

**République Algérienne Démocratique et  
Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique**  
Université Djilali Bounaama Khemis Miliana



**Faculté des Sciences et de la Technologie**  
**Département de la technologie**

Mémoire du Projet de Fin d'Etude  
Pour l'obtention du diplôme

**Master**

En Génie Electrique

Option : Automatique des systèmes de production

*Thème :*

*Identification et régulation d'un système hydraulique à base d'un  
automate programmable industriel*

Réalisé par :

 *GHELEM Nourddine*

 *DZIRI Ramadhan*

Encadré par : Dr. *HOCINE Abdelfettah*

Année Universitaire 2015/2016

## *DEDICACES*

*Je dédie ce travail à :*

*Mes parents*

*A toute ma famille*

*A tous mes amis proches*

*A mon binôme pour tout le travail qu'on a accompli*

*A toute la promotion 2016 et en particulier la  
promotion d'AUTOMATIQUE des systèmes automatisés*

*ASP*

*A tous ceux qui me sont chers*

*Ghelem Nourddine*

# *Dédicace*

*A celui à qui je dois tout;*

*A ma mère, mon père;*

*A tous mes frères et sœurs;*

*A toute ma famille;*

*A tous mes amis;*

*Qu'Allah nous aide tous à faire que le bien.*

**« Ramadhan dziri »**

## **REMERCIEMENTS**

*Nous remercions ALLAH le tout puissant de nous donner le courage et la volonté de mener à terme ce présent travail.*

*Nous tenons à exprimer nos vifs remerciements à notre promoteur **Dr. HOCINE Abdelfettah** pour nous avoir encadrés durant notre projet de fin d'études et nous conseillé tout le long de notre travail.*

*Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre projet.*

*Nous souhaitons aussi remercier **Mr. MIHABIL Yassin** le directeur du centre de formation professionnel à Ain Defla pour nous accepter à faire nos études au sein du centre et nous remercions aussi **Mlle. TAMAR Khadija** Enseignante à ce centre pour son aide précieux, et de son confiance.*

*Nous tenons à exprimer notre reconnaissance à tous les enseignants de la faculté de technologie plus particulièrement le département d'électronique qui nous ont attribués.*

*Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.*

## **Résumé**

L'automate programmable industriel est l'un des outils les plus utilisés pour la commande et la gestion des systèmes automatisés, il est fréquemment utilisé au sein des stations de traitement et de pompage de l'eau ,dans ce mémoire nous avons fait notre étude sur un banc d'essais géré par un automate programmable Schneider, ce banc simule une station de pompage réelle, notre objective est de fait une régulation de niveau d'eau dans deux réservoirs placées dans le banc , d'abord nous avons expliqué les différents composants de ce banc, ensuite nous avons fait une supervision de ce systèmes à l'aide d'une écran Magelis, puis nous avons abordé l'identification de ce système pour ceci nous avons utilisé deux méthodes d'identification récursif, puis nous avons fait une simulation sous Matlab de ces méthodes à la suite nous avons implémenté l'algorithme de ces méthode sous l'automate afin d'identifier le système en temps réel, pour la régulation du système nous avons utilisé les modèles mathématiques obtenus de l'identification sous une simulation Matlab sachant que nous avons simulé la régulation du système avec deux régulateurs numériques , finalement nous avons implémenté l'algorithme de régulateur PID dans l'automate ou nous avons réglé le niveau d'eau dans réservoirs placées dans le banc.

## **Mots clés**

Identification, régulateur PID, supervision, automate programmable, simulation

## **Abstract**

the programmable logic controller is one of the most used for the control and management of automated tools, it is frequently used in the treatment and water pumping stations in this memory we did our survey on a test bench run by a Schneider PLC, this bench simulates an actual pumping station, our objective is to take a controlling water level in two tanks placed in the bench, first we explained the different components of this bench, then we made a supervision of this system using a Magelis screen, the next step we have addressed the identification of this system for this we used two methods of identification recursive, then we did a simulation in Matlab of these methods after we implemented the algorithm of the method in the controller to identify the real-time system for the regulation of the system we used mathematical models obtained identification in a Matlab simulation knowing that we simulated control system with digital controllers, eventually we implemented the PID algorithm in the PLC or we set the water level in reservoirs placed in the bench.

## **Keywords**

Identification, PID controller, supervision, programmable controller simulation

## *Table des matières*

### **CHAPITRE I**

I.1. Introduction .....	5
I.2. Les systèmes automatisés .....	5
I.2.1. Définitions.....	5
I.2.2. Structure D'un Système Automatisé de Production (SAP) .....	5
I.2.2.1. La Partie Opérative (PO).....	6
I.2.2.2. La Partie Relation (PR) .....	8
I.2.2.3. La Partie Commande (PC) .....	8
I.3. Présentation générale du système .....	9
I.3.1. Partie opérative .....	9
I.3.1.1. Les actionneurs.....	9
I.3.1.2. Les capteurs.....	11
I.3.2. Partie commande.....	12
I.3.2.1. L'automate programmable industriel .....	13
I.3.2.2. L'automate Schneider Premium.....	15
I.3.2.2. Le variateur de vitesse.....	20
I.3.2.2.1. Le variateur Altivar 31 .....	21
I.3.2.3. Terminal Magelis à écran graphique.....	22
I.3.2.3.1. Terminal graphique à clavier.....	23
I.4. Présentation du fonctionnement du système .....	24
I.4.1. Plane hydraulique du système.....	26
I.5. Les logiciels utilisés.....	27
I.5.1. Le logiciel PL7.....	28
I.5.1.1. L'éditeur de configuration.....	28
I.5.1.2. La programmation de l'application .....	29
I.5.2. Le logiciel XBT L1000 .....	29
I.5.2.1. L'éditeur d'application.....	29
I.5.2.2. Objet touche fonction .....	30
I.6. Supervision du système .....	31
I.6.1. Les écrans de supervision .....	32
I.7. Conclusion .....	34

## **CHAPITRE II**

II.1. Introduction .....	37
II.2. Définition de l'identification .....	37
II.3. Choix du signal d'excitation .....	38
II.4. Les méthodes générales d'identification .....	40
II.4.1. Méthodes non paramétriques .....	40
II.4.2. Méthodes paramétriques .....	42
II.4.2.1. Les méthodes paramétriques graphiques (déterministes) .....	42
II.4.2.2. Les méthodes paramétriques statistiques .....	45
II.5. Les algorithmes de l'identification .....	46
II.5.1. Algorithme du moindre carré simple .....	46
II.5.2. Méthode de la Matrice Instrumentale .....	48
II.5.3. Méthode de moindre carré récursive .....	48
II.5.4. Méthode de moindre carré étendu récursive [11] .....	50
II.6. Validation du modèle .....	50
II.7. Quelques structures « procédé + perturbation » largement utilisées en identification .....	51
II.8. Estimation de l'ordre du système .....	53
II.9. Simulation MATLAB des algorithmes d'identification récursif .....	53
II.10. Implémentation des algorithmes d'identification sous l'automate programmable Schneider premium .....	56
II.10.1. L'identification paramétrique du système .....	56
II.10.2. Le signal d'excitation .....	57
II.10.3. La validation du modèle .....	57
II.11. Conclusion .....	59



## **CHAPITRE III**

III.1. Introduction.....	61
III.2. Notions de système .....	61
III.3. Classifications des systèmes .....	61
III.3.1. Classifications selon la nature de système .....	61
III.3.2. Classifications selon les entrées et les sorties .....	62
III.4. La commande des systèmes .....	62
III.5. La régulation des systèmes .....	62
III.6. La régulation numérique des systèmes .....	64
III.6.1. Synthèse des régulateurs numériques.....	64
III.6.1.1. Méthode directe de Synthèse des régulateurs numériques .....	64
III.6.1.2. Méthode par transposition des régulateurs continus.....	66
III.6.2. Régulateur PID analogique .....	66
III.6.3. Les régulateurs PID discrets.....	67
III.6.3.1. Généralités .....	67
III.6.3.2. LES DIFFERENTES ACTIONS.....	67
III.6.3.3. Le régulateur Proportionnel-Intégral P.I .....	71
III.6.3.4. Le régulateur universel P.I.D.....	72
III.6.3.5. Calcule des paramètres de régulateur PID.....	74
III.6.3.5.1. Approche par les méthodes expérimentales.....	74
III.6.3.5.2. Approche de Ziegler-Nichols pour les systèmes en boucle fermée (déjà régulés).....	75
III.6.3.5.3. Approche de Ziegler-Nichols pour les systèmes en boucle ouverte (pas encore régulés) .....	76
III.7. Simulation MATLAB des régulateurs utilisé pour le système .....	77
III.7.1. Régulation de système par un régulateur numérique $R(z)$ .....	78
III.7.2. Régulation PID du système .....	80
III.8. Implémentation du régulateur PID sous l'automate programmable .....	81
III.8.1. Réponse indiciel du système réel .....	81
III.8.2. Réponse du système real avec une consigne aléatoire .....	83
III.9. Conclusion .....	84

## *Notion et Abréviations*

SAP	Système Automatisé de Production
PO	Partie Opérative
PR	Partie Relation
PC	Partie Commande
TOR	Tout ou Rien
HMI	Human Machine Interface
API	Automates Programmables Industriels
SBPA	Séquences Binaires Pseudo Aléatoires
MCR	Moindre Carré Récursif
MCE	Moindre Carré Etendu
ARMAX	Auto Régressive à Moyenne Ajustée et variable Exogène
ARX	Auto Régressive à variable exogène
DFB	Derived Function Bloc
LD	Ladder Diagram
GRAFCET	GRAphe Fonctionnel de Commande par Etapes et Transitions
ST	Structured Text
IL	Instruction List
PID	Proportionnel Intégral Dérivée
PLC	Programmable Logic Controller
CPU	Central Processing Unit
EEPROM	Electrical Erasable Programmable Read Only Memory
RAM	Random Access Memory
VM	électrovanne proportionnel
EV	électrovanne TOR
$y(t)$	Sortie réel de système
$\hat{y}(t)$	Sortie estime de système
$\theta$	Matrice des paramètres réels
$\hat{\theta}$	Matrice des paramètres estime
$\emptyset$	Vecteur des entres et de sorties
Z	Matrice instrumentale
$P_N$	Matrice de gain
$RN(i)$	Résulta de teste de blancheur

