de la précision de prédiction obtenue à partir d'un bon choix de l'échantillon est considérablement plus importante que celle résultant d'un bon choix de l'ordre du modèle. Toutefois, en faisant cette comparaison, il faut garder à l'esprit que ces indices de performance et performance normalisée sont calculés à partir d'un pas en avant de prédiction et, dans ce cas, l'horizon d'un modèle de prédiction est égal à son intervalle échantillon. Lorsque l'horizon de prédiction est réduit, il n'est pas surprenant de parvenir à une prédiction plus précise, bien que cela n'est pas toujours le cas comme le prouve l'augmentation de l'indice de performance et performance normalisée pour les modèles utilisant 1,0 et 0,5 secondes pour les intervalles d'échantillon dans la figure 3.14.

L'échantillon de temps sélectionné pour la représentation d'un ARX local de ce processus, n'était pas directement applicable au modèle non linéaire. Toutefois, l'étude du modèle linéaire a indiqué qu'un échantillon de 24 secondes est lent.

3.6.2 Signaux d'excitation

L'objectif d'excitation du processus est de générer un processus de données E / S qui contient suffisamment d'information pour un réseau de neurones pour identifier les processus dynamiques non linéaires sur toute la plage de fonctionnement, et un signal RAS est communément utilisé pour atteindre cet objectif. Toutefois, il n'y a aucune raison pour que ce signal devrait atteindre une excitation adéquate pour les dynamiques non linéaires de tous les processus. Cela est particulièrement vrai pour ce processus CSTR de pH où la forte non linéarité est caractérisée par l'état statique de la courbe de titrage (Figure 3.2), où le gain subit un changement d'un facteur 150 entre les points de fonctionnement 7 et 4. L'un des obstacles à la modélisation d'un tel processus est d'obtenir des données de sortie dans la région où le gain du réseau d'apprentissage est élevé. Lorsque le pH est excité par un signal RAS standard, peu de données de sortie sont générées dans le voisinage du point de fonctionnement 4, où le gain du processus est maximum. La distribution de données de la figure 3.14 illustre le manque de sortie de données entre pH 7 et 10, et un modèle de réseau neuronal qui est formé à ce type de données peut avoir de grandes erreurs de prédiction dans cette région.

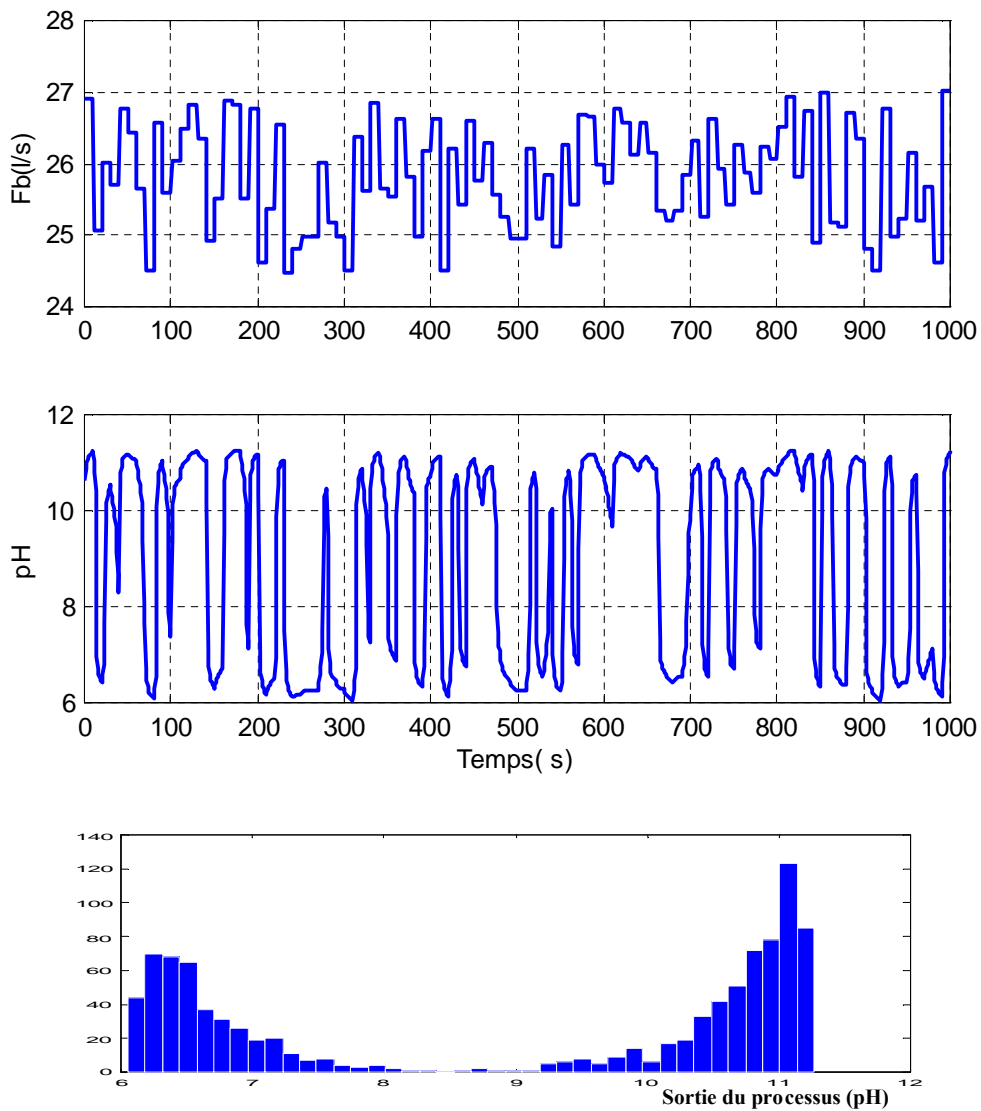


Figure 3.15 : La distribution de la sortie du processus pour un signal d'excitation conventionnel

Les changements des entrées consécutives générées par le RAS sont souvent de faibles valeurs et les sorties du processus restent dans la région faible du gain pour plusieurs impulsions consécutives d'horloge RAS. Ceci est un inconvénient car cela mène à une répartition inégale des données de sortie. Un moyen pratique d'améliorer l'uniformité de la distribution des données d'apprentissage est de forcer le signal dans la région du gain maximal du processus sur chaque impulsion d'horloge. Le signal forcé, représenté dans la figure 3.15, a une entrée uniformément répartie dans les deux intervalles au-dessus et au-dessous d'un seuil qui est choisi lorsque le processus est au maximum de l'état statique. Malgré que la distribution apparaisse encore comme très inégale, il y a une augmentation de

la densité de données entre pH 7 et 10 par rapport à la densité de données de sortie générées par un étalon RAS (Figure 3.14).

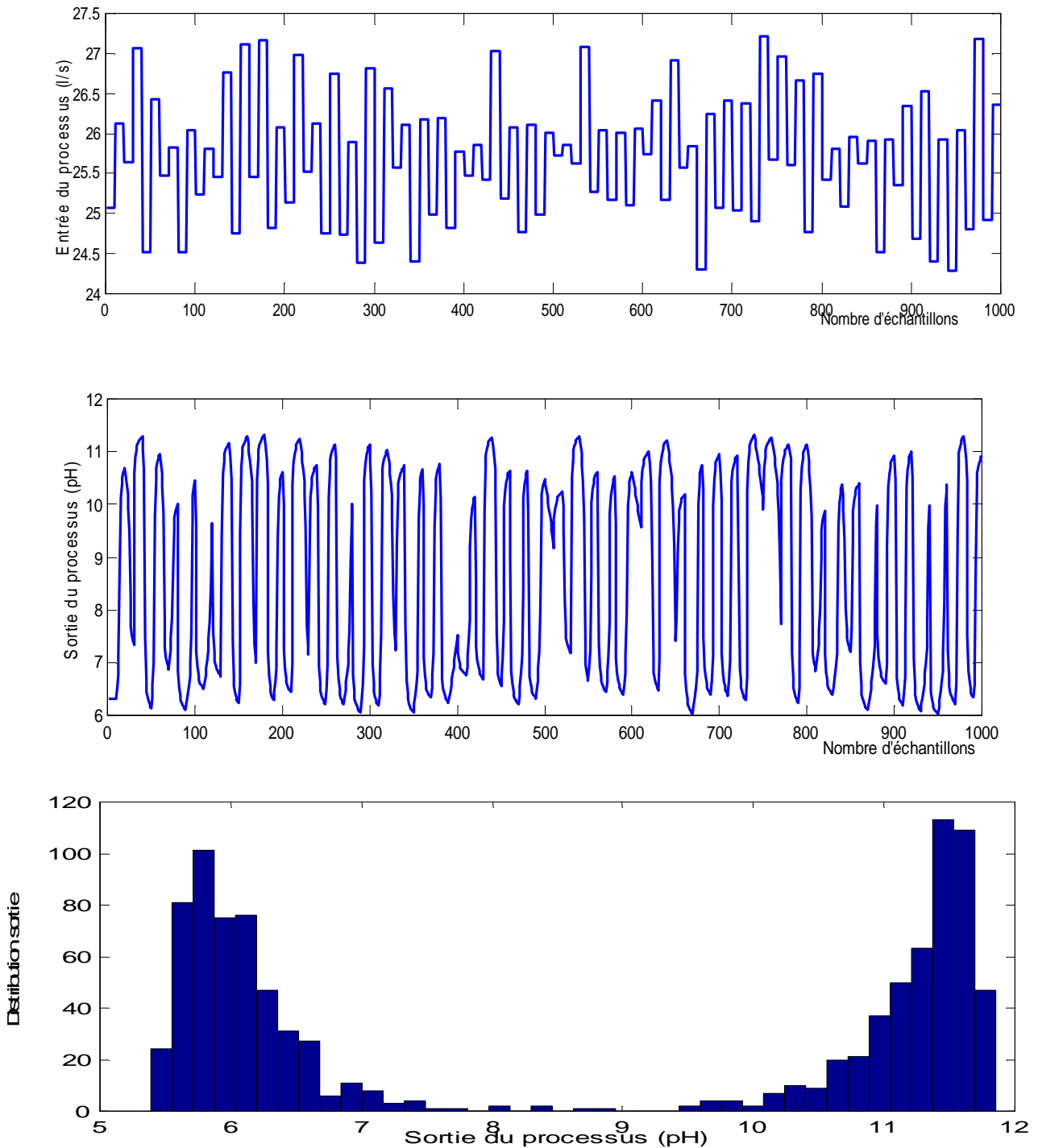


Figure 3.16 : La distribution de la sortie du processus pour un signal forcé d'excitation

Une nouvelle amélioration de la distribution des données de sortie (Figure 3.16) peut être réalisée par la réduction de l'amplitude de RAS sur une partie de l'identification expérimentale. Au milieu de l'identification expérimentale l'amplitude du signal RAS est

ramenée de 10% à 0,5%. La valeur centrale du signal RAS modifié a été choisie comme valeur d'entrée qui correspond au point de gain maximum du processus et ainsi, la réduction de l'amplitude du signal d'entrée fournit des données de sortie supplémentaires.

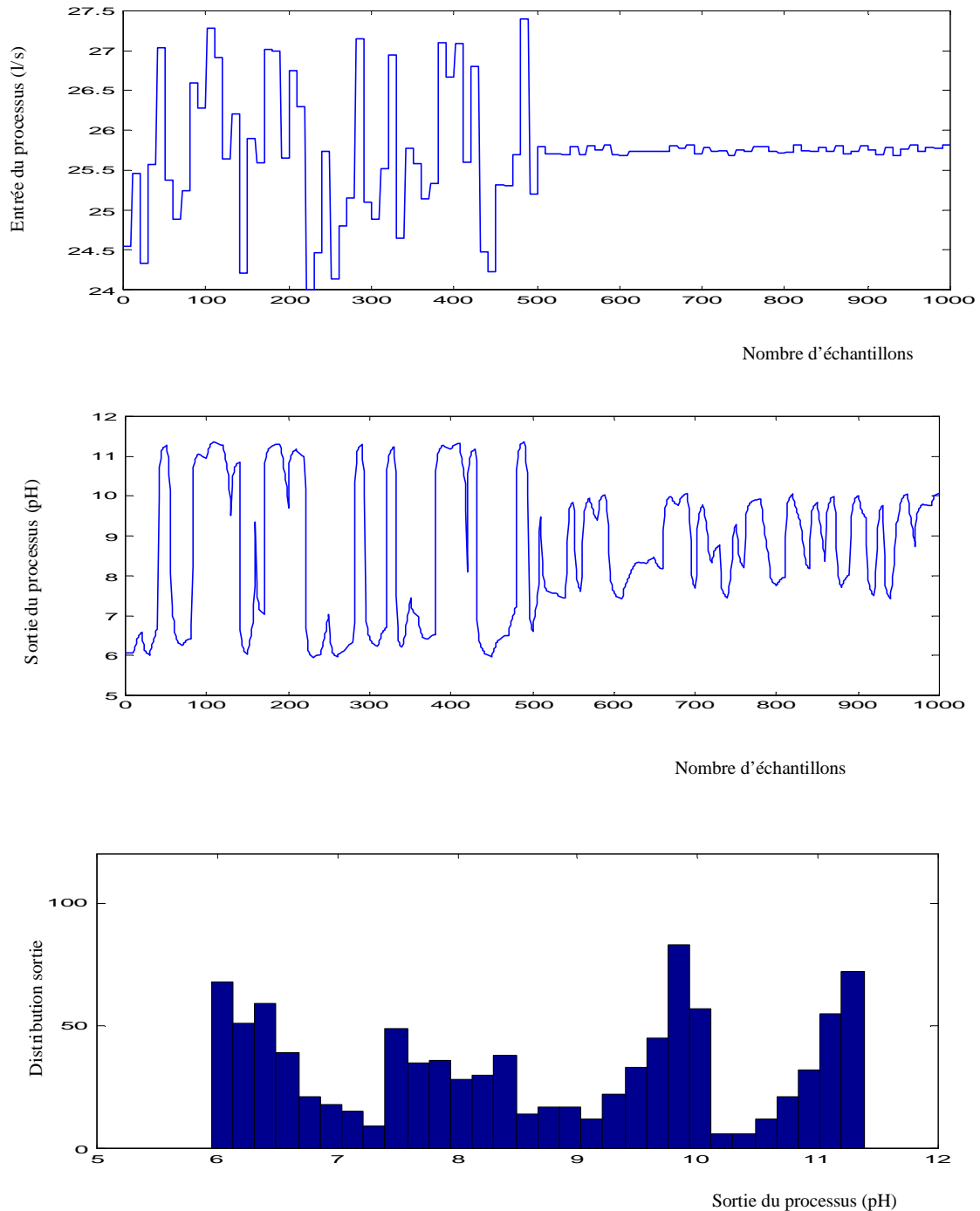


Figure 3.17 : La distribution de la sortie du processus pour un signal modifié d'excitation

Pour illustrer les avantages des signaux d'apprentissage modifiés et forcés RAS, trois structures de réseaux identiques (modèles NARX 2ème ordre avec 15 neurones cachés) ont été formés en utilisant un signal standard, un signal forcé et un signal modifié, respectivement, et cela a été répété pour trois échantillons de temps différents. La figure 3.17 compare la précision du réseau en utilisant les indices de performance pour les mêmes signaux d'essai de la section 3.3.1. L'indice de performance montre que la précision du modèle du réseau peut être améliorée par l'apprentissage alterné des signaux d'excitation du processus. Alors que la distribution des données de sortie est indépendante de l'échantillon temps, il est clair que l'amélioration de l'indice de performance est plus remarquable pour des échantillons de temps plus larges. Ainsi, il semble que pour les petits intervalles d'échantillons, l'augmentation dans le changement des données de sortie dans la région de gain élevé est suffisamment faible pour le modèle RNA. Pour les longs intervalles d'échantillon, le changement dans les données de sortie est trop important dans les régions du gain élevé pour permettre une interpolation exacte et, par conséquent, il existe une nette dégradation dans la prédiction de précision du modèle dans ces régions.

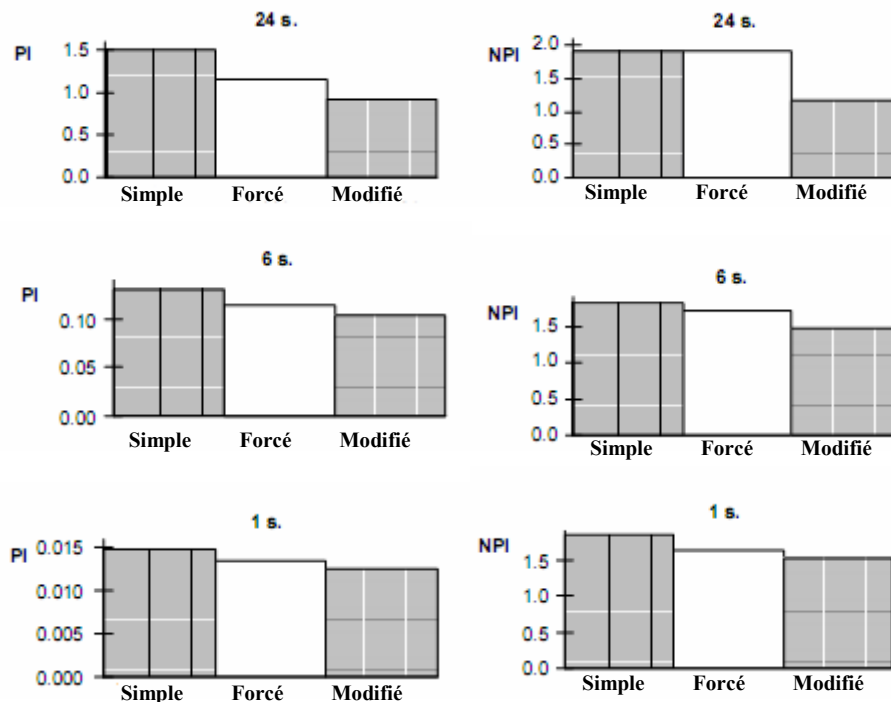


Figure 3.18 : Indice de performance (PI) et indice de performance normalisé (NPI) pour les RNA formés d'ordre 2 en utilisant un signal d'excitation simple, forcé et modifié.

Pour toutes les données d'intervalles d'échantillon, l'amélioration correspond à la provision de données d'apprentissage supplémentaires dans les régions de pH 7 à 10, où les données sont rares, avec le plus faible indice de performance pour chaque échantillon de temps avec le signal RAS modifié. Cette tendance se reflète également dans l'indice de performance normalisé, bien que le signal RAS forcé pour un échantillon de 24 secondes ne montre pas une diminution de l'indice. Ce cas illustre la différence entre les deux indices de performance. Alors que le réseau formé avec le signal RAS forcé a une fonction de pertes plus faible que celle obtenue à l'aide d'un signal d'excitation standard pour quatre des cinq signaux d'essai, la fonction de perte est relativement élevée à l'état statique, ce qui est mis en évidence à l'aide de l'indice de performance normalisé.

L'implémentation du signal d'excitation de tout processus réel doit satisfaire les contraintes sur l'amplitude et la puissance. Les signaux d'excitation forcé et modifié exigent une connaissance parfaite des caractéristiques statiques du processus. Il devrait être possible de concevoir un signal d'excitation pour la distribution des données d'apprentissage dans les régions de données rares indiquées par l'histogramme et ainsi améliorer le modèle neuronal.

4. Simulation de Commande du processus pH.

Ce chapitre décrit les modèles de commande utilisant les RNA pour la simulation du processus pH. Plusieurs modèles de commande sont étudiés et comparés. La commande prédictive à RNA est choisie comme étant une stratégie de commande appropriée et l'implémentation de l'algorithme est décrite. La performance de la commande prédictive à RNA est évaluée et elle est comparée à la commande linéaire classique.

4.1 Examen des stratégies de commande RNA.

Il existe un certain nombre de stratégies de commande fondées sur un certain type de modèle de processus. La plupart de ces commandes utilisent le modèle paramétrique linéaire direct ou inverse. Ces stratégies de commande basées sur le modèle linéaire sont bien établies et bénéficient souvent de la capacité à intégrer directement la robustesse dans la conception de la commande. Il a été établi (NARENDRA et Parthasarathy, 1990) que beaucoup de ces stratégies de commande fondées sur modèles pourraient employer des modèles de réseaux de neurones et, ainsi, bénéficier des propriétés d'approximations non linéaires des RNA. En effet, récemment, des modèles fondés sur les réseaux de neurones ont été utilisés pour la commande des processus chimiques (KESTELYN, 1990; Evans, 1994; Draeger, 1995). La section suivante passe en revue les principaux types de structures de commande basées sur les RNA.

4.1.1 Les modèles Inverses

Les Modèles inverses de processus jouent un rôle central dans certaines structures de commande fondées sur des modèles. Les modèles RNA considérés jusqu'ici sont appelés modèles directs puisque l'information circule dans le sens entrée du processus vers la sortie du processus. En revanche, la direction de la circulation de l'information à travers un modèle inverse de processus est opposée à celle du processus et, par conséquent, un modèle inverse prédit la variable manipulée. Un modèle RNA inverse peut être développé à l'aide d'une structure d'apprentissage directe ou indirecte. Conceptuellement, l'approche la plus simple est d'apprentissage inverse qui est montré dans la Figure 4.1. La variable manipulée du processus est appliquée au processus et la sortie du processus est utilisée, ainsi que les valeurs décalées des E/S du processus, comme entrées du RNA. La sortie du RNA est ensuite

comparée à l'entrée du processus pour produire un signal d'erreur qui est utilisé pour l'apprentissage du réseau de neurones. Intuitivement, cette approche va forcer le RNA à représenter le processus inverse.

Le problème avec l'apprentissage inverse est que l'apprentissage du modèle inverse tente de minimiser l'erreur de la variable manipulée, ce qui n'est pas l'objectif de la commande qui consiste à limiter l'erreur de sortie du processus.

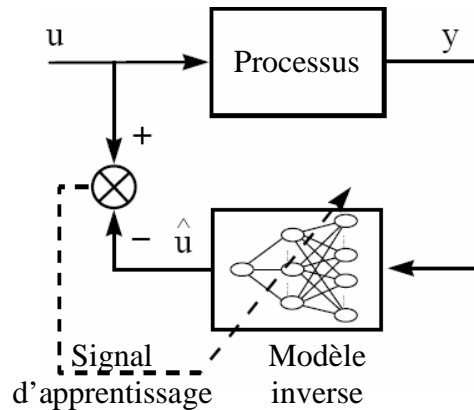


Figure 4.1 : Modèle inverse direct

La méthode spécialisée de modélisation inverse (PSALTIS, 1988) permet de corriger le problème en utilisant l'erreur de sortie du processus, ou bien un modèle direct de processus d'erreur de sortie comme le montre la figure 4.1 pour générer le signal d'erreur. Ce signal d'erreur est transmis en arrière par le modèle direct pour donner l'erreur de la variable manipulée, laquelle est utilisée pour l'apprentissage du modèle inverse. Pour l'apprentissage en ligne, l'erreur de sortie du modèle direct force le modèle inverse dans différentes régions de l'espace d'entrée du processus, ce qui correspond à la région d'exploitation couverte par le point de consigne r . Ce qui est en contradiction avec la méthode directe qui ne permet que l'apprentissage des régions d'exploitation sur lesquelles l'utilisateur a défini une séquence de la variable manipulée. L'utilisation du modèle plutôt que le processus pour générer le signal d'erreur peut être avantageux pour un processus bruité ou un processus réel dont l'usage n'est pas viable (Hunt, 1992).

Il existe d'autres problèmes potentiels associés à l'élaboration d'un modèle inverse d'un système non-linéaire. Par exemple, certaines classes de systèmes non linéaires ne sont pas inversibles (Economou et Morari, 1986). Le système non-linéaire doit avoir un mapping direct pour que l'inverse soit unique. Si ce n'est pas le cas, alors un modèle inverse incorrect peut avoir lieu, bien qu'il soit possible de diriger l'apprentissage du modèle inverse vers un

ensemble de modèles pseudo-inverses avec des propriétés désirées (JORDAN et Rumelhart, 1991). Enfin, l'inverse de certains systèmes non linéaires peut être instable, et une structure de commande fondée sur un modèle inverse instable est à éviter (NAHAS, 1992).

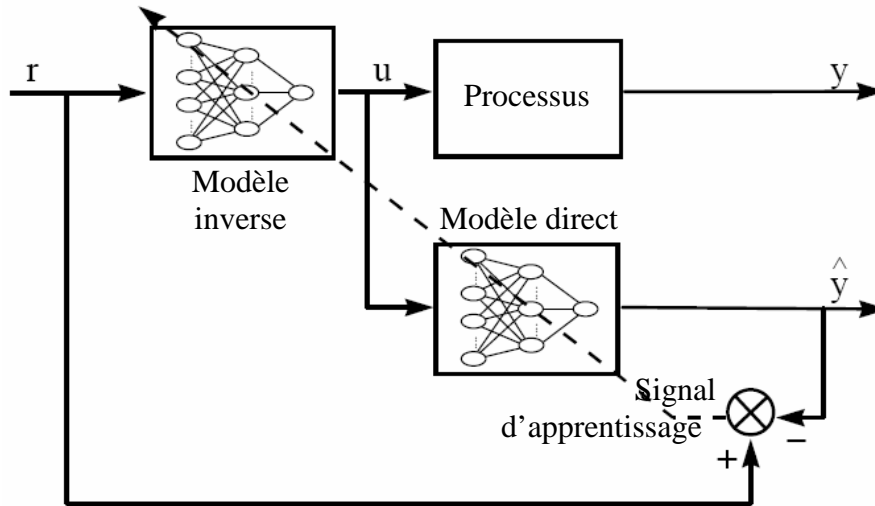


Figure 4.2 : Modèle inverse spécialisé

Probablement la plus simple utilisation d'un modèle inverse de processus est de le mettre devant le processus (Figure 4.2). Cette approche est appelée commande inverse directe et a été utilisée principalement dans les applications robotiques. Toutefois, l'absence de contre-réaction dans la structure de commande inverse directe de commande se traduit par un manque de robustesse pour le cas pratique d'un modèle inverse imparfait (Hunt, 1992). Il existe plusieurs structures de commande linéaire intégrant les modèles inverses qui ont une robustesse supérieure à la commande inverse directe et certains d'entre eux, décrits ci-dessous, ont été élargis à la commande non-linéaire.

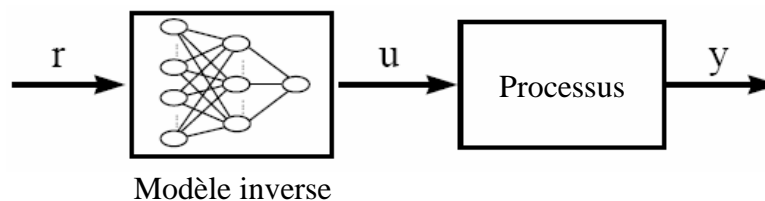


Figure 4.3: Commande inverse directe

4.1.2 Commande à modèle interne

La commande linéaire à modèle interne (IMC) (Garcia et Morari, 1982) a été étudiée de manière approfondie et des résultats de robustesse et de stabilité ont été prouvés. Pour les

systèmes stables en boucle ouverte certains de ces résultats peuvent être étendus à la commande non linéaire à modèle interne (Economou et Morari, 1986). L'utilisation de réseaux neuronaux dans la structure IMC a été proposée par plusieurs chercheurs (BHAT et McAvoy, 1990; Hunt et SBARBARO, 1991) et a été implémentée pour la simulation de processus (NAHAS, 1992; Hunt et SBARBARO, 1992).

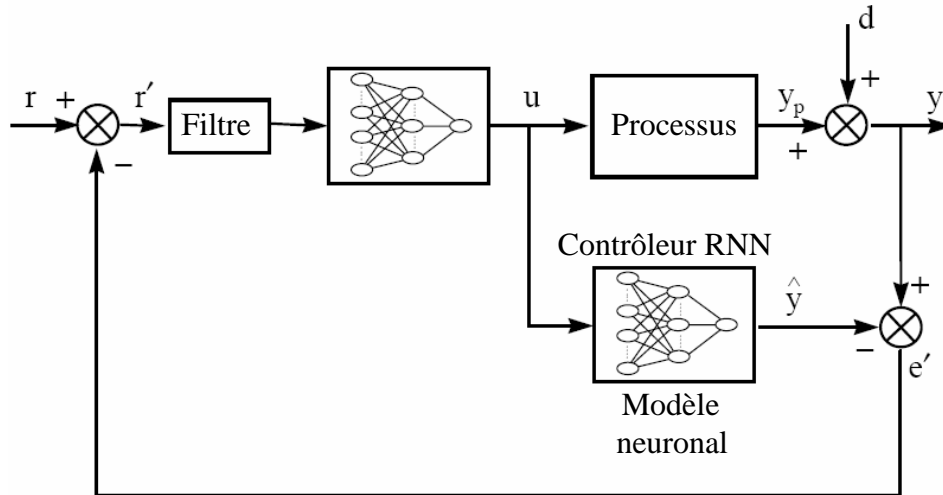


Figure 4.4 : Commande par modèle interne

La structure de commande (Figure 4.4) utilise à la fois un modèle de processus RNA direct et inverse. Le modèle direct est mis en parallèle avec le processus et la différence entre les deux, e' , est utilisée à des fins de feed-back. Ce signal feed-back est activé par le contrôleur RNA qui est conçu pour être un modèle inverse du processus.

4.1.3 La commande à modèle de référence

Elle a été largement utilisée pour la commande adaptative linéaire (et Åström WITTENMARK, 1989) et il a été prouvé que la structure de commande pourrait être utilisée pour la commande non-linéaire en employant les réseaux de neurones (et NARENDRA Parthasarathy, 1990). La structure de la commande à modèle de référence (Figure 4.5) intègre un modèle direct et un contrôleur RNA. Le but de la commande est de forcer la sortie du processus à suivre de manière asymptotique la sortie d'un modèle de référence. L'erreur entre le modèle de référence et la sortie du processus, e'' , est propagée à l'arrière par le modèle direct, qui est utilisé pour ajuster les poids du contrôleur RNA, à la manière de l'apprentissage inverse spécialisé. L'apprentissage du modèle RNA direct se fait en utilisant l'erreur processus / modèle, e' .

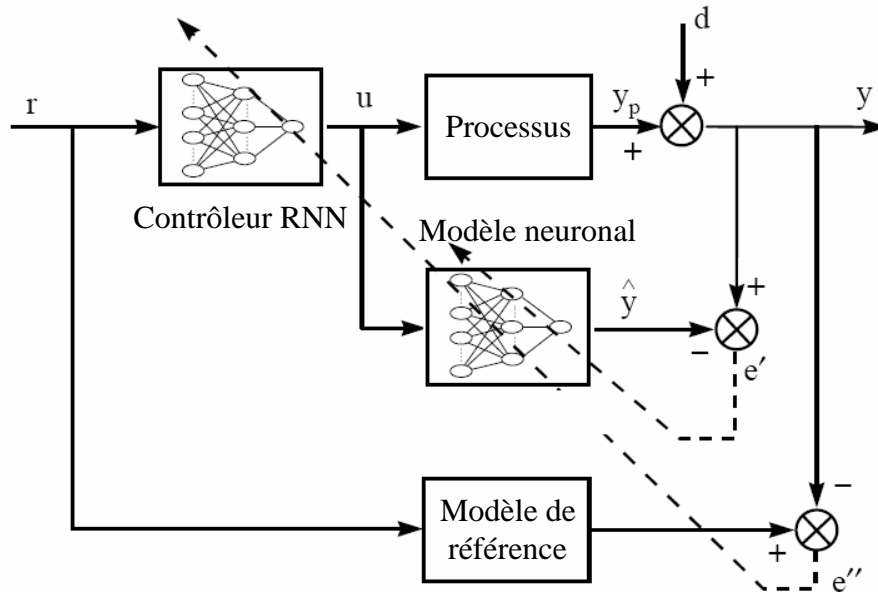


Figure 4.5 : Commande par modèle de référence

La performance de la commande à modèle de référence dépend beaucoup du choix du modèle de référence. Les applications de la commande à modèle de référence RNA ne sont pas répandues.

4.1.4 Commande prédictive

Il a été établi que la commande Prédictive à Modèle Linéaire (LMPC) est la plus importante technologie de commande pour les processus industriels depuis la commande PID (VANDOREN, 1997). Certes, le LMPC est de loin le schéma de commande le plus réussi pour les processus industriels (Garcia, 1989). Les raisons de ce succès résident dans la facilité relative avec laquelle le LMPC peut traiter les caractéristiques des procédés, comme le temps de retard, l'entrée, la sortie, l'état des contraintes et l'interaction des variables. Le LMPC intègre une classe d'algorithmes de commande, où un modèle dynamique linéaire est utilisé pour prévoir et optimiser les performances du processus de fonctionnement, qui inclue la commande prédictive généralisées (RICHALET, 1978).

Récemment, diverses formes du MPC ont été étendues aux schémas de commande non-linéaire RAWLINGS (1994) et HENSON (1998).