## *Liste des figures*

### **CHAPITRE I**

<b>Figure.I.1.</b> Structure D'un Système Automatisé.....	6
<b>Figure.I.2.</b> Les effecteurs .....	6
<b>Figure.I.3.</b> Les actionneurs .....	7
<b>Figure.I.4.</b> Les prés-actionneurs .....	7
<b>Figure.I.5.</b> Les capteurs.....	8
<b>Figure.I.6.</b> Salle de contrôle.....	8
<b>Figure I.7.</b> Vue réelle de L'électropompe.....	9
<b>Figure I.8.</b> Plaque signalétique de moteur .....	9
<b>Figure I.9.</b> Vue réelle de L'électrovanne proportionnelle. ....	10
<b>Figure I.10.</b> L'électrovanne TOR .....	10
<b>Figure I.11.</b> Vue réelle de La pompe de perturbation.....	11
<b>Figure I.12.</b> Capteur de pression.....	11
<b>Figure I.13.</b> Débitmètre.....	12
<b>Figure I.14.</b> Contrôleur de niveau.....	12
<b>Figure.I.15.</b> Vue réelle de l'armoire de commande du système .....	13
<b>Figure.I.16.</b> Vue réelle de Magelis XBT- F024610.....	13
<b>Figure I.17.</b> Vue réelle de L'automate Schneider Premium TSX 57253.....	15
<b>Figure I.18.</b> TSX PSY 2600M .....	16
<b>Figure I.19.</b> Le processeur TSX P57253.....	17
<b>Figure I.20.</b> Module entrée TOR TSX DEY08D2.....	18
<b>Figure I.21.</b> Module sortie TOR TSX DSY 08R5 .....	18
<b>Figure I.22.</b> Module entrées analogiques TSX AEY414.....	19
<b>Figure.I.23.</b> Module sorties analogique TSX ASY 800.....	20
<b>Figure.I.24.</b> Vue réelle du variateur Altivar 31.....	21
<b>Figure.I.25.</b> Le terminal Magelis à écran graphique XBT F024610.....	23
<b>Figure.I.26.</b> Vue réelle de l'unité régulation de débit et niveau d'eau. ....	25
<b>Figure.I.27.</b> La visualisation directe sur le banc.....	25
<b>Figure.I.28.</b> Le schéma hydraulique du banc d'essais .....	26

## *Liste des figures*

---

<b>Figure.I.29.</b> Editeur de configuration matérielle.....	28
<b>Figure.I.30.</b> L'éditeur d'application.....	30
<b>Figure.I.31.</b> Organigramme d'écrans de la supervision.....	31
<b>Figure.I.32.</b> L'écran 01 .....	32
<b>Figure.I.33.</b> L'écran 02 .....	33
<b>Figure.I.34.</b> L'écran 03 .....	33
<b>Figure.I.35.</b> L'écran 04 .....	34

## **CHAPITRE II**

<b>Figure.II.1.</b> Structure de l'identification des systèmes .....	37
<b>Figure.II.2.</b> Registre générateur de séquence SBPA .....	39
<b>Figure.II.3.</b> Allure d'une séquence binaire pseudo aléatoire à niveaux symétriques.....	39
<b>Figure.II.4.</b> Les étapes de l'identification.....	40
<b>Figure.II.5.</b> Structure d'identification par réponse impulsionnel .....	41
<b>Figure.II.6.</b> Identification par méthode de strejc .....	43
<b>Figure.II.7.</b> Méthode d'identification de Broida des modèles continues.....	44
<b>Figure.II.8.</b> Méthode d'identification de Broida des modèles échantillonnés.....	44
<b>Figure.II.9.</b> Structure d'identification paramétrique statique .....	45
<b>Figure.II.10.</b> Structure de représentation de modèle .....	46
<b>Figure.II.11.</b> Structure de modèle ARX .....	51
<b>Figure.II.12.</b> Structure de modèle Erreur de sortie .....	52
<b>Figure.II.13.</b> Structure de modèle ARMAX .....	52
<b>Figure.II.14.</b> Structure de modèle ARARX.....	52
<b>Figure.II.15.</b> Estimation de l'ordre du système avec la méthode de strejc.....	53
<b>Figure.II.16.</b> Le signal d'excitation SBPA .....	54
<b>Figure.II.17.</b> Résultat de l'identification de système avec MATLAB.....	55
<b>Figure.II.18.</b> Les blocs DFB des deux algorithmes d'identification .....	56
<b>Figure.II.19.</b> Bloc DFB de signal SBPA .....	57
<b>Figure.II.20.</b> Block DFB de validation du modèle .....	57
<b>Figure.II.21.</b> Résultats de l'identification de deux algorithmes avec l'automate.....	58

## CHAPITRE III

<b>Figure.III.1.</b> Structure de la commande des systèmes.....	63
<b>Figure.III.2.</b> Structure d'une régulation numérique .....	64
<b>Figure.III.3.</b> Structure de régulation en boucle fermé .....	65
<b>Figure.III.4.</b> Réponse indiciel d'un régulateur P .....	68
<b>Figure.III.5.</b> Réponse indiciel d'un régulateur I.....	68
<b>Figure.III.6.</b> Diagramme structurel de régulateur I .....	69
<b>Figure.III.7.</b> Réponse indiciel d'un régulateur D .....	70
<b>Figure.III.8.</b> Diagramme structurel de régulateur D.....	70
<b>Figure.III.9.</b> Diagramme structurel de régulateur PI.....	71
<b>Figure.III.10.</b> Diagramme structurel de régulateur PID .....	72
<b>Figure.III.11.</b> Diagramme structurel de régulateur PID .....	73
<b>Figure.III.12.</b> Approche de Ziegler-Nichols pour les systèmes en boucle fermée.....	75
<b>Figure.III.13.</b> Approche de Ziegler-Nichols pour les systèmes en boucle ouverte.....	76
<b>Figure.III.14.</b> Calcul de paramètres PID par la méthode de Ziegler-Nichols.....	77
<b>Figure.III.15.</b> Réponse on boucle fermé de $F(z)$ .....	78
<b>Figure.III.16.</b> Réponse indiciel du système avec le régulateur $R(z)$ .....	79
<b>Figure.III.17.</b> Structure de régulation PID du système.....	80
<b>Figure.III.18.</b> Réponse indiciel de système avec le régulateur PID .....	80
<b>Figure.III.19.</b> Structure de régulation de système en boucle fermé .....	81
<b>Figure.III.20.</b> Bloc DFB de régulateur PID.....	81
<b>Figure.III.21.</b> Réponse indiciel de système sont perturbations .....	82
<b>Figure.III.22.</b> Réponse indiciel de système avec perturbations.....	82
<b>Figure.III.23.</b> Bloc DFB de programme de consigne .....	83
<b>Figure.III.24.</b> Réponse de système avec programme de consigne .....	83

*Liste des tableaux*

**CHAPITRE I**

**Tab.I.1.** Types du cycle.....5

**CHAPITRE II**

**Tab.II.1.** Ordre de système par la méthode de strej .....43

**Tab.II.2.** Résultats du test de blancheur avec MATLAB.....56

**Tab.II.3.** Résultats du test de blancheur avec automate.....59

# *Introduction Générale*

## ***Introduction Générale***

---

La compétitivité des entreprises impose un recours à la fois fréquent et intensif à des technologies de production avancées. La productique et la complexité des opérations à exécuter, conduisent à la mise en œuvre de dispositifs et systèmes pour l'automatisation des ateliers de fabrication ou de production.

L'automate programmable industriel A.P.I est aujourd'hui le constituant le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie car il répond à des besoins d'adaptation et de flexibilité pour un grand nombre d'opérations. Cette émergence est due en grande partie, à la puissance de son environnement de développement et aux larges possibilités d'interconnexions.

Le domaine de l'hydraulique parmi d'autres, est témoin de cette révolution, et de nombreuses sociétés algériennes, orientées dans le stockage, l'épuration et la distribution de l'eau, cherchent à se procurer cette solution d'automatisme au niveau de leurs stations de pompes et de traitements.

Notre travail consiste à faire l'identification et la régulation du débit d'un système hydraulique à base d'un automate programmable industrielle Schneider.

Le travail à été fait sur un banc d'essais qui simule une station de pompage d'eau, ce banc est installé dans le centre de formation professionnelle Ain Defla 02.

Ce mémoire est représenté en trois chapitres :

Dans le premier chapitre nous donnons une généralité sur les systèmes automatisés, nous expliquons les différentes parties de ce type de système (partie opérative, partie commande .....), puis nous présentons le banc d'essais sur lequel on va travailler, nous abordons la partie opérative de ce système (actionneurs, capteurs) puis la partie commande (automate programmable, variateur de vitesse ...). A la fin de ce chapitre nous avons fait la supervision de système hydraulique à travers un écran Magelis à clavier, pour cet effet nous développons quatre écrans de supervision.

Dans la deuxième chapitre nous allons aborder l'identification du système hydraulique étudié, pour cela nous utilisons la méthode de moindre carré récursive MCR, et à cause du biais de l'estimation des paramètres du modèle avec cette méthode on va utiliser une deuxième méthode d'identification c'est la méthode de moindre carré étendu MCE, le biais de l'estimation est bien éliminé avec cette



## *Introduction Générale*

---

méthode. Premièrement nous avons fait une simulation MATLAB des deux algorithmes d'identification, puis nous implémentons ces algorithmes sous l'automate pour identifier en temps réelle le système. A la fin de ce chapitre nous traçons les graphes des résultats d'identification, puis nous donnons les modèles mathématiques du système obtenus par les deux méthodes.

Notre objective dans le troisième chapitre est de faire la régulation de niveau d'eau dans les deux cuves placées dans le banc d'essai, pour cela nous allons faire une simulation MATLAB des deux régulateur numérique, nous utilisons les modèles mathématiques obtenus dans la 2<sup>ème</sup> chapitre dans cette simulation. Pour la régulation du système réel nous implémentons l'algorithme du régulateur PID numérique sous l'automate programmable, puis nous calculons les paramètres de ce régulateur avec la méthode expérimentale de **Ziegler-Nichols**. A la fin de ce chapitre nous traçons les graphes de réponse de système avec deux types de consigne.