La Figure 4.6 montre un réseau de neurones MPC qui, contrairement aux autres stratégies de commande, n'intègre pas un modèle inverse de processus. L'optimiseur non-linéaire dans un MPC RNA est utilisé pour sélectionner la variable manipulée qui minimise une fonction de coût quadratique. Pour ce faire, l'optimiseur non-linéaire utilise le modèle RNA du

processus pour prédire les réponses futures possibles du processus aux différentes séquences futures possibles de la variable manipulée et aux perturbations actuelles mesurées. En utilisant le modèle RNA pour prédire plusieurs étapes à venir, le système de commande peut anticiper la trajectoire du processus et compenser les perturbations mesurées avant que leur impact sur la sortie du processus ne soit détecté.

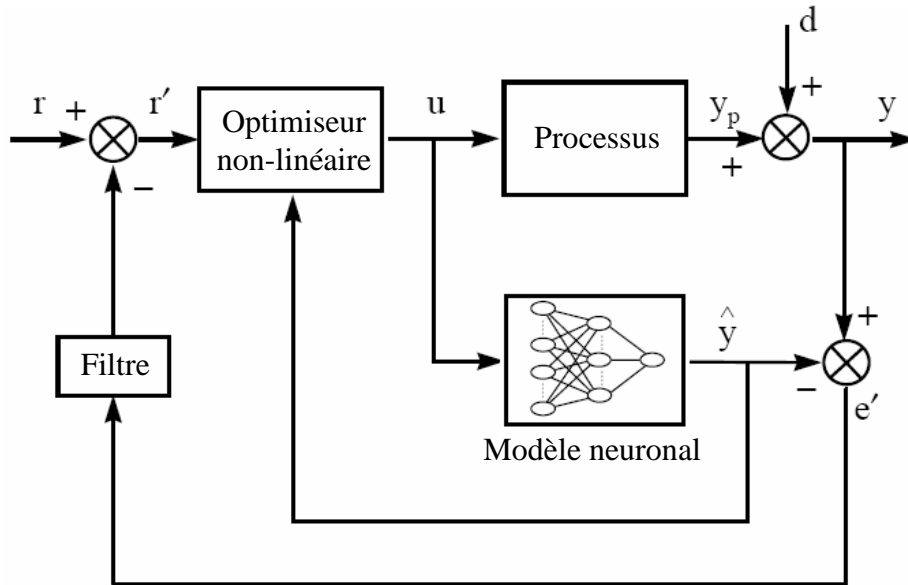


Figure 4.6 : Commande prédictive à RNA

L'optimiseur non-linéaire peut aussi tenir compte des contraintes du processus, qui est une caractéristique pratique importante du MPC puisque tous les processus réels sont soumis à certaines contraintes. En incluant les contraintes dans l'optimisation, la commande peut prédire les violations futures de contrainte et réagir en conséquence.

4.2 La commande linéaire de simulation du processus

L'objectif principal des stratégies de commande basée sur des modèles RNA est d'améliorer les performances de la commande linéaire classique. La procédure de conception et d'implémentation de contrôleurs non linéaires basés sur des modèles RNA est plus compliquée que celle des contrôleurs linéaires. Par conséquent, il est impératif d'établir si cet effort dans la méthode de conception est justifié en comparant les performances des contrôleurs linéaires avec celles des contrôleurs non linéaires.

Le processus est commandé en utilisant le PID standard, largement utilisé dans les schémas de commande des processus industriels.

4.2.1 Commande PID.

La simulation du processus pH a été réalisée au moyen de la commande linéaire afin de comparer les résultats avec ceux obtenus par la commande basée sur les RNA.

La vitesse de l'algorithme PID est implémentée comme suit

$$\Delta u(n) = K_c \left(\left(1 + \frac{T}{\tau_i} + \frac{\tau_D}{T}\right) e(n) - \left(1 + \frac{2\tau_D}{T}\right) e(n-1) + \frac{\tau_D}{T} e(n-2) \right) \quad (4.1)$$

où $\Delta u(n) = u(n) - u(n-1)$ est le changement de la sortie de commande, e est l'erreur de sortie du processus, T est l'intervalle d'échantillonnage, K_c est le gain proportionnel, τ_i temps d'action d'intégration, τ_D le temps de dérivée.

Parce que le processus est inverse l'erreur de sortie est calculée ainsi

$$e(n) = y(n) - r(n) \quad (4.2)$$

La technique de réglage de Ziegler-Nichols (1942) a été utilisée pour fournir une première estimation des paramètres de commande PID (K_c , τ_i et τ_D), qui sont ensuite ajustés par la méthode essais et erreurs.

La technique Ziegler-Nicholas est une méthode de réglage en boucle fermée qui a été implémentée comme suit

1. La perturbation du flux est fixée à sa valeur moyenne.
2. Avec le processus en boucle ouverte, la variable manipulée a été ajustée manuellement de façon à ramener le processus au point de fonctionnement statique désiré.
3. En utilisant la commande P, avec le point de consigne égal au point de fonctionnement et $u(0)$ dans l'équation 5.3 déterminé à partir de l'étape 2, la boucle est fermée.
4. Un signal carré a été utilisé pour appliquer de faibles variations au point de consigne (généralement 0,1 unités de pH).
5. Le gain proportionnel est progressivement augmenté jusqu'à ce que le processus oscille. C'est le gain recherché K_U , et la fréquence d'oscillation est la fréquence ω_{CO} .

Les paramètres PID recommandés sont calculés à partir de K_U et ω_{CO} en utilisant des expressions proposées par Ziegler-Nicholas (1942).

En utilisant la formule de Ziegler-Nichols on obtient $K_C=0,11$, $\tau_i=4,31s$ pour la commande PI ; $K_C=0,177$, $\tau_i=2s$, $\tau_D=1s$ pour la commande PID.