# **CHAPITRE I**

## **Banc de régulation de débit et de niveau d'eau**

## I.1. Introduction

Notre étude est faite sur un banc d'essai (system hydraulique), qui simule une station de pompage real. Ce banc d'essai représente fidèlement le fonctionnement des systèmes hydrauliques réelle ; mais a une échèle moins importante

Dans ce chapitre on présente ce système, nous avons expliqué les différents composants, au point de vu hardware et software, à la fin de ce chapitre nous montrons les différentes possibilités pour commander le système.

## I.2. Les systèmes automatisés

### I.2.1. Définitions

Un système automatisé ou automatique est un système réalisant des opérations et pour lequel l'homme n'intervient que dans la programmation du système et dans son réglage.

Les buts d'un système automatisé sont de réaliser des tâches complexes ou dangereuses pour l'homme, effectuer des tâches pénibles ou répétitives ou encore gagner en efficacité et en précision. [1]

### I.2.2. Structure D'un Système Automatise de Production (SAP)

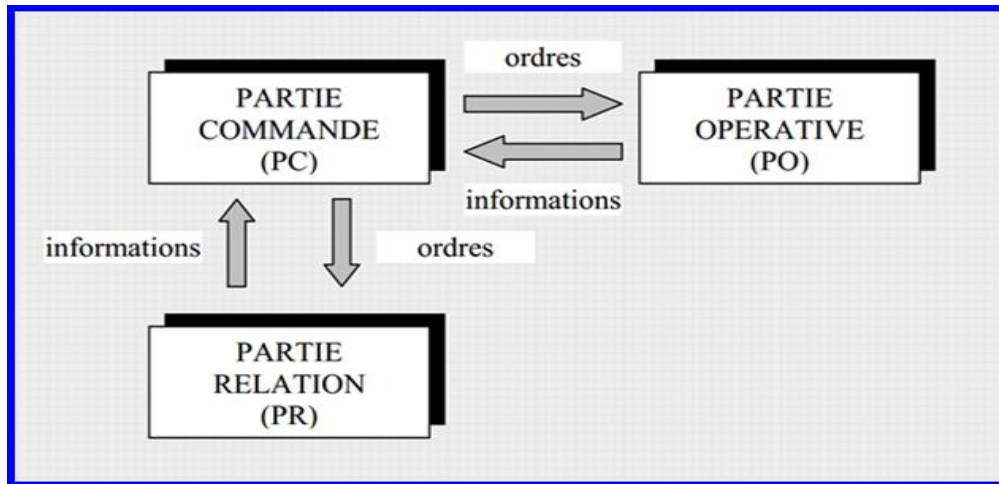
Un système est dit automatisé lorsque le processus qui permet de passer d'une situation initial à la situation finale, se fait sans intervention humaine et que ce comportement est répétitif. [2]

Un système automatisé réalise un certain nombre d'actions appelées « tâches ».

Un système automatisé accomplit une suite d'opérations, appelée «cycle», depuis un état initial jusqu'à un état final. Deux types de cycle :

Cycle	Ouvert	Fermé
Définition	Les tâches s'enchaînent et sans aucune vérification	Les tâches ne se déclenchent que lorsque c'est nécessaire
Exemple	Feux rouge (fonctionnement identique le jour et le nuit)	Distributeur de boissons

**Tab.I.1.** Types du cycle



**Figure.I.1.** Structure D'un Système Automatisé



Il est constitué de trois parties

**I.2.2.1. La Partie Opérative (PO)**

Qui opère sur la matière d'œuvre et le produit. Elle regroupe :

- **les effecteurs**

Qui agissent sur la matière d'œuvre (pales de ventilateurs...) (tout organe en contact avec la matière d'œuvre).

Système	Effecteur
 <p data-bbox="411 1597 555 1630">Ventilateur</p>	 <p data-bbox="978 1597 1225 1630">pale de ventilateur</p>

**Figure.I.2.** Les effecteurs

- **les actionneurs**

Éléments chargés de convertir l'énergie afin de l'adapter au besoin de la partie opérative ; cette énergie étant ensuite consommée par les effecteurs (moteur, vérin, électroaimant, résistance de chauffage, etc.)



moteur électrique



moteur pneumatique



vérin pneumatique

**Figure.I.3** Les actionneurs

- **les pré-actionneurs**

Est un constituant dont le rôle est de distribuer, sur ordre de la partie commande, l'énergie utile aux actionneurs. Les pré-actionneurs les plus utilisés sont les contacteurs (pour les moteurs électriques) et les distributeurs (pour les vérins pneumatiques).



distributeur pneumatique

distributeur électro  
pneumatique

Contacteur

**Figure.I.4.** Les pré-actionneurs

- **les capteurs**

Est un élément de prélèvement et de codage d'informations sur un processus ou sur l'environnement du système. Il convertit une grandeur physique (position, vitesse,...) en une information appelée compte-rendu et compréhensible par la Partie Commande.



**Figure.I.5.** Les capteurs

### I.2.2.2. La Partie Relation (PR)

Qui comporte le pupitre de dialogue homme-machine équipé des organes de commande permettant la mise en/hors énergie de l'installation, la sélection des modes de marche, la commande manuelle des actionneurs, la mise en référence, le départ des cycles, l'arrêt d'urgence... ainsi que des signalisations diverses telles que voyants lumineux, afficheurs, écrans vidéo, Klaxons, sonneries, etc. [2]



**Figure.I.6.** Salle de contrôle

### I.2.2.3. La Partie Commande (PC)

Elle Regroupe les composants (relais électromagnétique, opérateur logique, etc.) et les constituants (API, cartes à microprocesseur, micro-ordinateurs, etc.) destinés au traitement des informations émises par les organes de commande de la PR et capteurs de la PO.

Les ordres résultants sont transmis aux pré-actionneurs de la PO et aux composants de signalisation de la PR afin d'indiquer à l'opérateur l'état et la situation du système. [1]

### I.3. Présentation générale du système

#### I.3.1. Partie opérative

Notre système est contient tous les composants d'un système automatisé, il Ya des capteurs active et passif, et des actionneurs existes dans les systèmes hydrauliques réels, notre système est composé d'une électropompe pour le pompage de l'eau et des électrovannes proportionnelle et tout ou rien, et une petite pompe pour simuler les perturbations.

##### I.3.1.1. Les actionneurs

###### ➤ L'électropompe

Le pompage de l'eau à partir d'un réservoir de plastique placer en bas vers deux cuves en verre placer en haut est fait par une électropompe spéciale en bronze. Cette électropompe à une plage de performance importante, (débit jusqu'à 1200L/Min, et hauteur d'élévation jusqu'à 41m).



Figure I.7. Vue réelle de L'électropompe.

Cette pompe est entraînée par un moteur asynchrone monophasé, il est commandé par un variateur de vitesse Altivar 31, le moteur à une vitesse nominale de 2900 Tr/Min.



Figure I.8. Plaque signalétique de moteur

➤ **L'électrovanne proportionnelle**

Dans notre système il y'a trois électrovannes proportionnelles, elles s'ouvrent et se ferment en fonction du signal analogique de commande présent dans leurs entrées, nous avons utilisé deux électrovannes (VM1, VM2) pour varier le débit d'entré de l'eau et la troisième électrovanne est utilisée pour simuler les perturbations du système.



**Figure I.9.** Vue réelle de L'électrovanne proportionnelle.

➤ **Les électrovannes tout ou rein**

Pour faire l'acheminement de l'eau selon notre besoin, nous avons utilisé quatre électrovannes tout ou rien (Ev1, Ev2, Ev3, Ev4), ces vannes sont seulement deux états, soit ouverte soit fermée.



**Figure I.10.** L'électrovanne TOR

➤ **La pompe de perturbation**

Pour simuler des perturbations sur notre système, nous avons utilisé une petite pompe, placée en bas des cuves. Elle est commandée en tout ou rien à partir d'un contacteur, en état



active, cette pompe absorbe une quantité d'eau selon le pourcentage de l'ouverture de l'électrovanne VM3.



Figure I.11. Vue réelle de La pompe de perturbation

### I.3.1.2. Les capteurs

#### ➤ Capteur de pression

La régulation en boucle fermée du niveau de la cuve nécessite la connaissance de la valeur du niveau après chaque période, pour cela nous avons utilisé un capteur de pression pour mesurer le niveau de la cuve, il est placé au-dessous de la cuve, et à partir de la pression créée par l'eau le capteur indique le niveau de l'eau dans la cuve, et envoie un signal analogique concernant la valeur mesurée.

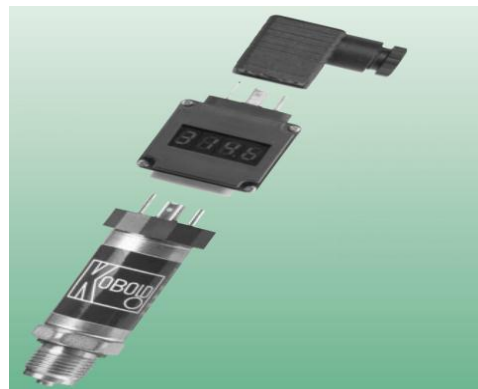


Figure I.12. Capteur de pression

#### ➤ Les Débitmètres

Pour mesurer le débit d'entrée d'eau dans les cuves, et le débit de sortie (perturbation), nous avons utilisé deux débitmètres. Un placé après l'électrovanne VM1, et l'autre placé

après l'électrovanne VM3, chaque débitmètre envoie à l'automate un signal analogique concerne le débit mesuré.



Figure I.13. Débitmètre

#### ➤ Contrôleur de niveau

Pour protéger l'électropompe contre la marche en vide et pour s'assurer que le réservoir situé en bas n'est pas vide, nous avons utilisé un capteur placé en bas de ce réservoir (contrôleur de niveau), si le niveau du réservoir à une valeur minimale (vide) le capteur va envoyer un signal à l'automate pour arrêter l'électropompe.



Figure I.14. Contrôleur de niveau

### I.3.2. Partie commande

La commande de notre système est gérée par un automate **Schneider Premium modulaire**, l'automate contient un module 8 entrées analogiques, un module 8 sorties analogiques, un module 8 entrées tout ou rien et un module 8 sorties tout ou rien, ces modules

sont utilisées pour faire la commande de différents actionneurs et capteurs, pour la commande de l'électropompe nous utilisons un variateur de vitesse (Altivar 31).

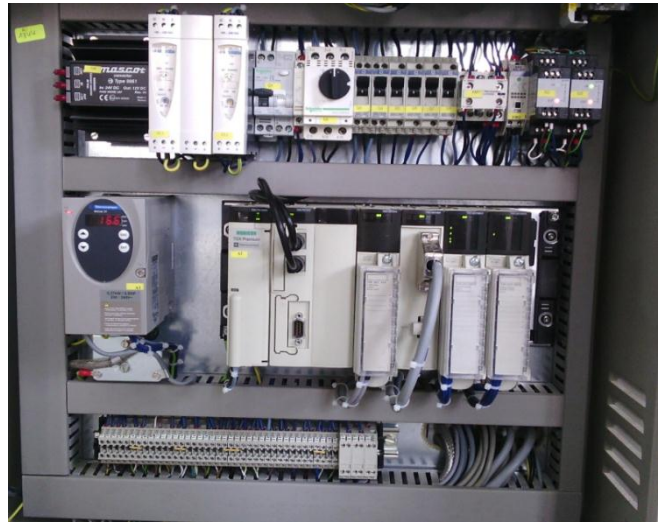


Figure.I.15. Vue réelle de l'armoire de commande du système

La supervision des systèmes est jouée un rôle important pour la visualisation de la démarche des systèmes, elle est aussi utilisée pour commander des systèmes supervisés, dans notre projet, nous avons utilisé un écran à clavier (XBT F024610) pour faire la supervision du système étudié.



Figure.I.16. Vue réelle de Magelis XBT- F024610

### I.3.2.1. L'automate programmable industriel

- **historique**

Les automates programmables industriels sont apparus à la fin des années soixante aux Etats Unis, à la demande de l'industrie automobile américaine (General Motoriser

leader), qui réclamait plus d'adaptabilité de ses systèmes de commande. Ce n'est qu'en 1971 qu'ils firent leur apparition en France.

Les années soixante-dix connaissent une explosion des besoins industriels dans le domaine de l'automatique, de la flexibilité et l'évolutivité des Systèmes Automatisés de Production (SAP). [3]

- **Définition générale**

L'automate programmable industriel A.P.I ou Programmable Logic Controller PLC est un appareil électronique programmable. Il est défini suivant la norme française EN-61131-1, adapté à l'environnement industriel, et réalise des fonctions d'automatisme pour assurer la commande de pré actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. C'est aujourd'hui le constituant essentiel des automatismes. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services et dans l'agriculture. [3]

La force principale d'un automate programmable industriel API réside dans sa grande capacité de communication avec l'environnement industriel. Outre son unité centrale et son alimentation, il est constitué essentiellement de modules d'entrées/sorties, qui lui servent d'interface de communication avec le processus industriel de conduite.

Et il a comme rôles principaux dans un processus

- D'assurer l'acquisition de l'information fournie par les capteurs ;
- En faire le traitement ;
- Elaborer la commande des actionneurs ;
- Assurer également la communication pour l'échange d'informations avec l'environnement.

- **Architecture des automates programmables industriels**

De forme compacte ou modulaire, les automates sont organisés suivant l'architecture suivante

- Un module d'unité centrale ou CPU, qui assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées/sorties, des mémoires RAM et EEPROM nécessaire pour stocker les Programmes, les données, et les paramètres de configuration du système.
- Un module d'alimentation qui, à partir d'une tension 220V/50Hz ou dans certains cas de 24V fournit les tensions continues +/- 5V, +/-12V ou +/- -15V.
- Un ou plusieurs modules d'entrées 'Tout ou Rien' ou analogiques pour l'acquisition des informations provenant de la partie opérative (procédé à conduire).
- Un ou plusieurs modules de sorties 'Tout ou Rien' (TOR) ou analogiques pour transmettre à la partie opérative les signaux de commande. Il y a \_\_\_20 des modules qui intègrent en même temps des entrées et des sorties.
- Un ou plusieurs modules de communication. [3]

### I.3.2.2. L'automate Schneider Premium

La commande de système est assurée par l'automate Schneider **TSX 57253**, à bas de cet automate on fait l'identification et la régulation du système hydraulique.



**Figure I.17.** Vue réelle de L'automate Schneider Premium TSX 57253.

L'automate est composé des modules suivant

- ♦ **Modules alimentation**

Le module alimentation **TSX PSY 2600M** est destiné à l'alimentation de rack et de ses modules installés.

Le module alimentation, doit équiper le rack TSX RKY p/pEX (emplacement repéré PS). La puissance nécessaire à l'alimentation du rack TSX RKY est fonction du Type et du nombre de modules implantés dans celui-ci. De ce fait, il est nécessaire D'établir un bilan de consommation du rack afin de définir le module Alimentation TSX PSY adapté à ce rack. [2]

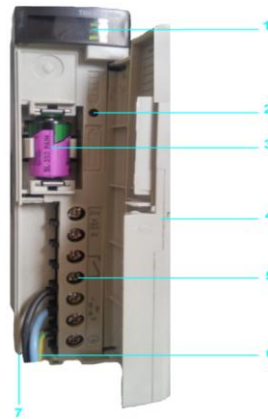


Figure I.18. TSX PSY 2600M

1. Un bloc de visualisation.
2. Un bouton-poussoir RESET à pointe de crayon provoquant une reprise à chaud de l'application.
3. Un emplacement recevant une pile assurant la sauvegarde de la mémoire RAM interne du processeur.
4. Un volet assurant la protection de la face avant du module.
5. Un bornier à vis permettant le raccordement :
  - au réseau d'alimentation,
  - du contact du relais alarme,
  - de l'alimentation capteurs pour les alimentations à courant alternatif (selon modèle).
6. Un passage pour collier de serrage des câbles.
7. Un fusible situé sous le module et assurant la protection :
  - de la tension 24 VR sur l'alimentation à courant continu non isolée avec TSX PSY 3610,
  - de la tension primaire sur les autres alimentations. [4]

♦ **Processeur**

Le processeur **TSX P57253** gère l'ensemble d'une station automate constituée de modules d'entrées/sorties TOR, de modules d'entrées/sorties analogiques qui peuvent être répartis sur le rack.



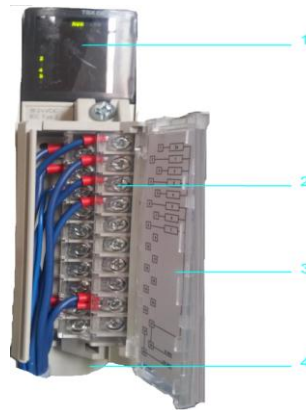
**Figure I.19.** Le processeur TSX P57253

1. Un bloc de visualisation comprenant 5 voyants.
2. Un bouton RESET provoquant un démarrage à froid de l'automate lorsqu'il est actionné.
3. Un connecteur mini-DIN femelle 8 contacts repéré TER pour le raccordement d'un terminal de programmation ou de réglage.
4. Un connecteur mini-DIN femelle 8 contacts repéré AUX pour le raccordement d'un périphérique.
5. Un emplacement pour une carte d'extension mémoire au format PCMCIA type I.
6. Un emplacement pour une carte de communication au format PCMCIA type III. Peut recevoir la carte extension mémoire SRAM de 2 Mo pour stockage de données (TSX MRP DS 2048P).
7. Un connecteur SUB-D 9 contacts pour communication Fipio gestionnaire de bus. [4]

♦ **Module entrée TOR**

Le module 8 entrées TOR **TSX DEY08D2**, est utilisé pour accueillir les signaux tout ou rien des deux capteurs, le premier est pour l'état du variateur de vitesse si il est en marche

ou en arrêt, et la deuxième est pour *Contrôler de niveau* qui détecte si le réservoir est au niveau bas (vide).



**Figure I.20.** Module entrée TOR TSX DEY08D2.

1. Un bloc de visualisation des voies et de diagnostic du module.
2. Un bornier à vis débrochable pour raccordement direct des entrées/sorties aux capteurs et pré actionneurs.
3. Une porte pivotante permettant l'accès aux vis du bornier servant également de support à l'étiquette de repérage.
4. Un support rotatif comprenant le dispositif de détrempage. [4]

#### ♦ **Module sortie TOR**

Les électrovannes tout ou rien (Ev1, Ev2, Ev3, Ev4) sont les responsables sur la direction de l'eau, elles sont reliées aux borniers du module 8 sorties TOR **TSX DSY 08R5**, la pompe de perturbation est aussi reliée à ce module.

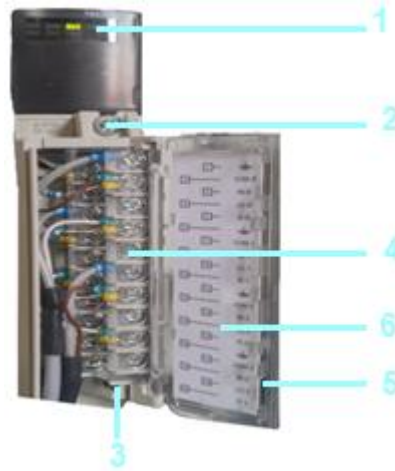


**Figure I.21.** Module sortie TOR TSX DSY 08R5



#### ♦ Module entrées analogiques

Les capteurs de pression, et les débitmètres produisent des signaux analogiques dépendent à les valeurs des mesures captée, ils sont reliés aux borniers du module 4 entrées analogiques **TSX AEY414**.



**Figure I.22.**Module entrées analogiques TSX AEY414

1. Un bloc de visualisation et de diagnostic du module.
2. Un connecteur recevant le bornier à vis.
3. Un support rotatif comprenant le dispositif de détrempage.
4. Un bornier à vis débrochable pour raccordement direct des entrées/sorties aux capteurs et pré actionneurs.
5. Une porte pivotante permettant l'accès aux vis du bornier servant également de support à l'étiquette de repérage.
6. Un codeur du bornier à vis. [4]

#### ♦ Module sortie analogique

L'électropompe et les trois électrovannes proportionnelles (VM1, VM2, VM3) sont commandées avec un signal analogique produit par l'automate, ce signal est envoyé à l'actionner concernée à travers le module 8 sorties analogiques **TSX ASY 800**, ces actionnaires sont reliés au module avec un Connecteur type SUB-D 25 contacts.



Figure.I.23.Module sorties analogique TSX ASY 800.