$$K_c = R_{p1}/R_{p2} = 0.177$$

$$R_{p1} = 10 \text{ K}\Omega$$

$R_{p2} = 2.2 \text{ K}\Omega$ variable

$\tau_i = R_i C_i$

$C_i = 100 \text{ nF}$

$\tau_i = 2\text{s}$

$R_i = 20 \text{ M}\Omega$

$\tau_D = R_D C_D = 1\text{s}$

$C_D = 100 \text{ nF}$

$\tau_i = 2\text{s}$

$R_i = 10 \text{ M}\Omega$

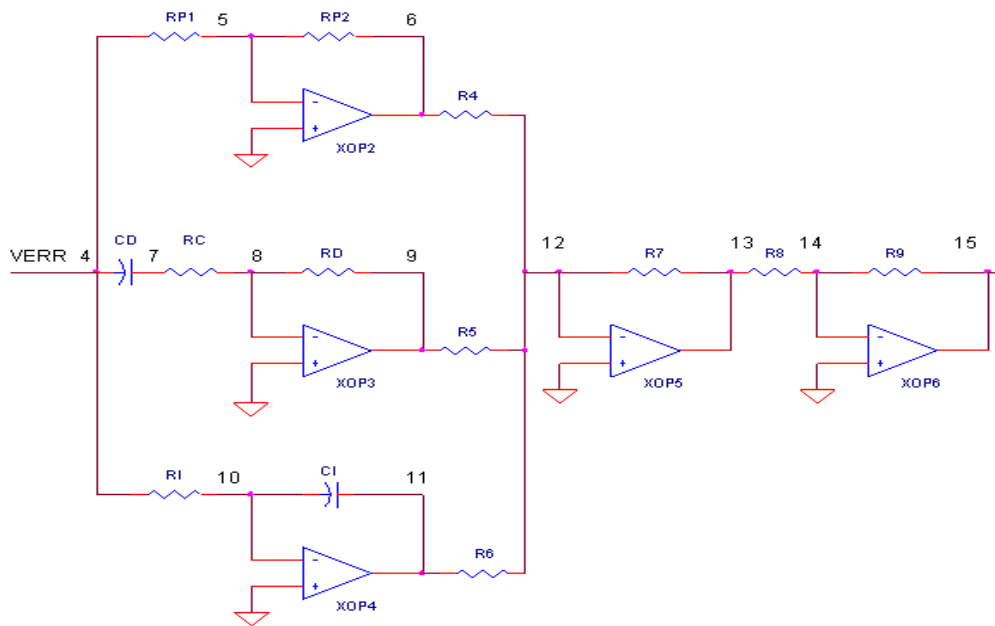


Figure 4.7 : schéma électrique d'un régulateur PID

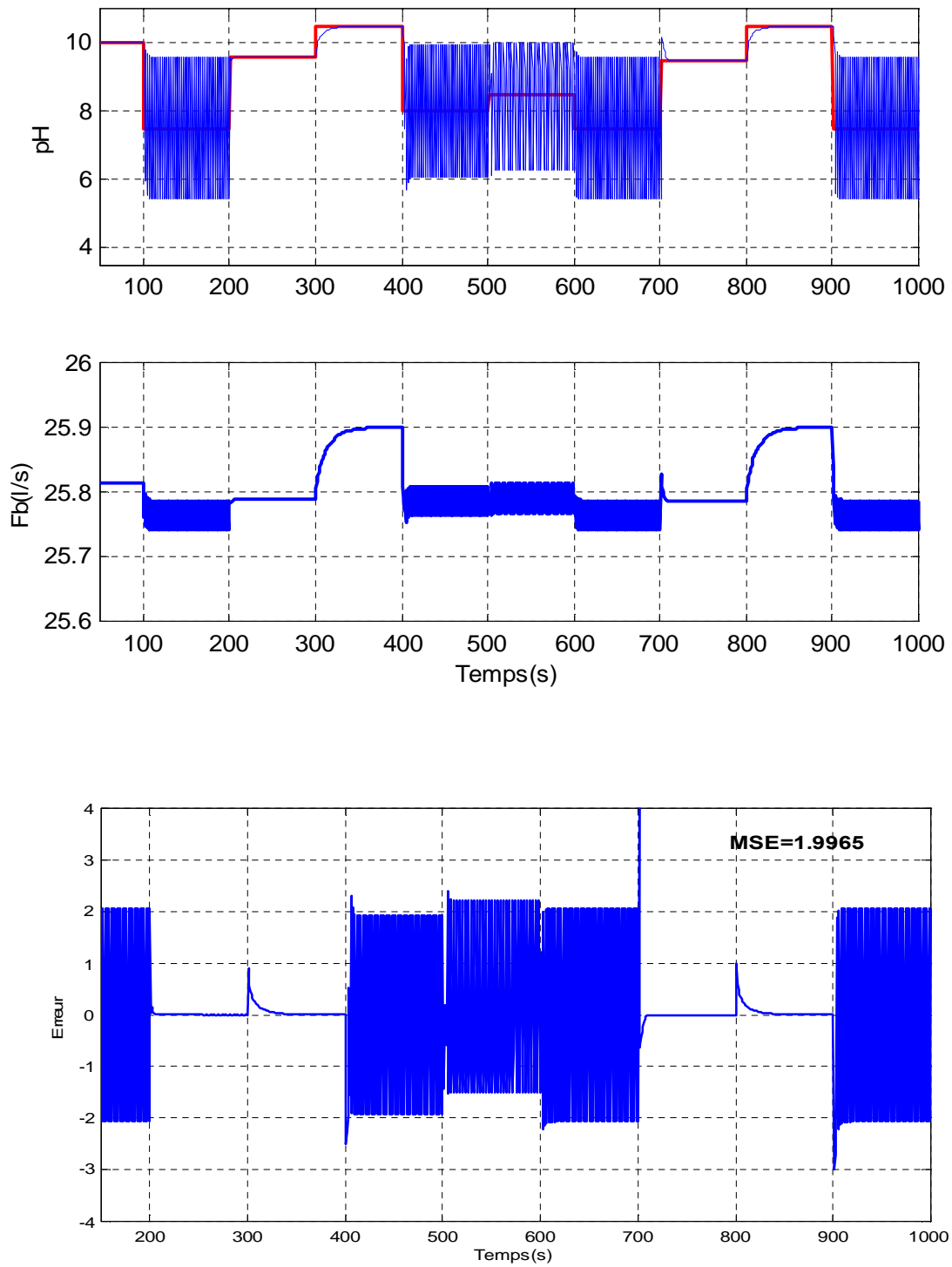


Figure 4.8 Commande PID aux points de consigne pH=5,5 ... pH=10,5

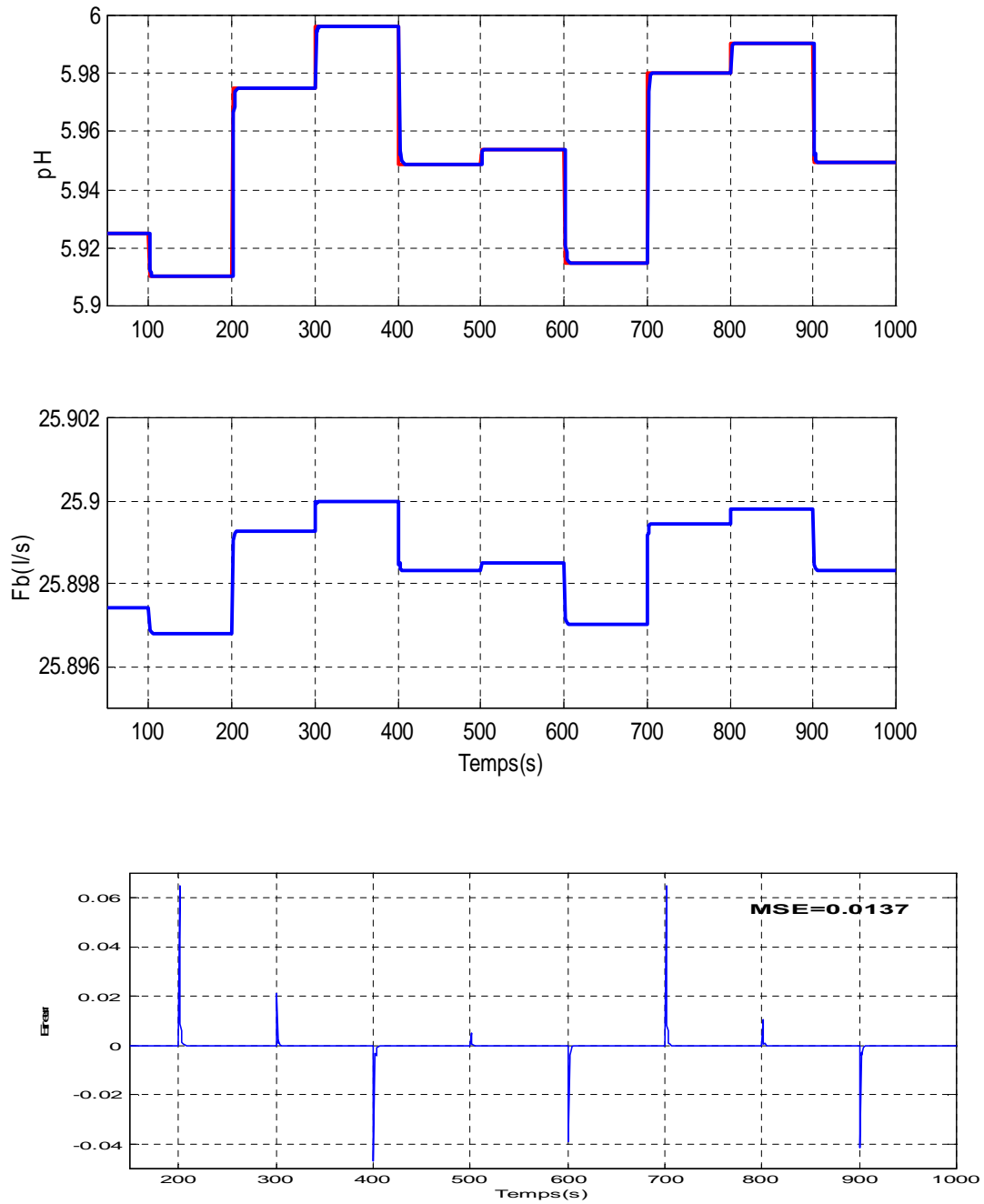


Figure 4.9 Commande PID au point de consigne pH=5,95

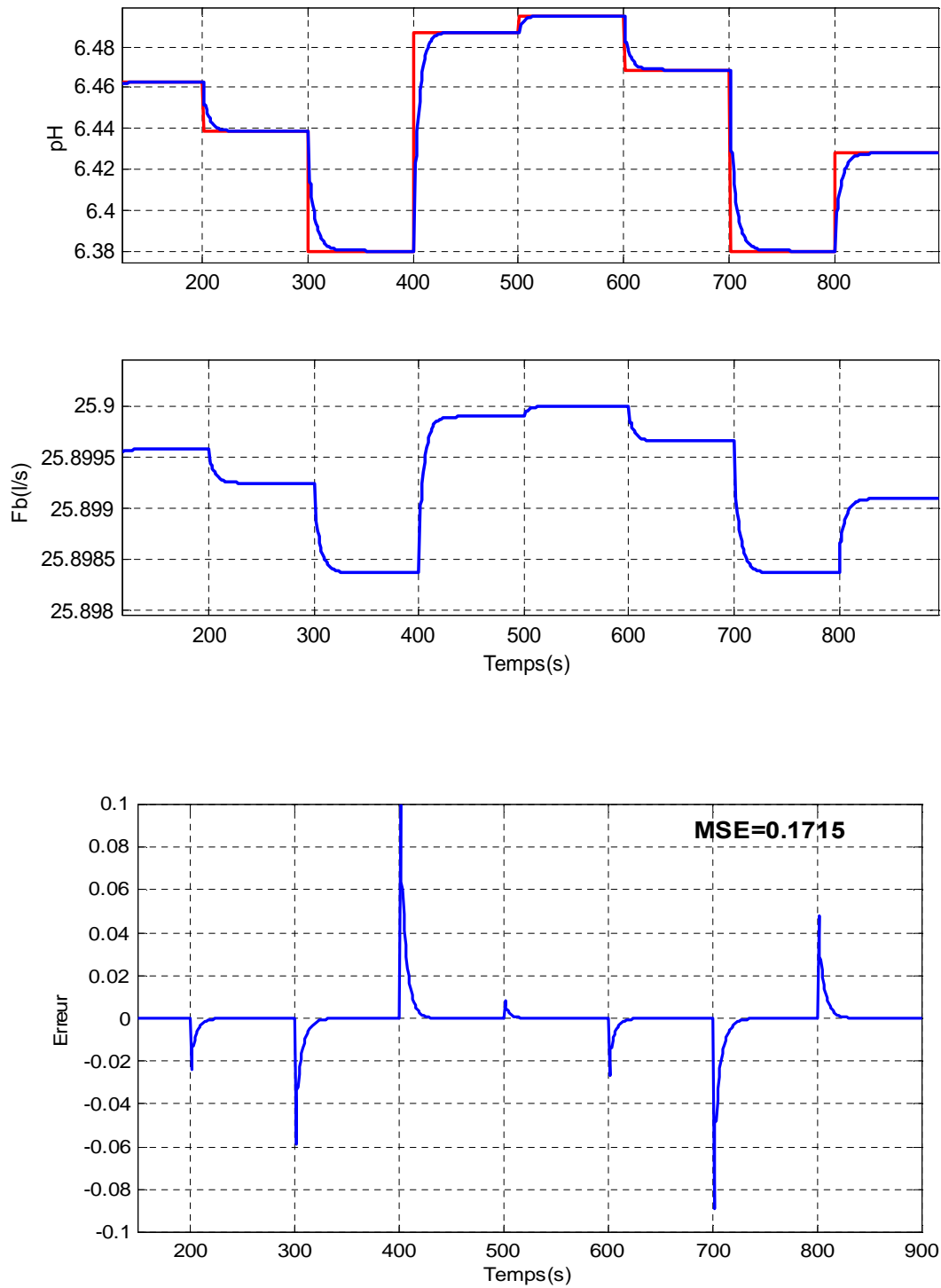


Figure 4.10 Commande PID au point de consigne pH=6,44

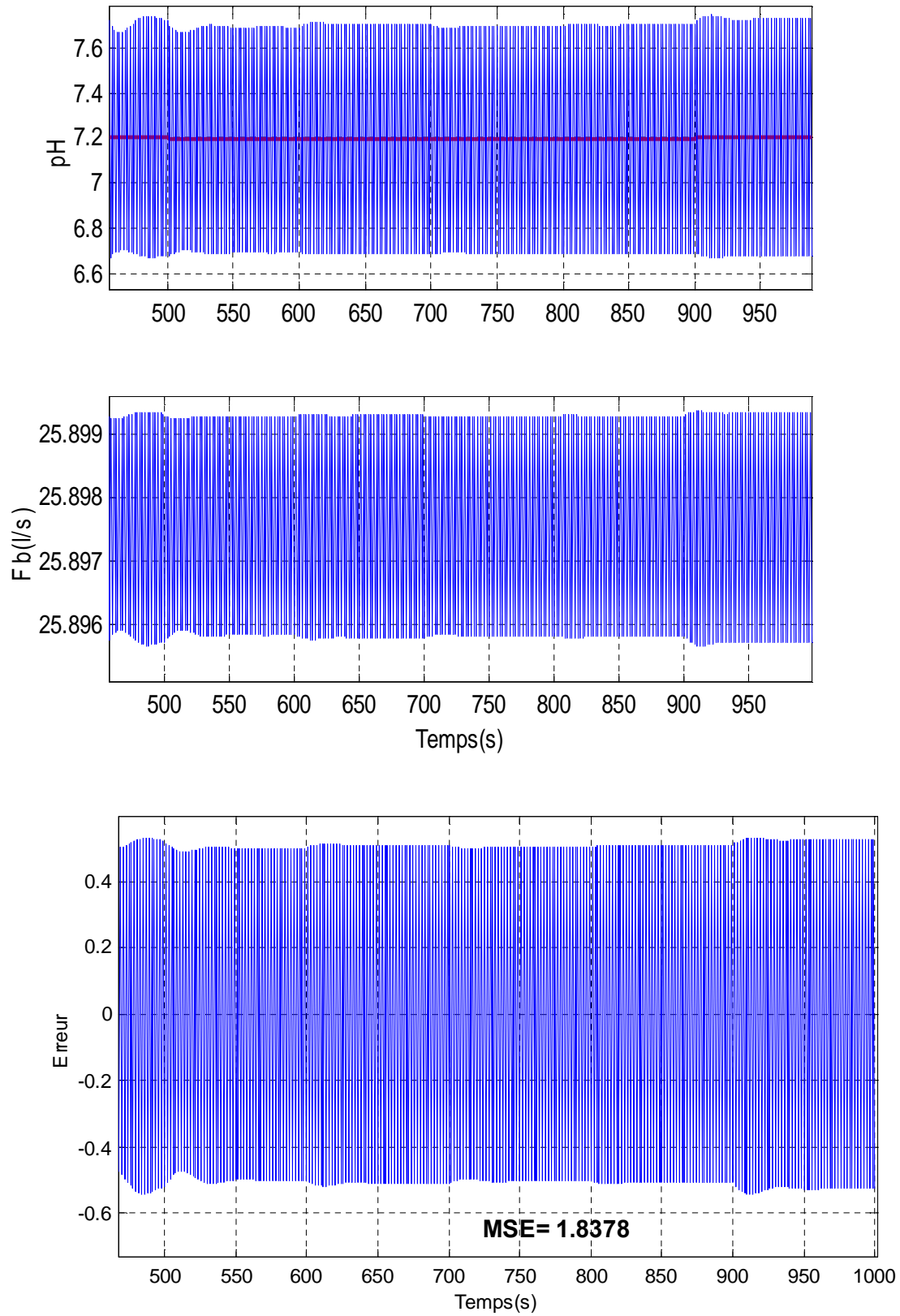


Figure 4.11 Commande PID au point de consigne pH=7,2

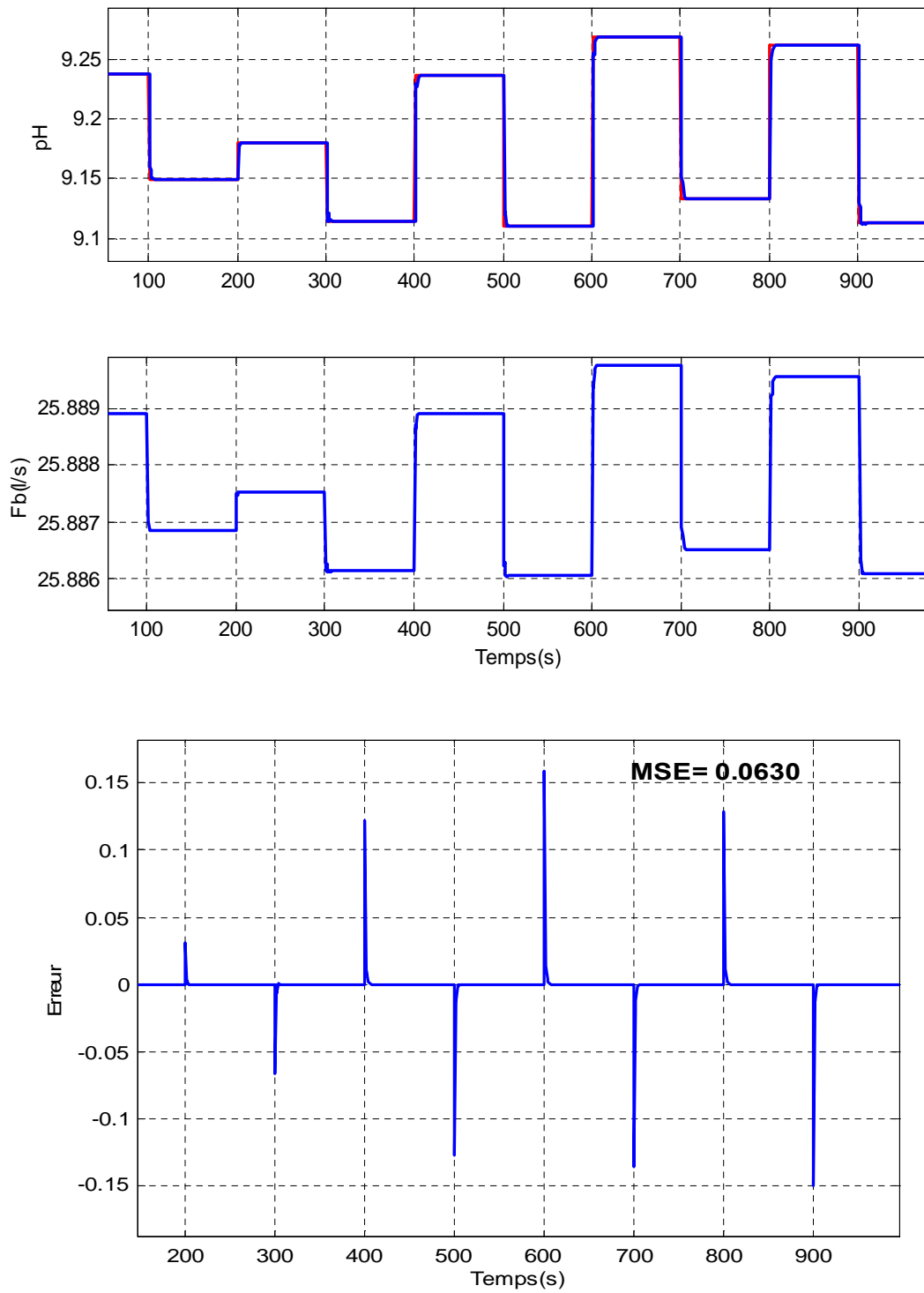


Figure 4.12 Commande PID au point de consigne pH=9,2

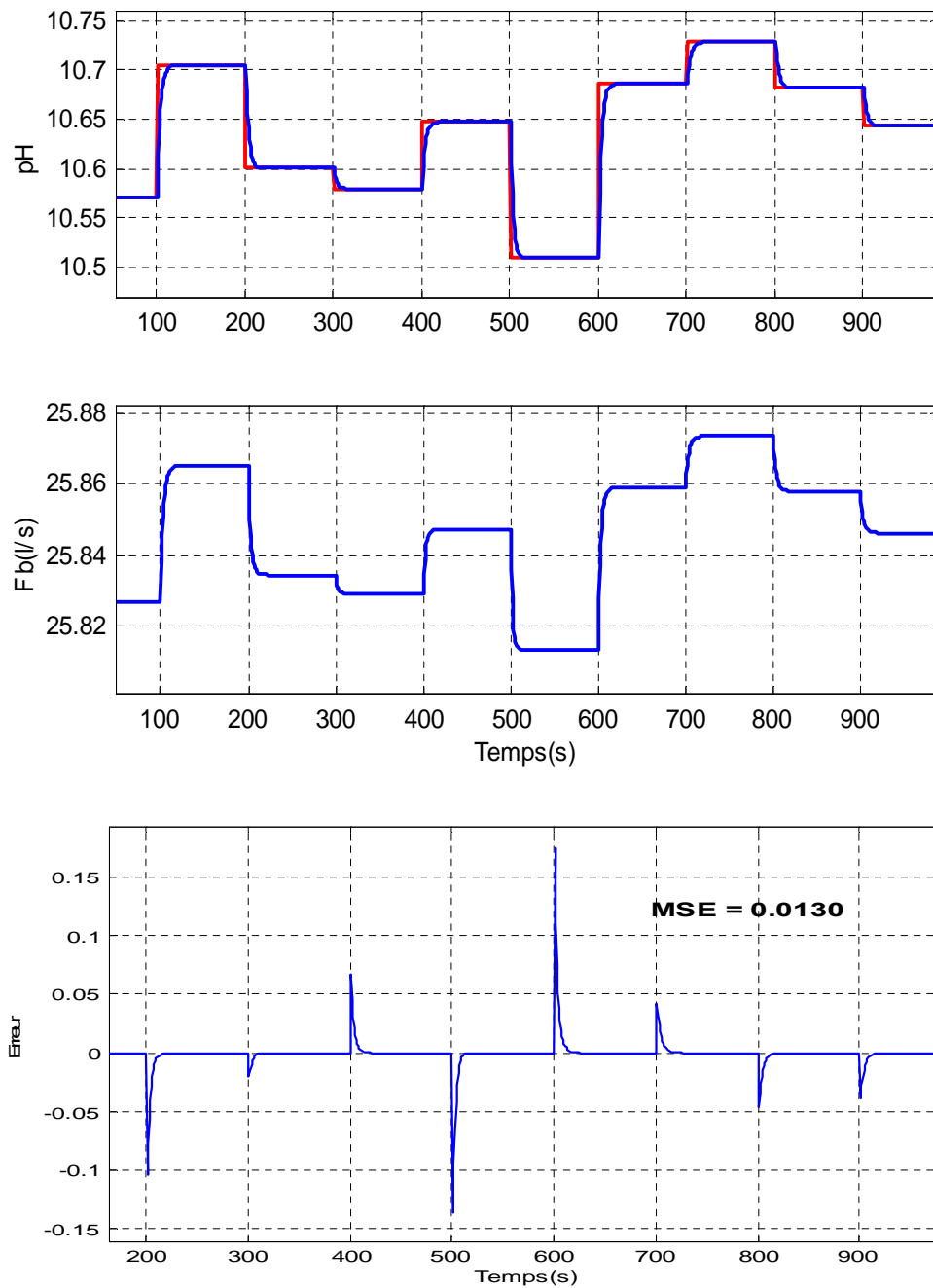


Figure 4.13 Commande PID au point de consigne pH=10,6

Les résultats décrits ici sont pour les commandes PID qui ont été ajustées pour les points de consigne pH=5,95 ; 6,44 ; 7,2 ; 9,2 ; 10,6.

La commande PID a réduit les oscillations pour les points de 5,95 et 6,44, mais a été instable aux points 7,2. Quand la variable manipulée varie de zéro à la valeur maximale, le réglage fin des paramètres du contrôleur pourrait améliorer la commande de l'un des points de consigne mais toujours au détriment de la détérioration de commande d'autres points.

4.3 La commande prédictive RNA de simulation du processus pH.

Le RNA MPC est étudié pour les raisons suivantes:

- RNA MPC ne nécessite pas un modèle inverse de processus, qui pose souvent problème.
- MPC a certaines propriétés importantes, qui ont été discutées dans la section 4.1.4, et qui le rendent particulièrement adapté pour la commande du pH. En particulier, la capacité des MPC à gérer les temps de retard et d'anticiper les effets des perturbations mesurées a été jugée efficace.
- Pour les stratégies de commande à modèle linéaire, le MPC a été d'un grand succès dans son adoption par les processus industriels et il a été récemment étendu avec succès aux processus non linéaires.