### I.3.2.2.Le variateur de vitesse

- **Introduction**

Pour les systèmes nécessitant un contrôle de débit ou de pression, et plus particulièrement pour les systèmes à fortes pertes par frottement, la méthode permettant les plus fortes économies d'énergie est le contrôle de la vitesse par variateur de vitesse. Bien que les machines à courant alternatif soient connues depuis le XIXe siècle, c'est l'amélioration des variateurs de vitesse (grâce aux progrès de l'électronique de puissance) qui va leur permettre de s'imposer sur les machines à courant continu. En particulier, la machine synchrone était autrefois difficile à commander faute de dispositif garantissant la condition des synchronismes entre le rotor et le stator. On réalisait alors un accrochage, c'est-à-dire un entraînement du rotor à l'aide d'une génératrice à la vitesse souhaitée avant d'alimenter les stators. [5]

- **Définition**

Un variateur de vitesse est un dispositif électronique destiné à commander la vitesse d'un moteur électrique.

Les variateurs de vitesse sont constitués principalement d'un convertisseur statique et d'une électronique de commande. Les variateurs récents contiennent aussi un étage de correction du facteur de puissance afin de respecter les normes de compatibilité électromagnétique. En général, le convertisseur statique est un hacheur ou un onduleur. L'électronique de commande réalise la régulation et l'asservissement de la machine à travers le convertisseur statique de sorte que l'utilisateur puisse commander directement une vitesse.

Sa conception dépend essentiellement de la stratégie de commande choisie (commande vectorielle, commande scalaire, etc.) [5]

Le variateur de vitesse est composé essentiellement :

- ✚ d'un redresseur qui, connecté à une alimentation triphasée (le réseau), génère une tension continue à ondulation résiduelle (le signal n'est pas parfaitement continu). Le redresseur peut être de type commandé ou pas,
- ✚ d'un circuit intermédiaire agissant principalement sur le "lissage" de la tension desortie du redresseur (améliore la composante continue). Le circuit intermédiaire peut aussi servir de dissipateur d'énergie lorsque le moteur devient générateur,
- ✚ d'un onduleur qui engendre le signal de puissance à tension et/ou fréquence variables,
- ✚ d'une électronique de commande pilotant (transmission et réception des signaux) le redresseur, le circuit intermédiaire et l'onduleur.

### I.3.2.2.1. Le variateur Altivar 31

Le but de notre travail est le réglage du débit de l'eau pompé à partir du réservoir situer en bas vers les deux cuves qui sont placés en haut, la fonction de pompage de l'eau et faite à l'aide d'une pompe entraînée avec un moteur asynchrone monophasé, pour la commande de ce moteur on utilise un variateur de vitesse **Altivar 31**, nous avons paramétrée le variateur pour commander et protéger le moteur.



Figure.I.24. Vue réelle du variateur Altivar 31.

Le variateur Altivar 31 est un convertisseur de fréquence pour moteurs asynchrones triphasés à cage. L'Altivar 31 est robuste, peu encombrant, facile à mettre en œuvre.

Les variateurs Altivar 31 communiquent sur les bus industriels Modbus et CANopen. Ces deux protocoles sont intégrés en standard dans le variateur. [5]

- **Fonctions**

Le variateur Altivar 31 dispose de six entrées logiques, de trois entrées analogiques, d'une sortie logique/analogique et de deux sorties à relais.

Les principales fonctions intégrées sont les suivantes :

- protections moteur et variateur,
- rampes d'accélération et de décélération, linéaires, en S, en U et personnalisées
- plus vite/moins vite,
- 16 vitesses pré sélectionnées,
- consignes et régulateur PI,
- commande 2 fils/3 fils,
- logique de frein,
- rattrapage automatique avec recherche de vitesse et redémarrage automatique,
- configuration des défauts et des types d'arrêts
- sauvegarde de la configuration dans le variateur...

Plusieurs fonctions peuvent être affectées sur une même entrée logique.

### **I.3.2.3. Terminal Magelis à écran graphique**

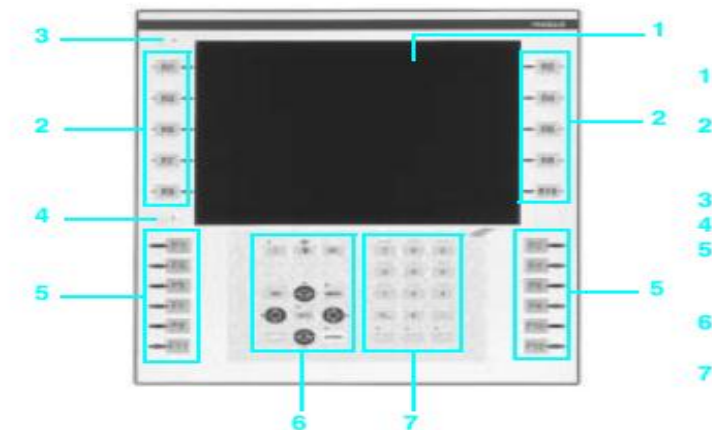
la supervision est une partie très importante dans la réalisation des systèmes automatisés, elle nous permet de visualiser la démarche du système elle offre aussi la possibilité de piloter et gérer le système supervisé ,pour bénéficier de ces fonctions importants ,nous utilisons un terminal Magelis à écran graphique **XBT F024610** ,elles se place en haut à gauche dans la machine , nous utilisons cette Magelis pour tracer les graphes de la consigne ,la sortie du système et la commande, on l'utilise elle aussi pour gérer et commander le système.

### I.3.2.3.1. Terminal graphique à clavier

Les terminaux XBT F sont spécialisés pour les fonctions de dialogue opérateur graphique, elles sont parmi les terminaux les plus utilisées pour les systèmes automatisés, à travers le rôle de Visualiser des données issues de l'automatisme, et de Commander l'automatisme par des commandes Tout ou Rien.

Les terminaux graphiques XBT F communiquent avec les automates programmables par une liaison série intégrée point à point ou multipoint, ou bien par un bus de terrain par l'intermédiaire d'une carte PCMCIA type III.

Les protocoles de communication sont ceux des automates de Schneider Electric, et ceux des autres principaux constructeurs du marché. Les terminaux graphiques XBT F (10,4") peuvent aussi être connectés sur réseau Ethernet TCP/IP. [6]



**Figure.I.25.**Le terminal Magelis à écran graphique XBT F024610.

- Un écran monochrome ou couleur (5,7" ou 9,5" ou 10,4" selon modèle).
- 2 x 4 ou 2 x 5 (selon modèle) touches fonctions dynamiques avec voyants de signalisation.
- 3. Un voyant de contrôle de communication.
- Un voyant occupation clavier.
- 2 x 5 ou 2 x 6 (selon modèle) touches fonctions statiques avec voyants de signalisation, étiquettes relégendables.
- 6. Douze touches services avec voyants de signalisation.

- 7. Douze touches alphanumériques (0...9, +/- .) associées à 3 touches d'accès alphabétique (A...Z).

Les terminaux graphiques XBT F proposent les fonctions suivantes :

- affichage de synoptiques animés, commande, modification de variables numériques ou alphanumériques,
- affichage d'une ligne de service (bandeau d'états et d'alarmes) avec heure courante,
- visualisation dynamique des données d'exploitation (consignes, mesures, recettes, messages de maintenance) et de défauts de procédé,
- commande par touches fonctions dynamiques ou statiques,
- mise à l'échelle des variables analogiques,
- courbes temps réel et courbes de tendance,
- historique d'alarmes et gestion de groupes d'alarmes,
- gestion de pages aides, de pages formulaires, de pages recettes,
- appels de pages à l'initiative de l'opérateur ou à l'initiative de l'automate,
- trois niveaux de mots de passe,
- impressions de pages formulaires, de l'historique horodaté et des alarmes,
- support de l'application du protocole de communication dans la carte mémoire application au format PCMCIA type II.

La variable qui anime un objet peut être directement sélectionnée dans une liste de symboles issus du logiciel PL7 ou du logiciel Concept. Le programme d'application des terminaux et stations graphiques est stocké dans une carte mémoire au format PCMCIA. [6]

#### **I.4. Présentation du fonctionnement du système**

L'unité régulation de débit et niveau d'eau permet de mettre en évidence les difficultés rencontrées dans des processus industriels réels.

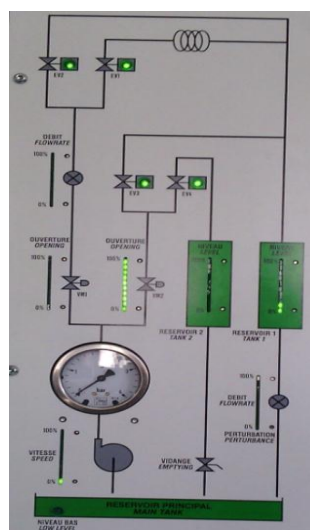
Elle permet de réaliser :

- Une régulation à une colonne avec possibilité d'ajouter un élément de retard pur.
- une régulation à deux colonnes simulant des rapports de volume.



**Figure.I.26.**Vue réelle de l'unité régulation de débit et niveau d'eau.

La visualisation du synoptique de l'installation et le contrôle des phénomènes de régulation sont assurés par l'intermédiaire d'un terminal de dialogue opérateur Magelis ou encore directement sur le banc.



**Figure.I.27.** La visualisation directe sur le banc.

## I.4.1. Plane hydraulique du système

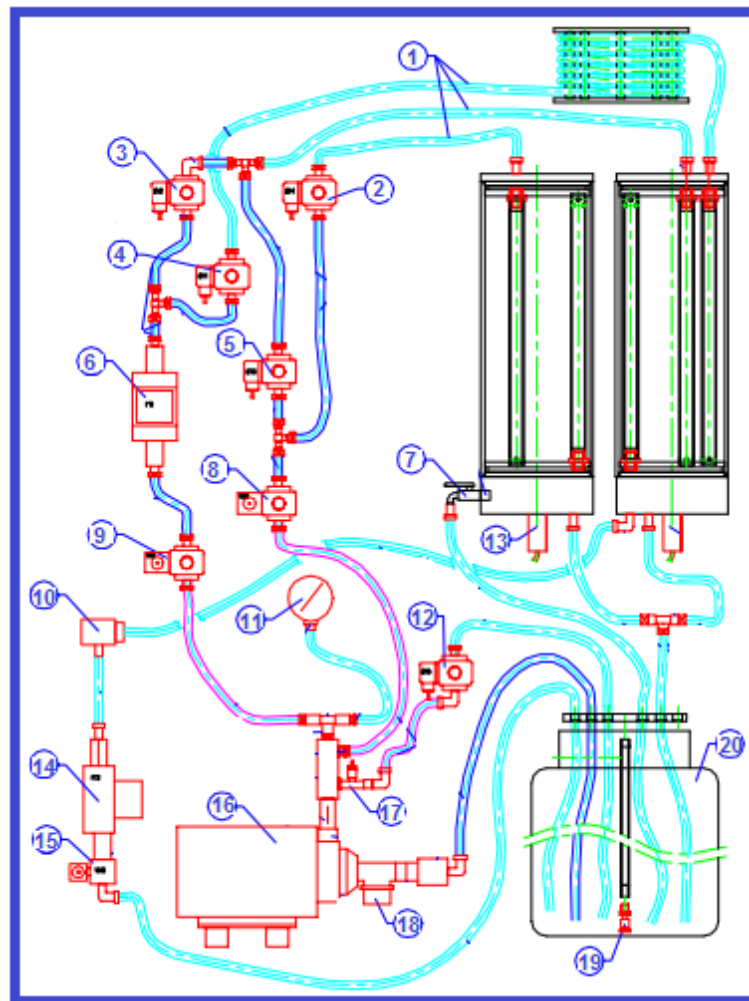


Figure.I.28. Le schéma hydraulique du banc d'essais

- |                                        |                                         |
|----------------------------------------|-----------------------------------------|
| 1. les Canales                         | 11. manomètre                           |
| 2. électrovanne TOR 01 (EV1)           | 12. électrovanne TOR passe bas          |
| 3. électrovanne TOR 02 (EV2)           | 13. capteur de niveau d'eau             |
| 4. électrovanne TOR 03 (EV3)           | 14. débitmètre 02 (débit de perd)       |
| 5. électrovanne TOR 05 (EV4)           | 15. électrovanne proportionnel 03 (VM3) |
| 6. débitmètre 01                       | 16. La pompe                            |
| 7. vanne manuel de vidange             | 17. Capteur de pressions                |
| 8. électrovanne proportionnel 01 (VM1) | 18. Filtre d'eau                        |
| 9. électrovanne proportionnel 02 (VM2) | 19. Sonde de niveau bas de l'eau        |
| 10. pompe de perturbations             | 20. Réservoir de source                 |



En trouve sur cette visualisation Des bargraphes permettent d'observer les différents débits, l'ouverture des vannes proportionnelles, la vitesse de la motopompe et le niveau d'eau dans les deux réservoirs, et aussi des pictogrammes symbolisent les vannes tout-ou-rien. Ils s'allument lorsque la vanne concernée est ouverte.

Le banc est offre plusieurs possibilités de commande du système, on a trios actionneurs VM1, VM2, et l'électropompe, et les vannes tout-ou-rien EV1 à EV4 permettent de définir le chemin suivi par le liquide, et selon le choix du l'actionneur a commandé, et l'état des vannes TOR, on a une combinaison des Choix possibles.

- ✚ La première possibilité est de prendre l'électropompe comme l'actionneur à commander, dans ce cas on a deux possibilité, soit on ouvre la vanne VM1, et on fait un réglage de débit avec ou sans retard pur selon l'état des vannes EV1 et EV2 .Ou on ouvre la vanne VM2, et en fermée la vanne VM1. Dans ce cas, on fait une régulation du niveau du réservoir 1 ou réservoir 2 selon l'état des vannes EV3et EV4.
- ✚ Dans La deuxième possibilité nous appliquons la commande sur la vanne VM1, dans ce cas on fixe la vitesse de l'électropompe manuellement, et on fait une régulation de niveau ou débit avec ou sans retard à travers l'état de les vannes, EV1et EV2.
- ✚ Un troisième choix est possible si on applique la commande sur la vanne VM2, dans ce cas on obtient une régulation du niveau avec ou sans perturbation, selon l'état des vannes EV3et EV4.
- ✚ Et pour une commande multi variables, nous appliquons la commande sur les trois actionneurs (VM1, VM2, et l'électropompe), dans ce choix on a la possibilité de obtient une multi-sorties, on prend comme sorties le débit et les niveaux des réservoirs

## **I.5. Les logiciels utilisés**

Les logiciels de programmation des automates programmables et des HMI sont un rôle plus important pour l'installation et la mise en marche des systèmes industriels automatisés, une bonne maitrise de ce logiciel offre à l'automaticien plusieurs méthodes et solutions possibles pour la réalisation de l'heur projets.

Dans ce projet nous utilisons deux logiciel, le premier est PL7 qui utilisée pour la programmation et le développement des algorithmes d'identification et de commande intégrés sous l'automate Schneider premium, le deuxième logiciel est XBT-L1000, nous utilisons ce logiciel pour fait le dessine de la supervision de notre système sur l'écran graphique.

### I.5.1. Le logiciel PL7

Le logiciel PL7 Pro est le logiciel de programmation des automates TSX 37 et TSX/PMX/PCX 57 fonctionnant sous Windows, il propose une multitude de langages de programmation des automates, leur bibliothèque est nous donne tous qui nécessaire pour traduire la démarche d'un automatisme selon notre besoin. [7]

#### I.5.1.1. L'éditeur de configuration

L'éditeur de configuration de PL7 assure pour chaque application les fonctions suivantes :

- la configuration matérielle,
- la configuration logicielle,
- la configuration des objets Grafcet dans le cas de la programmation en langage grafcet.

##### ▪ Editeur de configuration matérielle

L'éditeur de configuration matérielle de PL7 permet de manière intuitive et graphique, de déclarer et configurer les différents éléments constitutifs de l'automate:

- le rack,
- l'alimentation
- le processeur
- le module métier

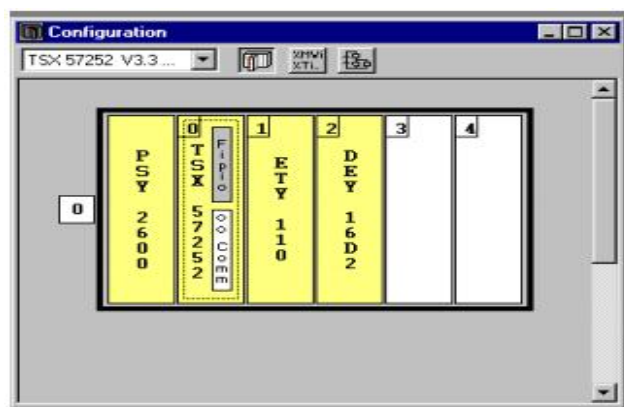


Figure.I.29. Editeur de configuration matérielle

**▪ Editeur de configuration logicielle**

L'éditeur de configuration logiciel de PL7 assure le paramétrage logiciel de L'application en renseignant le nombre de blocs fonctions, de registres et la taille des zones de variables globales...etc.

**▪ Editeur de configuration des objets grafcet**

Dans le cas d'une programmation grafcet, l'éditeur de configuration permet de définir les objets grafcet (étapes, macro-étapes...) et les paramètres d'exécution (nombre d'étapes et de transitions actives).

**I.5.1.2. La programmation del'application****▪ L'éditeur de programme**

Les éditeurs de programme permettent la programmation des fonctions et métiers mis en œuvre dans l'application. Quatre éditeurs programme sont proposés :

- éditeur langage à contact (LD),
- éditeur langage listes d'instructions (IL),
- éditeur langage littéral structuré (ST),
- éditeur langage Grafcet (G7),

**I.5.2. Le logiciel XBT L1000**

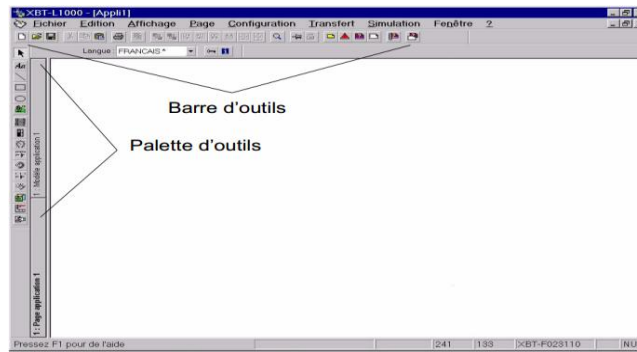
Les terminaux munis d'écrans graphiques couleurs permettent la supervision du fonctionnement du système et la visualisation de l'état de la partie opérative ou des grandeurs physiques (valeurs, tendances, courbes...) de manière dynamique.

Les terminaux de dialogue Magelis à écrans graphiques sont configurables avec le logiciel XBT L1000, dans l'environnement Windows.

Pour les terminaux et les stations graphiques, le logiciel XBT L1000 propose une bibliothèque d'objets graphiques animés tels que barographes, vumètres, sélecteurs, potentiomètres, et courbes de tendances.

**I.5.2.1. L'éditeur d'application**

La fenêtre principale permet de réaliser les différentes pages.



**Figure.I.30.L**’éditeur d’application

### ▪ La barre d’outils

La barre d’outils permet de réaliser différentes fonctions comme :

- 1 Création de pages.
- Couper/Copier/Coller des objets.
- Mise au premier plan/mise en arrière-plan des objets.
- Alignement d'objets

### ▪ La palette d'outils

La palette d’outils présente les icônes permettant de manipuler les objets, Pour utiliser un outil, il suffit de le sélectionner dans la palette et de le positionner sur la page.

## I.5.2.2. Objet touche fonction

### ▪ Touches fonctions statiques

Les touches fonctions statiques sont définies pour l'ensemble de l'application. Elles n'agissent que sur des bits de la table de dialogue.

Elles peuvent avoir les fonctionnalités suivantes :

- Commande impulsionnelle,
- Commande pousser/pousser,
- Accès à une page.

### ▪ Touches fonctions dynamiques

Les touches fonctions dynamiques sont associées à une ou plusieurs pages. Les touches de fonctions dynamiques peuvent avoir les fonctionnalités suivantes :

- Commande impulsionnelle.

- Commande pousser/pousser.
- Ecriture valeur.
- Accès à une page.
- Positionnement sur un champ de saisie.

## I.6. Supervision du système

Pour une visualisation générale du système et des résultats obtenus, nous avons fait une supervision pour représenter la démarche de notre système, nous utilisons une Magelis XBT F02462 graphique à clavier.