4.3.1 La structure de commande prédictive à RNA

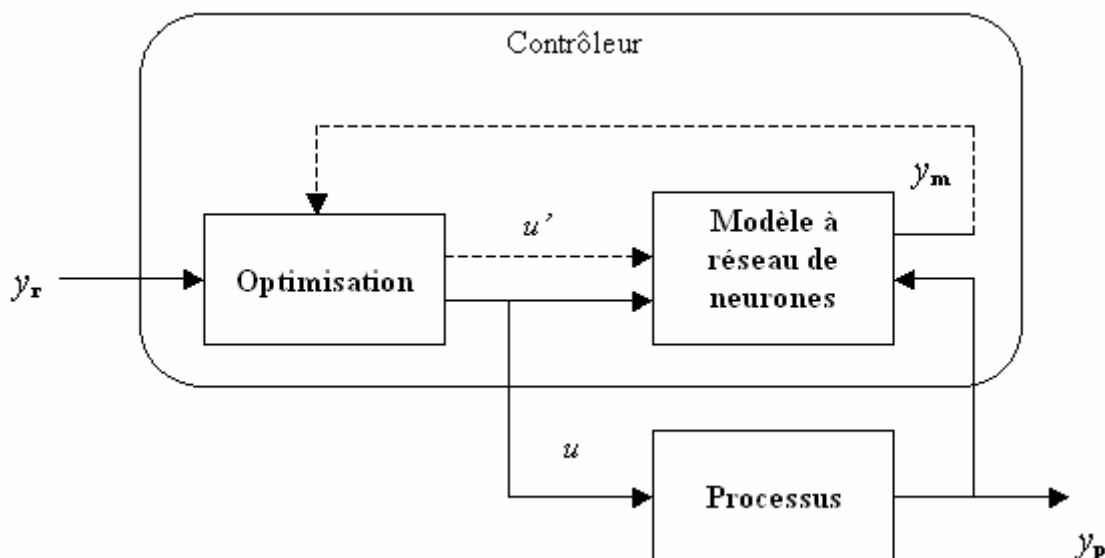


Figure 4.14 : Structure de la commande prédictive

La philosophie générale de RNA MPC est identique à celui de LMPC. Le contrôleur détermine un ensemble de changements futurs de la variable manipulée qui minimise une fonction de coût sur un horizon de prédiction, sujets à des contraintes d'entrée et de sortie. La fonction de coût généralement comprend la somme des carrés des erreurs entre les sorties de prédiction et les valeurs de consigne évaluées sur l'horizon de prédiction, et généralement comprend un terme qui pénalise le taux de changement de la variable manipulée. Pour une telle fonction de coût, le problème du MPC peut être posé comme suit :

$$\min_{\mathbf{u}(t)} J = \sum_{i=P1}^{i=P2} (\tilde{y}(t+i) - r(t+i))^2 + \lambda \sum_{i=1}^{i=C} (u(t+i-1) - u(t+i-2))^2 \quad (4.3)$$

$$\mathbf{u}(t) = [u(t) \dots u(t+C-1)]^T \quad (4.4)$$

$$u(t+i) = u(t+C-1) \quad , i \geq C \quad (4.5)$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max} \quad (4.6)$$

où J est la fonction de coût à minimiser, P1 à P2 définit l'horizon de prédiction, C est l'horizon de commande (C • P2), $\tilde{y}(t+i)$ est la sortie du processus prédite à l'instant t + i, u (t) est le vecteur des valeurs de la variable manipulée de longueur C et λ est un coefficient de pondération. Le variable indépendante dans J est u (t) dont les éléments sont soumis à la contrainte de l'inégalité 4.6.

Des corrections doivent être apportées à la sortie du modèle pour tenir compte du déséquilibre processus/modèle et des perturbations non mesurées (RAWLINGS, 1994), et cela peut être fait avec une perturbation additive,

$$\tilde{y}(t+i) = \hat{y}(t+i) + e'(t) \quad (4.7)$$

où $\hat{y}(t+i)$ est le futur pas i du modèle RNA de prédiction.

Une approche simple, qui est adoptée ici, est d'utiliser le déséquilibre processus/modèle pour évaluer la perturbation (HENSON et SEBORG, 1993)...

$$e'(t) = y(t) - \hat{y}(t) \quad (4.8)$$

En introduisant la valeur de consigne modifiée, r'(t)...

$$r'(t+i) = r(t+i) - e'(t) \quad (4.9)$$

et en combinant les équations 4.3, 4.7 et 4.9

$$\min_{u(t)} J = \sum_{i=P1}^{i=P2} (\hat{y}(t+i) - r'(t+i))^2 + \lambda \sum_{i=1}^{i=C} (u(t+i-1) - u(t+i-2))^2 \quad (4.10)$$

qui est la forme de MPC utilisé dans ce travail dont la structure est illustrée à la figure 4.6.

D'autres aspects importants de l'algorithme de commande du MPC sont

- La modification du point de consigne, en utilisant le déséquilibre processus/modèle (équations 4.8 et 4.9), permet au schéma MPC d'éliminer le décalage de l'état de repos quelle que soit le déséquilibre processus/modèle (RAWLINGS, 1994).

$$r'(t+i) = r(t+i) - D(e'(t)) \quad (4.11)$$

- A chaque intervalle d'échantillonnage, bien que la solution de l'équation 4.10 donne un vecteur de commande future $u(t)$, seul le premier $u(t)$ est implémenté. Les autres éléments du vecteur solution ne sont pas pris en considération.

- En minimisant la fonction de coût pour la situation habituelle $C < P2$, l'action de commande est supposé rester constante au-delà de l'horizon de commande tel que spécifié par l'équation 4.10. Aussi, si les futurs changements du point de consigne pour l'horizon de prédiction sont inconnus, alors le point de consigne actuel est utilisé.

- Le commande possède quatre paramètres de réglage $P1$, $P2$, C et λ . $P1$ doit être réglé sur le temps mort de la variable manipulée, k , parce que tous $\hat{y}(t+i|t)$ avec $i < k$ ne doit pas être inclus dans la première partie de la fonction de coût puisque $u(t)$ n'a pas d'effet sur eux. Ainsi, les principaux paramètres de réglage sont $P2$, C et λ .

Certains détails d'implémentation de l'algorithme MPC employé dans cette recherche sont:

- Le décalage processus / modèle est filtré par un simple filtre passe-bas de premier ordre. Le coefficient du filtre passe-bas α , a été fixé à 0,9, ce qui correspond à une constante de temps d'environ 10 intervalles d'échantillon.

- La fonction de coût de l'équation 4.10 est de la forme...

$$J = \sum_{i=P1}^{i=P2} \Delta y^2 + \lambda \sum_{i=1}^{i=C} \Delta u^2 \quad (4.12)$$

où Δy est l'erreur de sortie / consigne modifiée et Δu est le taux de changement de la variable manipulée. Pour $\lambda = 1$, les deux parties de la fonction de coût peuvent être de différents ordres de grandeur et leurs valeurs seront également dépendantes de $P2$ et C respectivement. Par conséquent, avant la minimisation de la fonction de coût, le coût du mouvement de commande a été réduit à l'échelle $[0,1]$ en divisant par $C \times \Delta U_{max}$, où

ΔU_{max} est la plage de la variable manipulée, et le coût de l'erreur de sortie a été divisé par $P2-P1 + 1$. Le coût de l'erreur de sortie pourrait ne pas être réduit à une plage fixe puisque la plage Δy est inconnue. Cette réduction de l'échelle permet d'obtenir des amplitudes des deux parties de la fonction de coût indépendantes des horizons de prédiction et de commande, et la valeur maximale du coût de mouvement de commande est l'unité. Il est donc plus facile de comparer les résultats de commande et d'estimer l'effet de λ .

4.3.2 Optimisation de la fonction de coût

L'équation 4.8 est un problème d'optimisation non-linéaire lorsque le modèle (ou une quelconque des contraintes) sont non-linéaires, comme c'est le cas pour RNA MPC. Puisque le modèle est non linéaire, le problème d'optimisation peut être non-convexe. Il existe des méthodes de programmation non-linéaires qui peuvent être appliquées au problème d'optimisation non-linéaire du MPC à chaque étape (RAWLINGS, 1994), mais aucune n'est pleinement satisfaisante, tel que conclu par HENSON (1998).

La méthode d'optimisation de recherche directe Nelder-Mead (Nelder et Mead, 1965) a été utilisée dans ce travail. C'est un algorithme d'optimisation multi-dimensionnel qui n'est pas très efficace en termes de nombre d'évaluations de la fonction de coût (Dennis et Schnabel, 1983). Toutefois, parce que le Nelder-Mead algorithme n'utilise pas de dérivées, il convient pour les problèmes non-linéaires ayant des discontinuités ou un niveau de bruit élevé (PRESSE, 1992; MATLAB, 1999). L'algorithme recherche un espace à N dimensions en utilisant une figure géométrique, appelé simplexe, avec N + 1 sommets. A chaque itération l'algorithme déplace un des sommets dans une direction de descente en utilisant une parmi plusieurs opérations géométriques sur le simplexe. L'initialisation de l'algorithme exige un point de départ, un vecteur de recherche de tolérance et une valeur de fonction de tolérance. Le vecteur de tolérance et la valeur de la fonction de tolérance sont fixés à 10^{-4} .

L'un des inconvénients de l'emploi de l'algorithme Nelder-Mead pour la minimisation de la fonction de coût du MPC (équation 4.10) est que la contrainte (équation 4.6) doit être manipulée indirectement. Cela peut être fait en appliquant une fonction bornée non-linéaire inversible à l'espace de recherche et, à cette fin la fonction sigmoïde est utilisée (POTTM RNA et SEBORG, 1997)...

$$\mathbf{u} = f(\mathbf{u}') = \frac{\mathbf{u}_{max} - \mathbf{u}_{min}}{1 + \exp(-\mathbf{u}')} + \mathbf{u}_{min} \quad (4.13)$$

où u' est le vecteur de recherche sans contrainte et, maintenant satisfait l'équation 4.6. Ainsi, l'algorithme Nelder-Mead recherche l'espace de dimension C en ajustant le vecteur sans contrainte de recherche u' . Mais avant chaque évaluation de la fonction de coût, le vecteur est transformé en un espace de contrainte par l'équation 4.13. Puisque $f(\dots)$ est inversible, tous les vecteurs solutions sans contrainte u' , donnent un résultat unique de contrainte.

La convergence est la question essentielle quant à l'utilisation de l'algorithme Nelder-Mead de minimisation de l'équation 4.10. L'algorithme Nelder-Mead peut converger vers un local, plutôt que vers un minimum global de la fonction non-linéaire de coût (équation 4.10). Toutefois, c'est le cas pour toutes les méthodes de programmation non linéaire, car la convergence vers un minimum global pour une fonction générale non-linéaire ne peut pas être garanti (Nash, 1998). Pour augmenter la probabilité que la solution serait proche du minimum global, l'algorithme Nelder-Mead est initialisé de façon séquentielle à partir de points qui ont été distribués dans tout l'espace de recherche. Un treillis de points initiaux de dimension C est formé avec des sommets n_C uniformément répartis de chaque côté du treillis. Ce qui donne $(n_C)^C$ initiaux points de recherche et le minimum global est affecté au minimum des $(n_C)^C$ minimisations individuelles.

L'algorithme Nelder-Mead généralement converge après moins de 100 itérations dans la phase initiale d'expérimentation et a peu de chances de converger vers une solution après 1000 itérations. Ainsi, pour chaque minimisation individuelle la recherche est terminée et le résultat n'est pas pris en compte si un des critères de convergence n'est pas satisfait après 1000 itérations. Une série d'expériences a été réalisée sur des surfaces de coût à 1 et 2 dimensions obtenues par l'évaluation de la simulation de la fonction de coût à des points uniformément répartis sur la plage de fonctionnement. Le minimum global des surfaces de coût, qui ne contiennent pas souvent un minimum local, a été obtenu par une recherche exhaustive. Dans tous les cas examinés, le minimum global de ces points était proche du minimum trouvé par la procédure séquentielle Nelder-Mead de recherche pour $n_C \cdot 4$ et, par conséquent, $n_C = 4$ a été adopté par la suite pour l'étude de simulation du MPC.

4.3.3 Sommaire de l'algorithme MPC

L'algorithme du MPC, exécute les instructions suivantes à chaque intervalle d'échantillon...

1. Prélever la réponse du processus $y(t)$, et calculer l'erreur processus/modèle $e'(t)$, entre $y(t)$ et $\hat{y}(t|t-1)$. Filtrer l'erreur et calculer la séquence des points références modifiés au futur au-dessus de l'horizon de prévision P1 à P2.

2. Utiliser le modèle RNA du processus pour prévoir les prochaines valeurs de sortie du processus.
3. Obtenir une solution pour l'équation 4.10.
4. Mettre en application $u(t)$.

4.3.4 L'horizon de commande

Employant un horizon de commande supérieur à un n 'a pas de justification en termes de taux de changement de l'action de commande. La plupart des articles publiés sur le MPC RNA n'ont pas utilisé un horizon de commande supérieur à un. Une étude (POTTM RNA et SEBORG, 1998) appliquée à un MPC RNA au processus de neutralisation pH CSTR, a également conclu que l'horizon de commande supérieur à 1 donne une action de commande très agressive.

4.3.5 Construction du modèle CSTR:

En utilisant le logiciel Simulink installé dans MATLAB 7.0, on peut modeler, simuler, et analyser les systèmes dynamiques linéaires et non-linéaires.

En première phase, on a formé les équations 3.11,3.12,3.13 et 3.14 qui sont les équations qui définissent la dynamique du système.

4.3.5.1 Conception du bloc calculant ζ et ξ :

Ce bloc contient les deux équations différentielles 3.12 et 3.13, la seule entrée est F_b est il a deux sorties qui sont ζ et ξ .

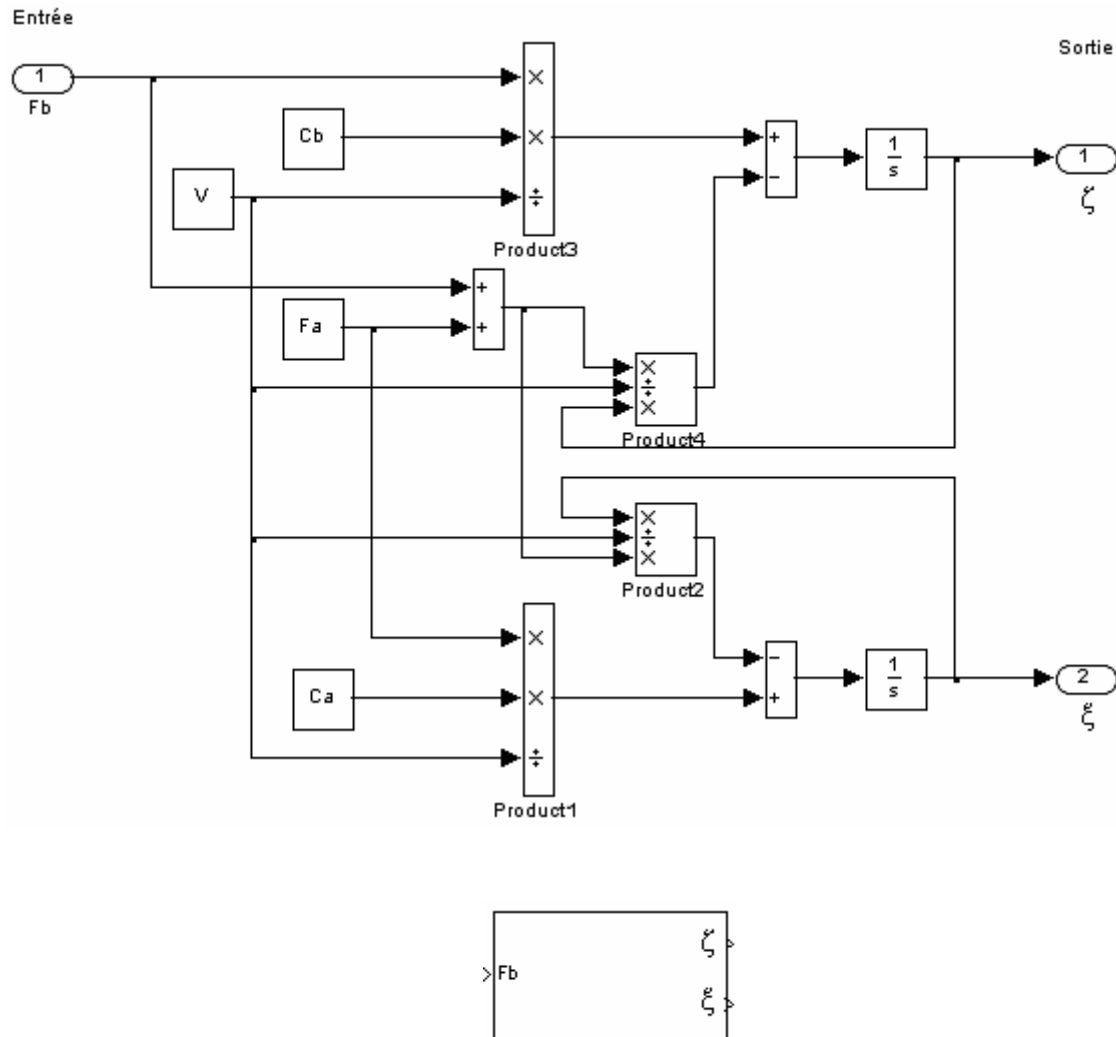


Figure 4.15 : Bloc simulink calculant ζ et ξ

4.3.5.2 Conception du bloc pour calcul de $[H^+]$:

C'est le bloc qui résout l'équation de troisième degré par rapport à $[H^+]$ (3.11).

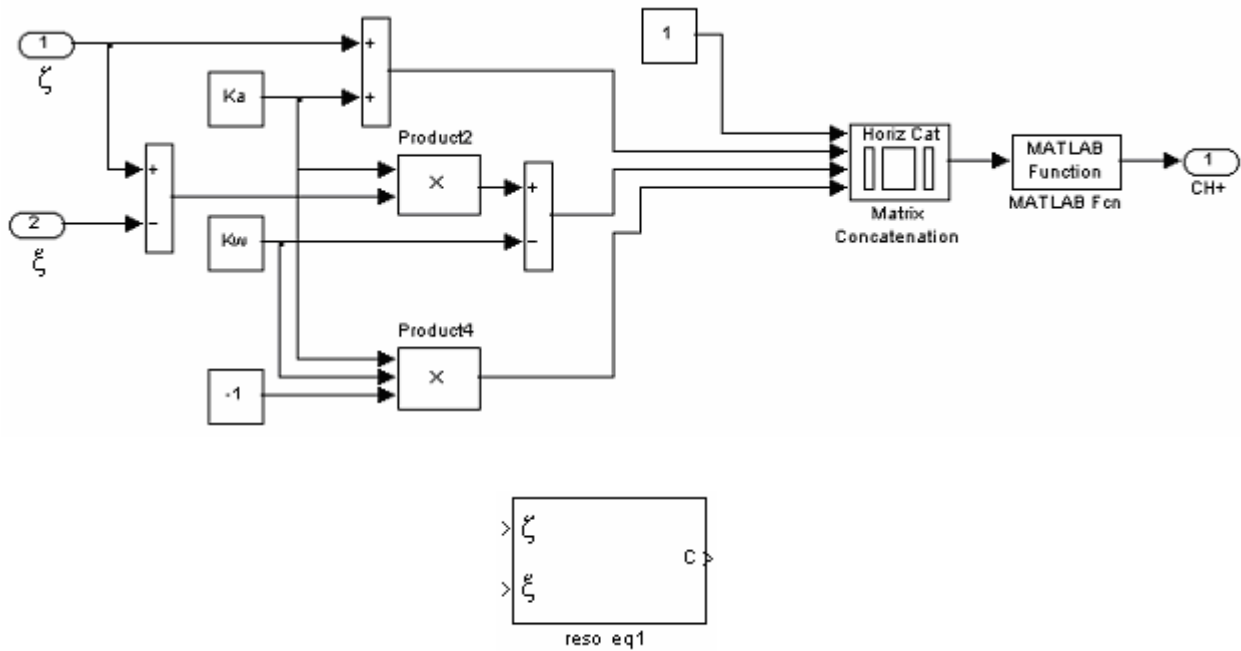


Figure 4.16 : Bloc simulink calculant $[H^+]$

4.3.5.3 Conception du bloc pour calcul de pH:

Ce bloc calcule le logarithme de la concentration de $[H^+]$:

$$pH = -\text{Log}_{10}([H^+])$$

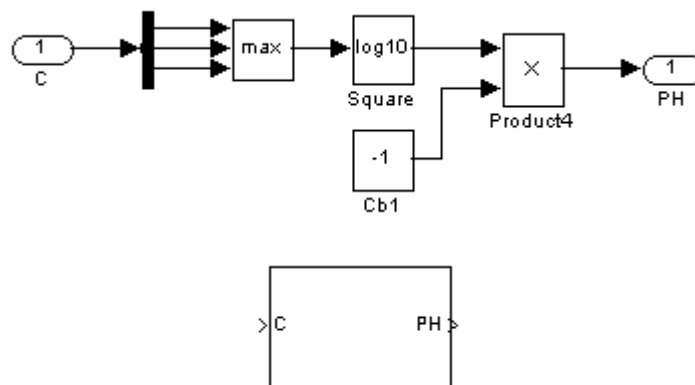


Figure 4.17 : Bloc simulink calculant pH

En combinant ces trois blocs on obtient le modèle du CSTR.

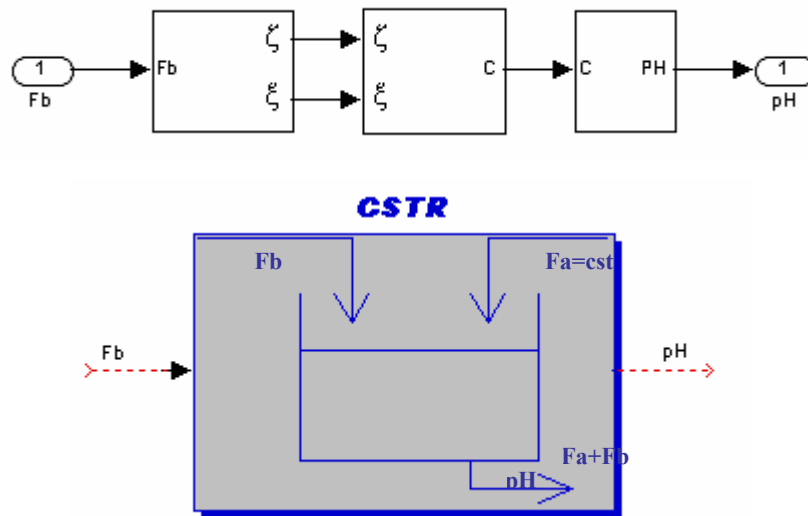


Figure 4.18 : Bloc simulink du CSTR

F_A est maintenu constant et un signal F_B de forme escalier a été injecté à l'entrée du modèle.

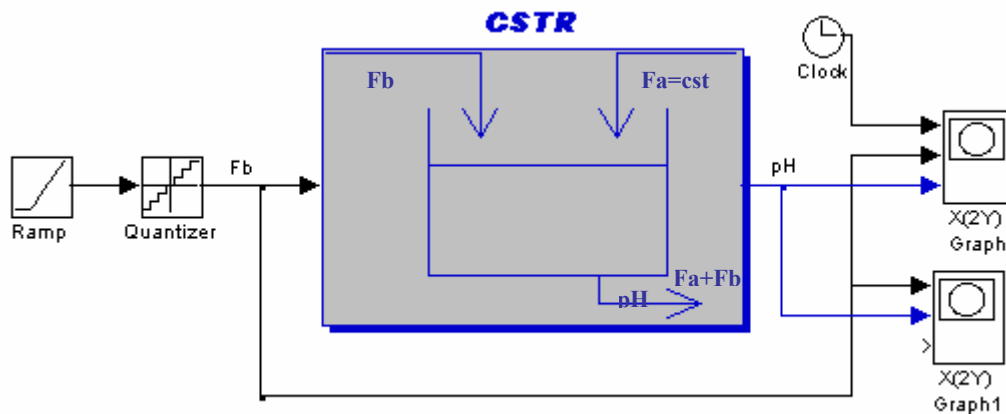


Figure 4.19 : Modèle CSTR

la courbe obtenue est la suivante :

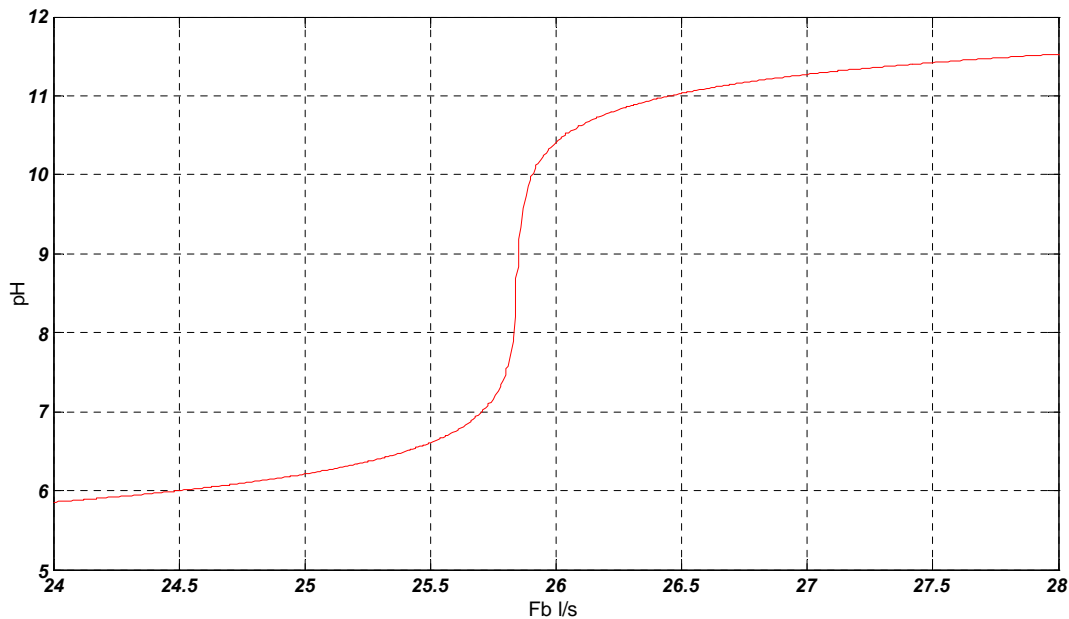


Figure 4.20 Courbe de titration du processus CSTR

4.4 Simulation du MPC :

Une série d'expériences de simulation a été exécutée pour établir des paramètres appropriés de MPC P2, C et λ . Le début de l'horizon de prévision, P1, était fixe à 2 intervalles d'échantillon qui était le retard du processus, et par conséquent, la longueur d'horizon de prévision, P2, a dû se prolonger au moins à 3 intervalles d'échantillon. Les expériences ont employé un modèle de MLP RNA dont la conception est décrite plus haut.

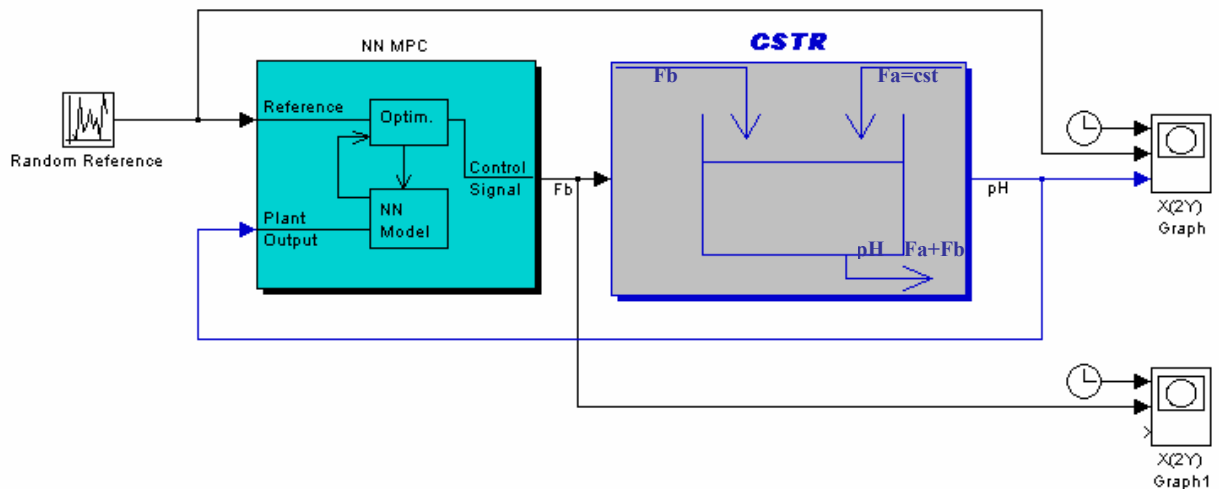


Figure 4.21 : Circuit de simulation

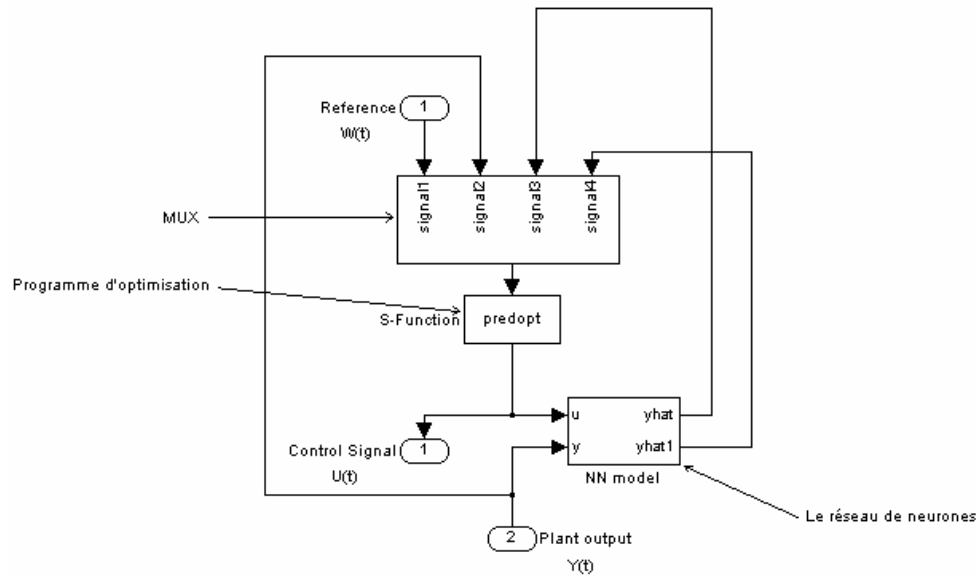


Figure 4.22 : Structure du NN MPC

4.4.1. L'horizon de commande:

Le schéma 4.22 montre l'effet d'augmenter l'horizon de commande pour un horizon de prévision de 5 intervalles d'échantillonnage et un facteur du poids λ nul. Les résultats montrent que l'augmentation de l'horizon de commande résulte une action de commande plus instable. La même expérience s'est appliquée à d'autres horizons de prévision et à coefficients de commande des poids différents de zéro. Cette tendance a été également notée pour de différents points de référence examinés. Par conséquent, l'utilisation des horizons de commande plus grands ont rapporté l'action de commande non stable.

L'objectif de commande pour la plupart des systèmes de pH, tels que la neutralisation d'eau usagée, est sujet du règlement du pH à la perturbation d'écoulement et de concentration, et par conséquent, la commande du processus pH a été étudiée comme problème de normalisation, et les contrôleurs ont été examinés en utilisant une perturbation constante d'alimentation en appliquant une série de changements de la référence aux différents points pour introduire une variation de 0.1 unité pH .