Pour la configuration de ce Magelis, et pour la création de l'application, nous utilisons le logiciel XBT-L1000.

Dans un premier temps, nous configurons une communication entre Magelis et l'automate, nous choisissons le protocole unitelway pour cet effet, la Magelis est connectée avec le port TER de l'automate avec le câble XBT915, après nous configurons la table de dialogue pour notre application, nous choisissons le mot %mw100 comme le début de cette table.

Dans notre application de supervision, nous proposons quatre écrans. Le changement entre ces écrans se fait à partir des fonctions dynamiques du Magelis.

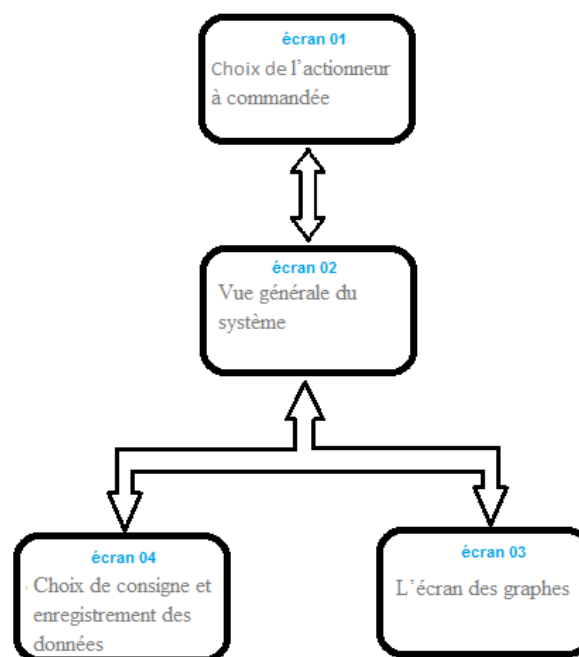


Figure.I.31.Organigramme d'écrans de la supervision

### I.6.1. Les écrans de supervision

#### ✚ Écran 01

Cet écran est l'écran d'accueil, quand Magelis s'allume s'affiche-t-il.



Figure.I.32. L'écran 01

L'écran 01 nous permet de choisir l'actionneur sur lequel nous allons appliquer la commande, le choix se fait à partir les deux flèche sur le côté droit de l'écran.

#### ✚ Écran 02

L'écran 02 est donne une vision générale sur le système, nous trouvons dans cet écran des baragraphes représentons la variation des grandeurs mesurées par les capteurs ('niveau d'eau, débit d'entrée, pourcentage d'ouverture des électrovannes...), et un potentiomètre représentant la vitesse de l'électropompe.

Dans cet écran, on a la possibilité d'accès à l'écran 03 qui contient les graphes des mesures et sorties du système et l'accès aussi à l'écran 04 pour choisir le type de consigne ainsi que le démarrage de l'enregistrement des données sous automate.

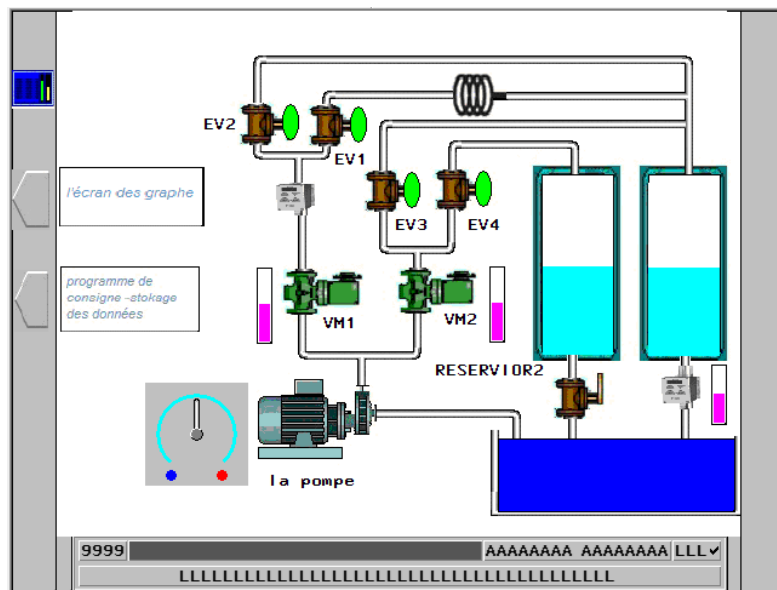


Figure.I.33. L'écran 02

**Ecran 03**

Les graphes sont parmi les outils le plus utilisée pour la lecture des informations, elles sont simple et facile pour tracer, dans cet écran nous trouvons les graphes de consigne, commande et mesure du système, on a aussi la possibilité de lecture et écriture les déférents valeurs concerne le système (paramètre PID,  $U_c$  (consigne),  $U$  (commande)...).

Les quatre graphes sont tracé dans la même fenêtre, avec une fenêtre de 20 minutes, au-dessus de cet écran on a un voyions allumé en rouge si le programme de consigne est en mode stop, et en mode run ce voyions est allumé en vert.

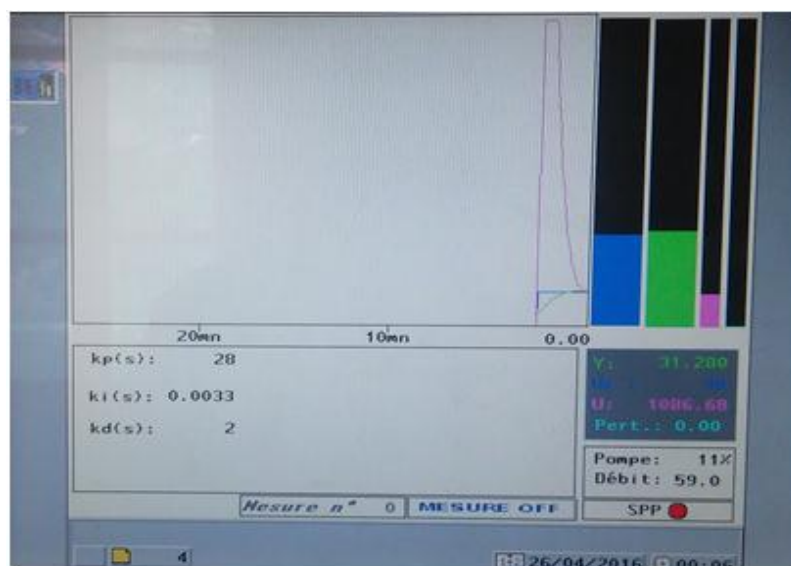


Figure.I.34. L'écran 03

### ✚ Ecran 04

Dans la partie de l'identification et régulation de niveau d'eau dans les cuves ,nous utilisons deux types de consigne ,un échelon et un signal aléatoire (programme de consigne), l'écran 04 offre cette possibilité ,en a deux cas du programme de consigne si il est en run la consigne du système réel est le même que la consigne programmé par le programmeur de consigne ,et si il est en mode stop la consigne réel est posé à partir le claviers directement.

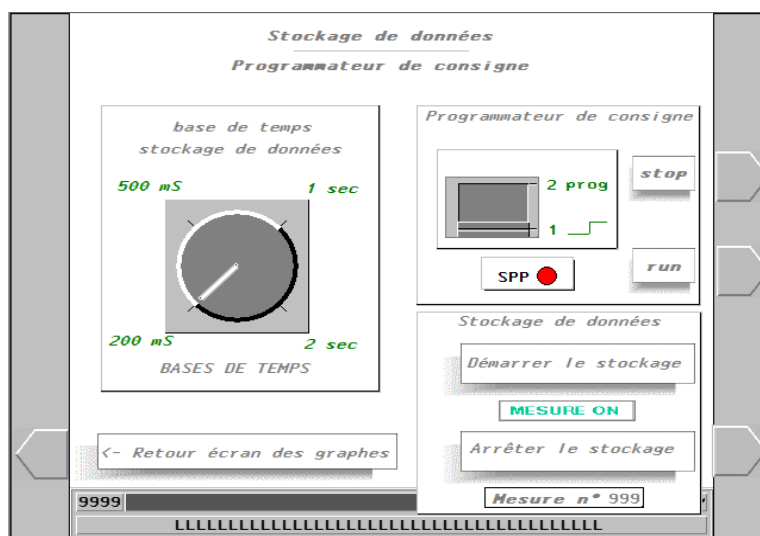


Figure.I.35. L'écran 04

Les résultats de l'identification du système sont tracée par le logiciel MATLAB à travers les mesures enregistrées sous l'automate, pour cette effet l'écran 04 offre la possibilité d'enregistrée 500 valeurs, on aussi la possibilité de choisi la base de temps utilisée pour l'enregistrement.

## I.7. Conclusion

L'automate programmable est parmi les outils les plus importants et les plus utilisées dans ces dernières années, il est fréquemment utilisé dans les stations des traitements et filtrage des eaux, et dans les unités de pompage de l'eau, notre travail à comme objectif de faire une régulation de débit et de niveau de l'eau dans deux réservoirs.

Dans ce chapitre nous avons présenté le système hydraulique sur lequel notre étude à été faite, nous avons expliqué la partie opérative (actionneurs, capteurs...) et la partie



commande (automate, variateur de vitesse...) Du système, puis nous avons montré les différentes possibilités de commande que nous offre le système, à la fin du chapitre nous avons fait une supervision du système avec un écran graphique à clavier, ou nous avons expliqué les différents pages créer pour cette effet.

**Chapitre II**  
**Identification du système**  
**hydraulique**

## II.1. Introduction

L'identification des systèmes est parmi les bases de la théorie de la commande, elle est utilisée pour obtenir un modèle mathématique qui représente la dynamique du système réel a commandé.

Dans ce chapitre nous avons fait l'identification on ligne du système étudié, dans premier temps nous simulons deux algorithmes récursifs de l'identification avec le logiciel MATLAB, ainsi que validation des modèles obtenus avec MATLAB, puis nous implémentons ces algorithmes dans l'automate programmable Schneider premium, en fin nous avons fait une comparaison entre les résultats obtenus avec les différents algorithmes.