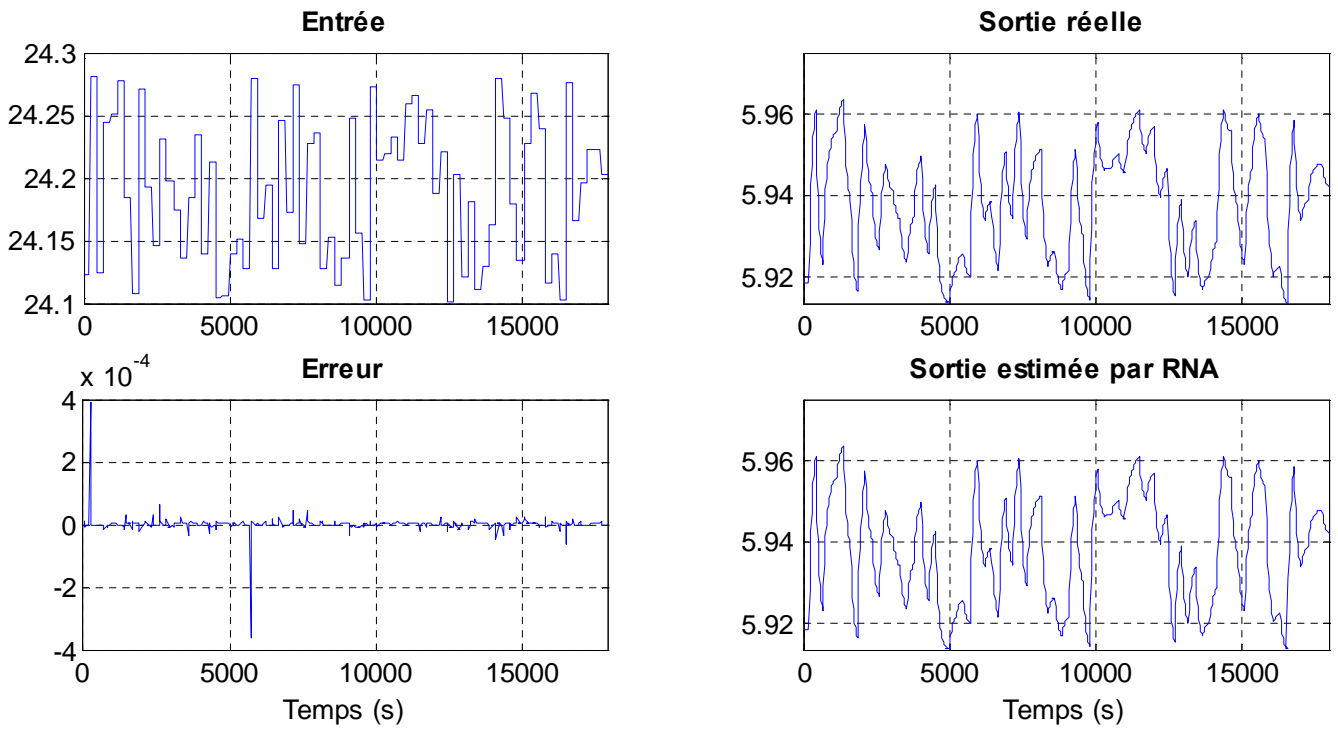


Figure 4.23 : Validation du réseau neuronal

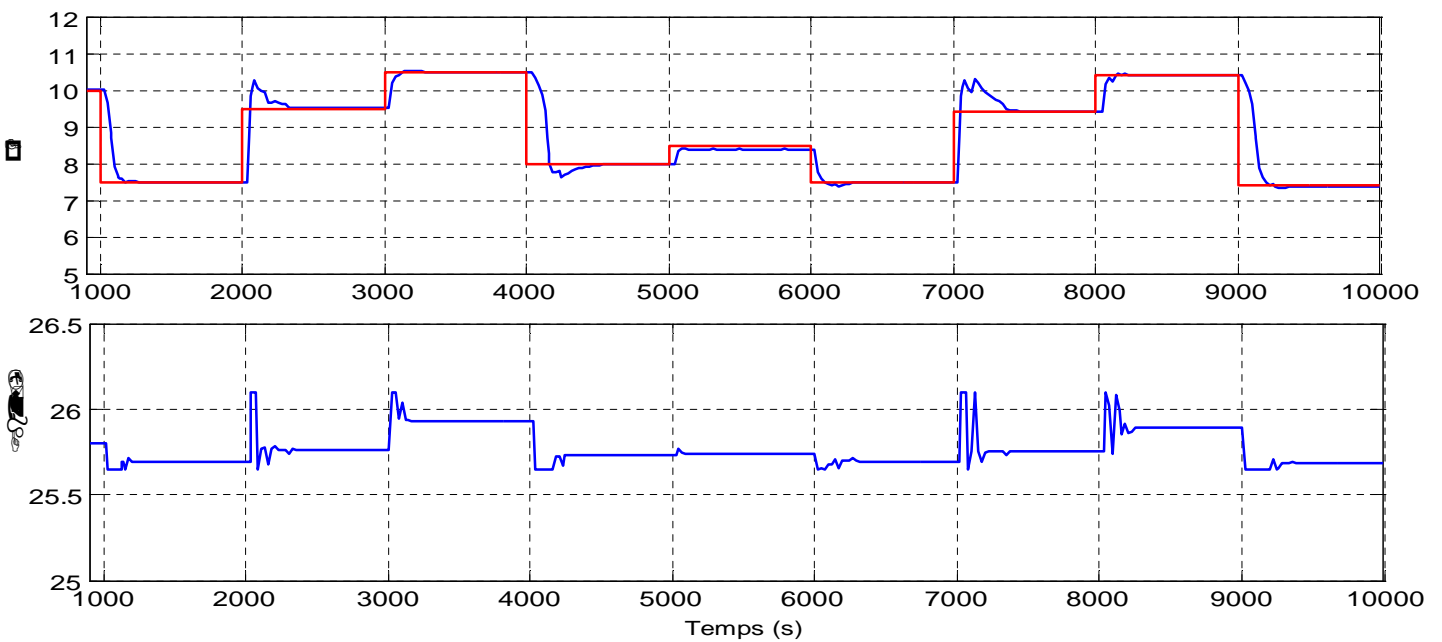


Figure 4.24 : Simulation point 7 à 10

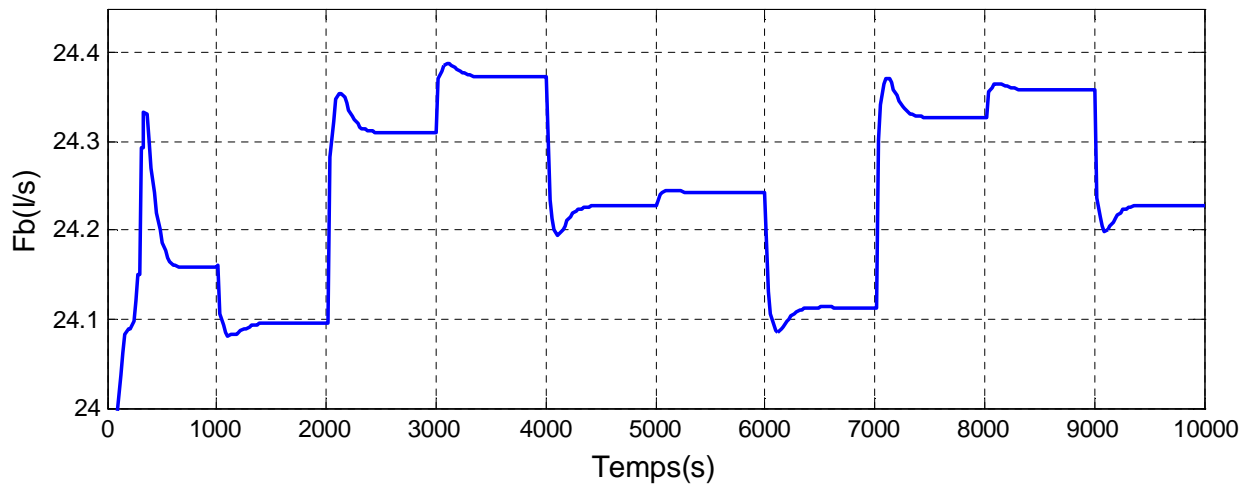
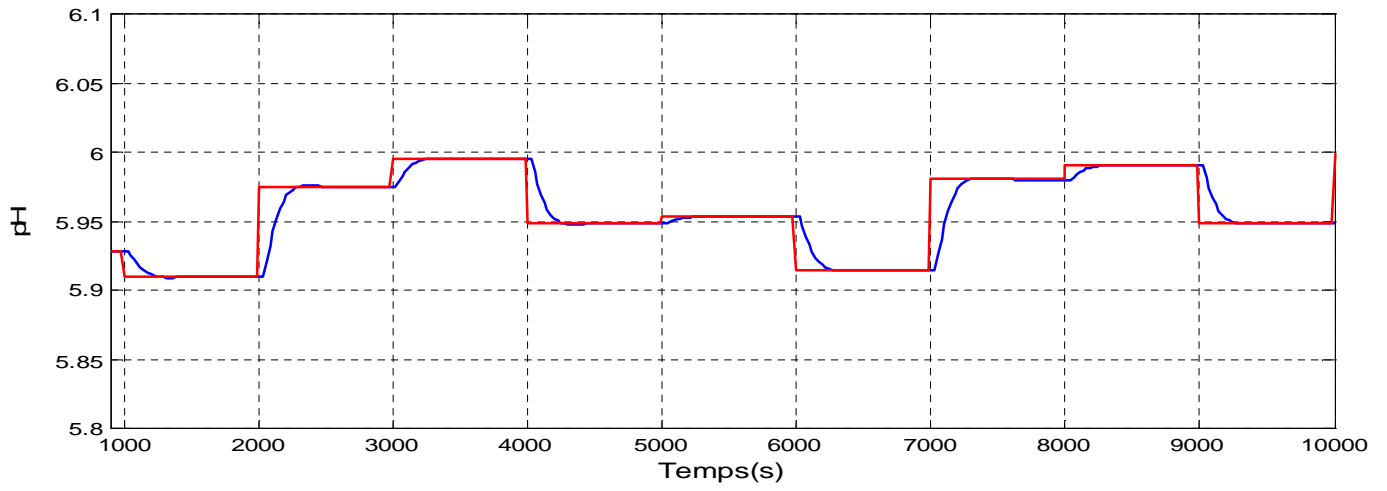


Figure 4.25 : Simulation point 5.95

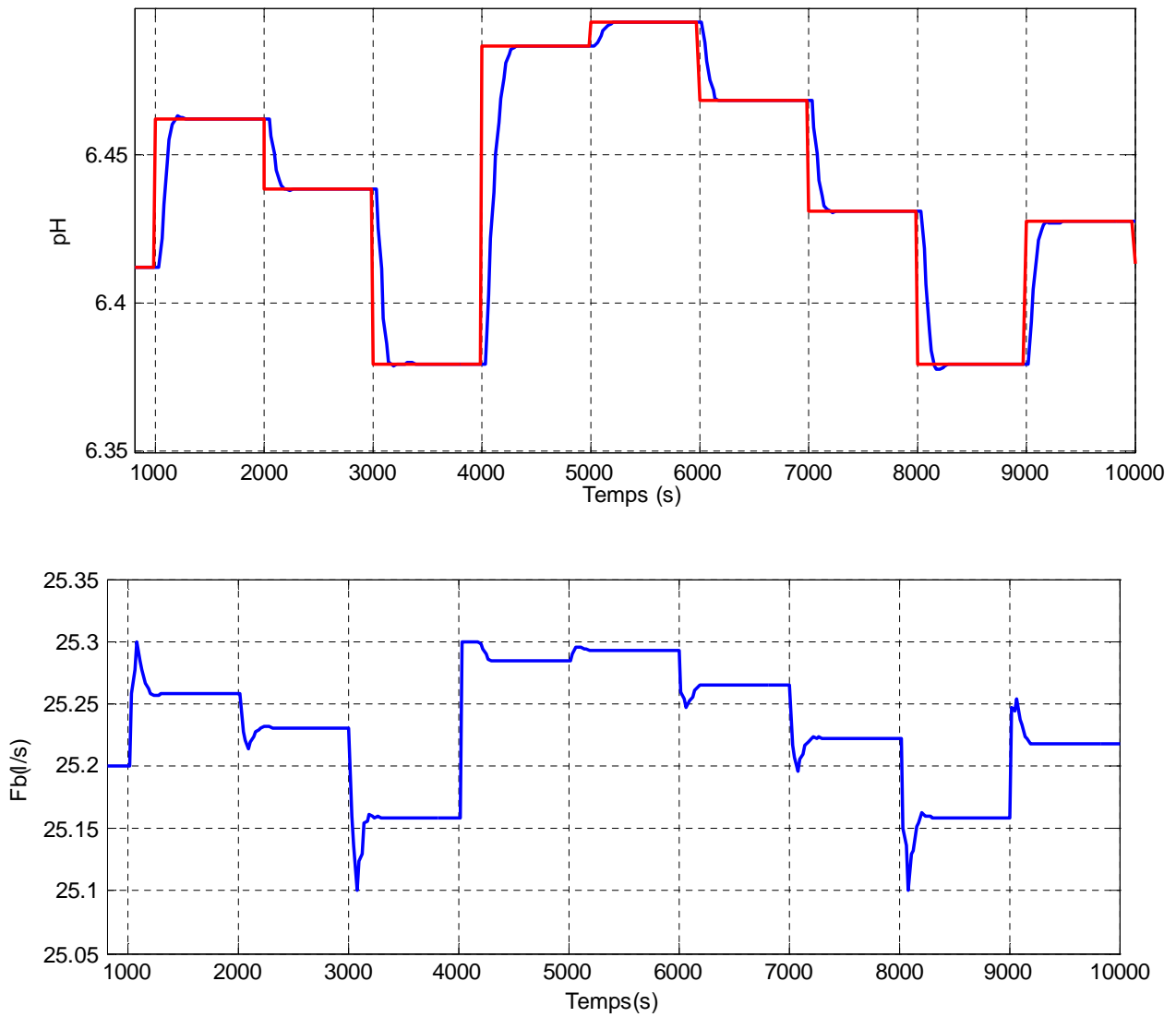


Figure 4.26 : Simulation point 6.4

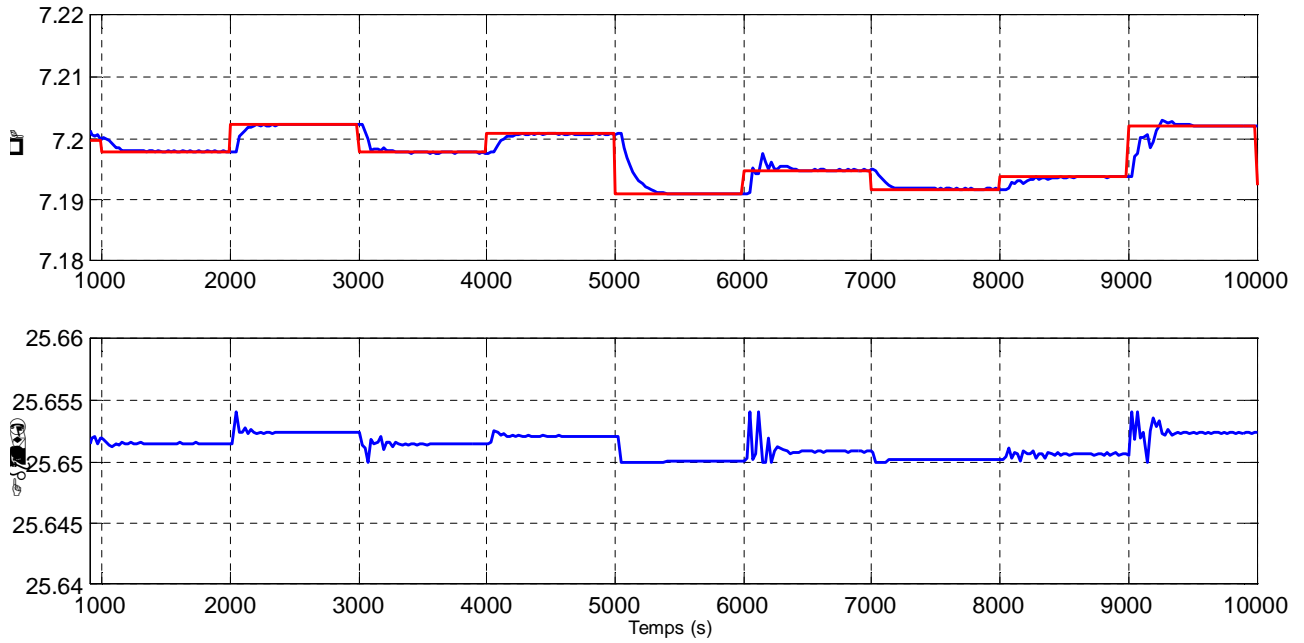


Figure 4.27 : Simulation point 7.2

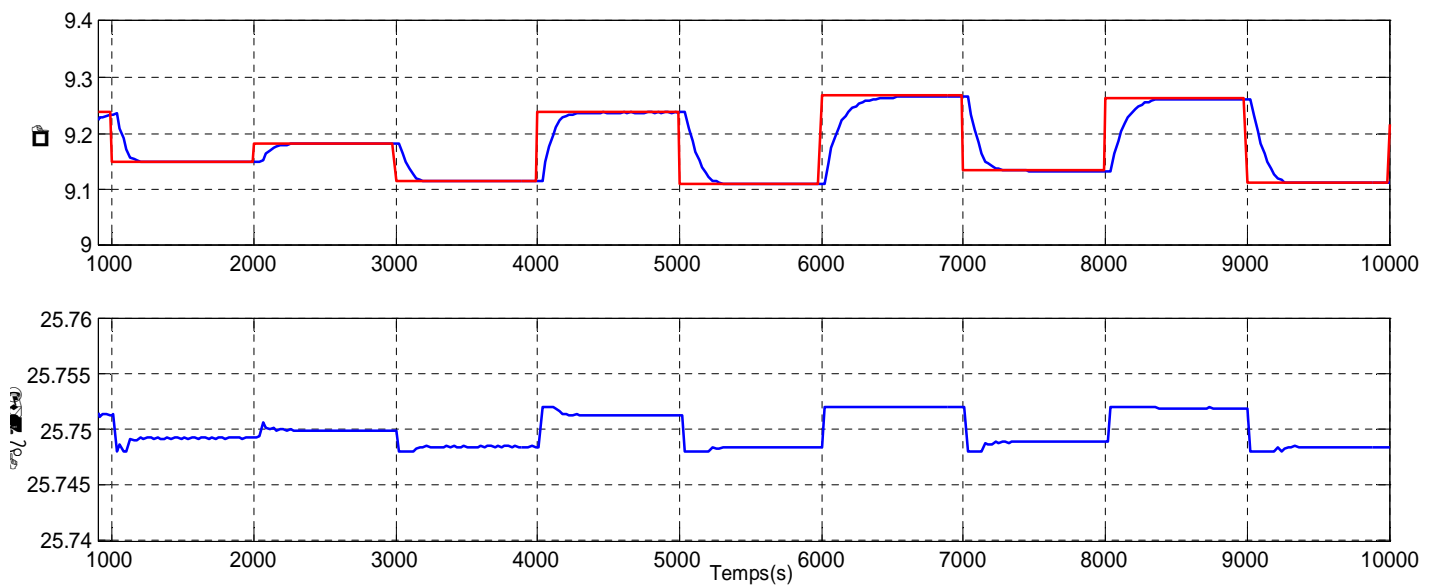


Figure 4.28 : Simulation point 9

4.5 Conclusion

Plusieurs stratégies de commande à modèles ont été décrites et comparées. Parmi les commandes candidats la commande MPC a été sélectionnée pour la commande en-ligne du processus pH. La commande MPC a été préférée car elle ne nécessite pas un modèle inverse et en raison de ses qualités pratiques. Le principe de fonctionnement de la commande MPC non-linéaire repose sur la commande LMPC. La principale différence pour la commande MPC non-linéaire est que le problème d'optimisation est non-linéaire et, pour résoudre ce problème, une procédure de recherche séquentielle employant le Nelder-Mead algorithme d'optimisation a été utilisé. Les principales propriétés de la commande MPC ont été discutées et l'implémentation de l'algorithme a été résumée.

Pour la commande MPC un horizon commande d'un intervalle d'échantillonnage a donné la réponse la plus stable de commande, et il n'y avait aucun avantage à employer un horizon de commande plus grand. Cela a également pour avantage la réduction du temps de calcul d'optimisation. L'horizon de prédiction devrait être d'au moins 3 intervalles d'échantillons. De plus, des horizons de 4 à 6 intervalles d'échantillon ont donné un bon compromis entre le rejet de perturbation et l'effort de la commande.

CONCLUSION

Dans le premier chapitre nous avons étudié et réalisé un banc d'essai de titration à base d'un microcontrôleur pour la titration, la mesure du pH et le calcul du point d'équivalence. L'originalité de cette réalisation réside dans l'implémentation de l'algorithme de la méthode du gradient et l'utilisation de la pompe micrométrique à effet de champ. Des mesures expérimentales ont été effectuées. Les résultats obtenus avec une erreur de 0.01% montre l'efficacité de ce banc d'essai indispensable dans tout laboratoire de chimie ou de biologie.

Dans le chapitre 3 une modélisation du pH d'un réacteur CSTR à l'aide de réseaux de neurones artificiels a été réalisée. Nous avons utilisé les techniques introduites dans le chapitre 2 pour le choix approprié de l'échantillon du temps et de l'ordre du modèle.

La modélisation par réseaux de neurones nous a permis de concevoir un signal d'excitation du processus et un échantillon de temps convenables. Ce qui a pour effet d'améliorer de manière significative la précision de prédiction. A partir des résultats obtenus en étudiant les modèles linéaires RNA de ce processus il est prouvé que ces derniers peuvent être appliqués avec succès aux systèmes non linéaires.

L'utilisation d'un modèle RNA pour la modélisation et la commande non-linéaire du pH a été étudiée. Un schéma de commande basé sur modèle a été développé en simulation en utilisant le modèle mathématique du processus.

Un examen des méthodes de commande du pH a confirmé que la commande non-linéaire basée sur modèle pourrait être bénéfique pour la commande du pH de certains procédés. La recherche a examiné le perceptron à multicouches. En utilisant les réseaux neuronaux à base du perceptron à multicouches, on doit prendre en considération, pour la modélisation dynamique non-linéaire, la sélection du signal d'excitation du processus, le temps d'échantillonnage et la structure du modèle.

. Il a été démontré que l'ordre des modèles linéaires locaux est directement en rapport avec l'ordre de non-linéarité du modèle. L'utilisation des indices de parcimonie linéaire est utile pour le choix de l'ordre du modèle. Il est important de tester un modèle de réseau de neurones sur des données qui sont différentes de celles utilisées à l'apprentissage. Une autre méthode de

validation des modèles basée sur des tests de corrélation a été décrite, et ces tests peuvent également fournir des informations sur l'adéquation de l'ordre du modèle de réseau.

Ces étapes de la conception du modèle de réseau de neurones ont une grande influence sur leur exactitude, comme on l'a démontré par l'étude de modélisation du pH d'un CSTR dans le chapitre 3. Ces techniques de conception ont été appliquées et ils ont donné une amélioration significative quand à l'exactitude de prédiction du modèle. La répartition des données de sortie du processus a été évaluée à l'aide d'un histogramme. Cela a permis une amélioration dans la conception du signal d'excitation du processus en donnant une plus grande distribution en sortie du processus, ce qui améliore la précision de prédiction du modèle de réseau de neurones dans la région à gain élevé.

Le processus pH incorpore plusieurs caractéristiques typiques à de nombreux procédés industriels, qui les rendent difficile à modéliser et à commander. Le processus est fortement non-linéaire. Un modèle mathématique du processus a été développé pour permettre la modélisation et le commande des processus à étudier dans la simulation.

Il existe plusieurs stratégies de commande à modèles qui pourraient être utilisées avec un modèle de réseau de neurones. La commande prédictive a été choisie parce qu'elle a des propriétés qui ne figurent pas dans les autres commandes. En particulier, la commande prédictive ne nécessite pas un modèle inverse et elle est naturellement adaptée aux contraintes. Un modèle de réseau de neurones à commande prédictive utilisant l'algorithme d'optimisation de Nelder-Mead a été élaboré.

La commande prédictive RNA a été comparée à la commande linéaire PID. Pour atteindre une meilleure performance du contrôleur linéaire il faut effectuer un réglage fastidieux qui consomme beaucoup de temps. Si l'on arrive à obtenir une commande satisfaisante au voisinage d'un point de consigne, la commande se détériore quand on s'en éloigne. En revanche, la commande prédictive RNA opère de manière satisfaisante sur un large éventail de points de fonctionnement. La performance de suivi du point de consigne du contrôleur RNA est nettement meilleure que celle du contrôleur linéaire.

Il existe plusieurs domaines où ces travaux pourraient être poursuivis. On pourrait étendre ce travail au problème de neutralisation des eaux usées, mais on le rendrait plus complexe pour la modélisation et la commande. Par conséquent, il semble raisonnable de ne pas poursuivre cette option jusqu'à ce que l'actuel système acide unique/base unique soit profondément étudié.

Bibliographie

- ASTROM, K.J. and WITTENMARK, B. (1989) Adaptive control. Addison Wesley.
- BHAT, N.V. and McAVOY, T.J. (1990) Use of neural nets for dynamic modelling and control of chemical process systems. Computers Chem. Engng., vol. 14, no. 4/5, pp. 573-583.
- BHAT, N.V. and McAVOY, T.J. (1992) Determining model structure for neural models by network stripping. Computers Chem. Engng., vol. 16, no. 4, pp. 271-281.
- BILLINGS, S.A. and LEONTARITIS, I.J. (1981) Identification of nonlinear systems using parameter estimation techniques. Proc. IEE Conf. On Control and its Applications, Warwick, UK, pp. 83-187.
- BISHOP, C.M. (1995) Neural networks for pattern recognition. Oxford University Press. FAUSETT, L. (1994) Fundamentals of neural networks: Architectures, algorithms and applications. Prentice-Hall.
- Bonnin Nathalie Programme DosA, version 3.3.1 Janvier 2004.
- BROWN, M. and HARRIS, C. (1994) Neurofuzzy adaptive modelling and control. Prentice-Hall.
- CHEN, S. and BILLINGS, S.A. (1992) Neural networks for nonlinear dynamic system modelling and identification. Int. J. Control, vol. 56, no. 2, pp. 319-346.
- COOK, P.A. (1986) Nonlinear Dynamical Systems. Prentice-Hall.
- CYBENKO, G. (1989) Continuous value neural networks with two hidden layers are sufficient. Math. Contr. Signal and Sys., vol. 2, pp. 303-314 .
- CYBENKO, G. (1989) Continuous value neural networks with two hidden layers are sufficient. Math. Contr. Signal and Sys., vol. 2, pp. 303-314 .
- DIDIER René R, Pierre GRECIAS Chimie Sup. PCSI cours et exercices d'application Edition LAVOISIER, Paris 1995.
- DOHERTY Sean Kevin Control of pH in Chemical Processes using Artificial Neural Networks, Liverpool 1999.
- DORF, R.C. (1992) Modern control systems. Addison Wesley.
- DRAEGER, A, ENGELL, S. and RANKE, H. (1995) Model predictive control using neural networks. IEEE Control Systems, vol. 5, no. 5, pp.61-66.

ECONOMOU, G. and MORARI, M. (1986) Internal model control. 5. Extension to nonlinear systems. *Ind. Eng. Chem. Process Des. Dev.*, vol. 25, pp.403-411.

ELMAN, J.L. (1990) Finding structure in time. *Cognitive Science*, vol. 14, pp. 179-211.

EVANS, J.T. (1994) Investigation of a multi-layer perceptron network to model and control a non-linear system. Ph.D. Thesis, Liverpool John Moores University, UK.

Falzone, R. Seeber, D. Tonelli, C. Ciavatta, C. Gessa² and D. Montecchio
Dependence on molecular weight of acid-base properties of humic and fulvic acids. *Analisis*, Vol 26, PP 214-219, 1998.

FOSTER, B., COLLOPE, F. and UNGAR, L.H. (1990) Forecasting using neural networks. *AIChE Annual Meeting 1990*, Chicago, USA.

FRANKLIN, G.F, POWELL, J. and WORKMAN, M.L. (1990) Digital control of dynamic systems. Addison Wesley.

FRIEDLAND, B. (1986) Control system design. McGraw-Hill.

FUNAHASHI, K. (1989) On the approximate realization of continuous mappings by neural networks. *Neural Networks*, vol. 2, pp. 183-192.

GARCIA, C.E. and MORARI, M. (1982) Internal model control - 1. A unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev.*, vol. 21, pp. 308-323.

GARCIA, C.E., PRETT, D.M. and MORARI, M. (1989) Model predictive control: Theory and practice - a survey. *Automatica*, vol. 25, pp. 335-348.

GIBSON, J.E. (1963) Nonlinear automatic control. McGraw-Hill.

GODFREY, K.R. (1980) Correlation methods. *Automatica*, vol. 16, pp. 527-534.

Gulaian M., Online adaptive pH control with multiple reagents *Decision and Control*, 1990., Proceedings of the 29th IEEE Conference on 5-7 Dec. 1990 Page(s):2806 - 2808 vol.5.

GUSTAFSSON, T.K. and WALLER, K.V. (1983) Dynamic modelling and reaction invariant control of pH. *Chem. Eng. Sci.*, vol. 38, pp. 389-398.

HENSON, M.A. (1998) Nonlinear model predictive control: current status and future directions. *Computers Chem. Engng.*, vol. 23, no. 2, pp. 187-202.

HOSKINS, J.C. and HIMMELBLAU, D.M. (1988) Artificial neural network models of knowledge representation in chemical engineering. *Computers Chem. Engng.*, vol. 12, pp. 881-890.

HUNT, K.J. and SBARBARO, D. (1991) Neural networks for Nonlinear Internal Model Control. IEE Proc. -D. vol. 138, no. 5, pp. 431-438.

HUNT, K.J., SBARBARO, R, ZBIKOWSKI, R and GAWTHORP, P.J. (1992) Neural networks for control systems - a survey. Automatica, vol. 28, no. 6, pp. 1083-1112.

ISERMANN, R., (1980) Practical aspects of process identification. Automatica, vol. 16, pp. 575-587.

ISERMANN, R., LACHMANN, K.-H. and MATKO, D. (1992) Adaptive control systems. Prentice-Hall.

JORDAN, M.I. and RUMELHART, D.E. (1991) Forward models: supervised learning with a distal teacher, Occasional Paper #40, Centre for Cognitive Science, MIT.

Jutila P. Pilot plant testing of an adaptive pH control algorithm based on physico chemical modelling. Mathematics and Computers in Simulation PP 523-533. 1984.

KESTELYN, J. (1990) Application Watch. AI Expert, June 1990, pp. 71.

LEONARITIS, I.J. and BILLINGS, S.A. (1985) Input-output parameter models for nonlinear systems. Int. J. Control, vol. 41, pp. 303-344.

LEONARITIS, I.J. and BILLINGS, S.A. (1987) Experiment design and identifiability for non-linear systems. Int. J. Sys. Sci., vol. 18, pp. 189-202.

LI, W and SLOTINE, J.-J.E. (1989) Neural network control of unknown nonlinear systems. Proc. Am. Cont. Conf., pp. 1136 -1141.

LJUNG, L. (1987) System identification - Theory for the user. Prentice-Hall.

MAH, R.S.H. and CHAKRAVARTHY, V. (1992) Pattern recognition using artificial neural networks. Computers Chem. Engng., vol. 16, no. 4, pp. 371-377.

McAVOY, T.J., HSU, E. and LOWENTHAL, S. (1972) Dynamics of pH in Controlled Stirred Tank Reactor. Ind. Eng. Chem. Process Des. Dev. Vol. 11, pp. 68-70.

McCULLOCH, W.S. and PITTS, W. (1943) A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133.

MILSANT Francis 110 études pratiques de l'amplificateur opérationnel. Edition EYROLLE, Paris 1981.

MOKADEM D., Etude des algorithmes adaptatifs pour la commande des processus. Mémoire de magistère Université de Sétif 1998.

NAHAS, E.P., HENSON, M.A. and SEBORG, D.E. (1992) Nonlinear internal model control strategy for neural network models. *Computers Chem. Engng.*, vol. 16, pp. 1039-1057.

NARENDRA, K.S. and PARTHASARATHY, K. (1990) Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks.* vol. 1, pp. 4-27.

OGATA, K. (1970) *Modern control systems.* McGraw-Hill.

PHILLIPS, C. L. and HARBOUR, R.D. (1988) *Feedback control systems.* Prentice-Hall.

POLLARD, J.F., BROUSSARD, M.R., GARRISON, D.B. and SAN, K.Y. (1992) Process identification using neural networks. *Computers Chem. Engng.*, vol. 16, pp. 253-270.

POTTMANN, M. and SEBORG, D.E. (1992a) Identification of non-linear processes using reciprocal multiquadratic functions. *J. Process Control*, vol. 2, no. 4, pp. 189-203.

POTTMANN, M. and SEBORG, D.E. (1997) A nonlinear predictive control strategy based on radial basis function models. *Computers Chem. Engng.*, vol. 21, pp. 965-980.

PROLL, T. and KARIM, M.N. (1994) Model-predictive pH control using real-time NARX approach. *AIChE Journal.* Vol. 40, No. 2, pp. 269-282.

PSALTIS, D, SIDERIS, A. and YAMAMURA, A. (1988) A multilayered neural network controller. *IEE Control Systems Magazine*, vol. 8, pp. 17-21.

RAWLINGS, J.B., MEADOWS, E.S. and MUSKE, K.R. (1994) Nonlinear model predictive control: a tutorial and survey. *Proc. Advanced Control of Chemical Processes*, pp. 185-197.

RICHALET, J., RAULT, A, TESTUD, J.L and PAPON, J. (1978) Model predictive heuristic control: application to industrial processes. *Automatica*, vol. 14, no. 5, pp. 413-428.

Rudan-Tasic D., C. Klofutar Gran's plots for end point detection in potentiometric titration of wine. *Analisis*, , Vol 26, PP 25-28, 1998.

SEBORG, D.E., EDGAR, T.F. and SHAH, S.L. (1986) Adaptive control strategies for process control : a survey. *AIChE J.*, vol. 32, no. 6, pp. 881-913.

SELMAN, B. (1989) Connectionist systems for natural language understanding. *Artif. Intell. Rev.*, vol. 3, pp. 23-31.

SIMPSON, P.K. (1990) *Artificial neural systems.* Pergamon Press.

Tavernier Christian Les microcontrôleurs PIC description et mise en œuvre. Edition DUNOD, Paris 2000.

VANDOREN, V.J. (1997) Multivariable controllers enter the mainstream. Control Engineering, vol. 44, no. 4, pp. 107-112.

WELLSTEAD, P.E. (1979) Introduction to physical system modelling. Academic Press.

WELLSTEAD, P.E. and ZARROP, M.B. (1991) Self tuning systems. Wiley.

WILLIS, M.J., DI MASSIMO, C., MONTAGUE, G.A., THAM, M.T. and MORRIS, A.J. (1991) Artificial neural networks in process engineering. IEE Proc.-D, vol. 138, no. 3, pp. 256-266.

ZIEGLER, J.G. and NICHOLS, N.B. (1942) Optimum settings for automatic controllers. Trans. ASME, vol. 64, no. 11, pp. 759-768.