## II.2. Définition de l'identification

L'Automatique consiste en l'étude des systèmes réels des différentes disciplines scientifiques (Electronique, mécanique, thermique, chimie, écologie, biologie, économie, sociologie, physique, cosmologie...), en vue de l'analyse, de la prédiction, de la surveillance, de la commande, et / ou de l'optimisation des systèmes. La condition nécessaire pour cela est l'obtention d'un modèle mathématique du système réel (on réalise une modélisation). Un système est un objet dans lequel des variables de différents types interagissent et produisent des signaux observables. Lorsque le modèle du système n'est pas connu, il est nécessaire de procéder à son identification. [8]

L'identification c'est l'opération de détermination des caractéristiques dynamiques d'un procédé (système). On s'intéresse à l'identification des modèles dynamiques paramétriques échantillonnés qui sont les plus appropriés pour la conception et l'ajustement des systèmes numériques [9]

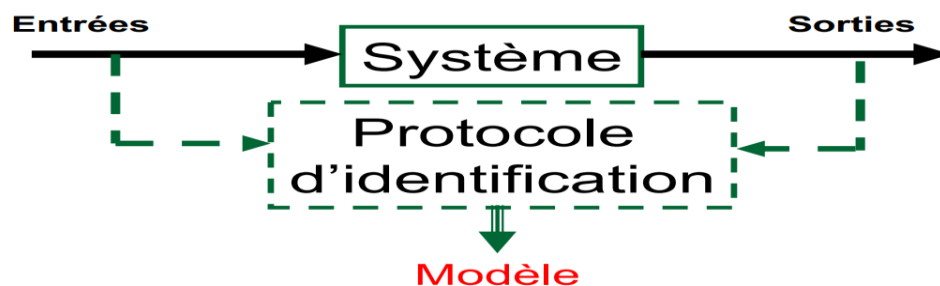


Figure.II.1. Structure de l'identification des systèmes

Le problème d'identification des systèmes suscite un grand intérêt et ce depuis plusieurs décennies. L'activité de recherche menée autour de ce problème a fait l'objet de très nombreux colloques et séminaires et a donné lieu à d'innombrables publications. Si l'identification des systèmes linéaires est maintenant arrivée à maturité, celle des systèmes non linéaires reste à ce jour un thème de recherche loin d'être tari.

### II.3. Choix du signal d'excitation

Pour bien identifier, il faut bien exciter dans tout le spectre de fréquences susceptible de contenir des constantes de temps du système.

- $\sin(\omega t)$  : parfait d'un point de vue spectre (balayage en fréquence) mais peu des systèmes acceptent ce genre d'entrées.

- $\delta(t)$  : parfait d'un point de vue théorique, mais difficile de réaliser une bonne approximation de l'impulsion.

- $u(t)$  : moins bon d'un point de vue spectral ( $u(f) = \frac{1}{2} \delta(f) + \frac{1}{j2\pi f}$ ), mais facile à implanter.

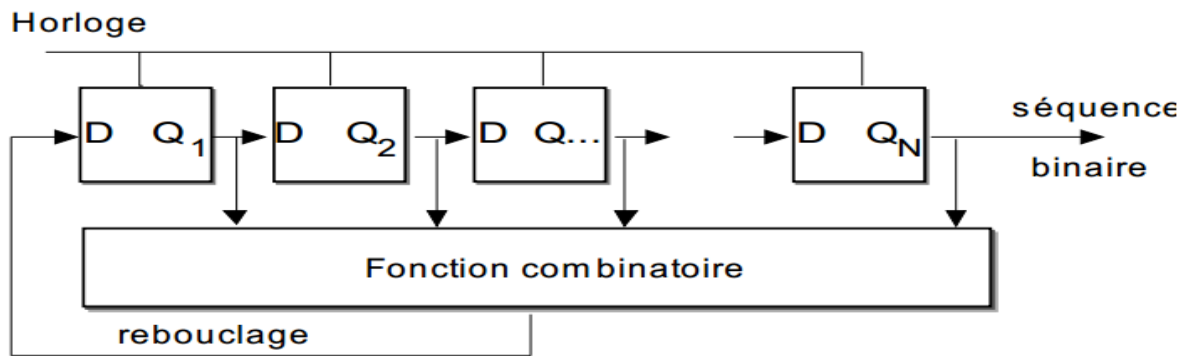
- $b(t)$  : bruit blanc idéal d'un point de vue spectral mais comment le réaliser ?

Il arrive que nous n'ayant aucune possibilité d'exciter le système (ex : machine en production), il faudra alors profiter des commandes « naturelles » du système de signal d'entrée du système. Dans ce cas, le premier travail consiste à calculer le spectre du signal d'entrée (FFT par exemple). Il faudra vérifier a posteriori que les constantes de temps identifiées sont bien dans des domaines de fréquence dans lesquels le système a été excité. [10]

La solution théorique standard est d'appliquer une SBPA (séquence binaire pseudo aléatoire).

#### ❖ Les séquences binaires pseudo aléatoires (SBPA)

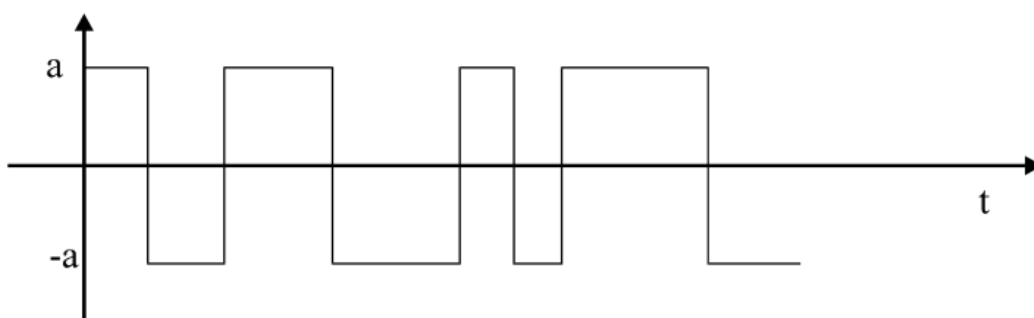
C'est un signal binaire généré par un registre à décalage de N bits rebouclé par une fonction spécifique à base de "OU exclusifs". L'état des N bascules forment un mot de N bits observé par sérialisation. [10]



**Figure.II.2.**Registre générateur de séquence SBPA

Si la fonction de bouclage est bien choisie, le registre peut prendre au mieux toutes les combinaisons possibles des  $N$  bits soit  $2^N$  états différents, sauf la combinaison "0 0 0 ...0" (le registre resterait à une valeur nulle indéfiniment!). Cette structure sert de générateur de mots binaires aléatoires, la séquence étant formée de  $2^N - 1$  valeurs différentes du mot. A partir des  $2^N$  valeurs, il y a répétition de la première valeur ; le générateur "reboucle" sur sa première valeur. La séquence est donc cyclique et se répète tous les  $2^N - 1$  coups d'horloge. [10]

Pour les applications d'identification, on utilise la sortie obtenue par sérialisation (la sortie de n'importe quelle bascule peut servir de sortie série). La sortie du registre binaire présente une succession de "0" et de "1" pseudo-aléatoire. Après translation de niveau et amplification d'un facteur  $a$ , la séquence présente l'allure suivante



**Figure.II.3.** Allure d'une séquence binaire pseudo aléatoire à niveaux symétriques

Le signal obtenu est aléatoire (on ne peut prédire la valeur suivante connaissant la valeur courante) et forme une Séquence Binaire Pseudo Aléatoire. Sa longueur

maximale est donc  $L = 2^{N-1}$ ; à partir des  $2^N$  valeurs, il y a répétition de la première valeur (périodicité de la SBPA). [10]

## II.4. Les méthodes générales d'identification

L'identification est une approche expérimentale pour la détermination du modèle dynamique d'un système. Cette approche peut être décomposée en quatre étapes. [8]

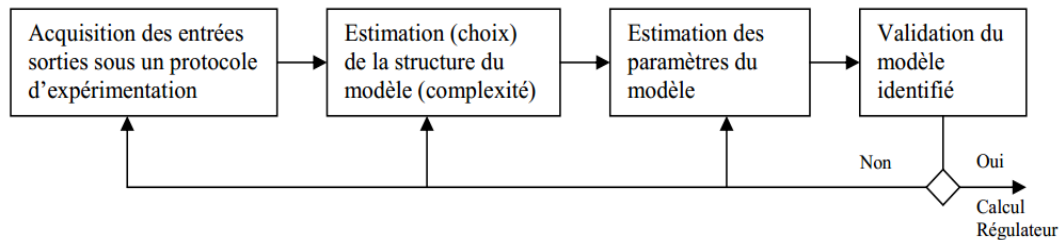


Figure.II.4. Les étapes de l'identification

### II.4.1. Méthodes non paramétriques

- **Analyse fréquentielle**

Les essais expérimentaux harmoniques sont rarement employés, car ils sont longs et fastidieux. Des essais réalisés avec une pulsation comprise entre  $0.1 \cdot \omega_c$  et  $10 \cdot \omega_c$ , avec une grande constante de temps pour le processus, peuvent durer des heures, d'autant plus qu'il est parfois nécessaire de recommencer deux à trois fois l'expérimentation pour chaque pulsation (on doit attendre le temps d'établissement pour chaque mesure). Les outils classiques de l'automatique sont utilisés pour l'analyse de la réponse en fréquence du système à identifier : Bode, Nyquist, Black... [8]

- **Réponse impulsionnelle**

Un système linéaire invariant est entièrement caractérisé par sa réponse à un Dirac, dont la transformée de Fourier est constante sur tout le spectre. La transformée de Fourier de la réponse impulsionnelle du système est sa fonction de transfert.

Un Dirac « réel », peut être approché, du point de vue de ses effets, par une impulsion brève, de durée finie et d'énergie limitée. [8]

La réponse du système, est en conséquence faible et bruitée, donc difficilement exploitable. Ainsi, pour atteindre la réponse impulsionnelle  $h(t)$  par voie indirecte, il est

possible d'appliquer les techniques de corrélation. Le processus est alors excité par **une Séquence Binaire Pseudo Aléatoire (S.B.P.A.)**. Notons que l'utilisation d'un générateur de S.B.P.A. pour l'essai ou la simulation d'un système, est aussi connue sous le nom de techniques de Monte-Carlo (très largement utilisées par les physiciens lors de simulations).

Ajoutons, que les signaux aléatoires sont caractérisés par leurs propriétés statistiques (alors que les signaux d'excitation utilisés précédemment étaient de nature déterministe). [8]

#### ♦ Autocorrélation

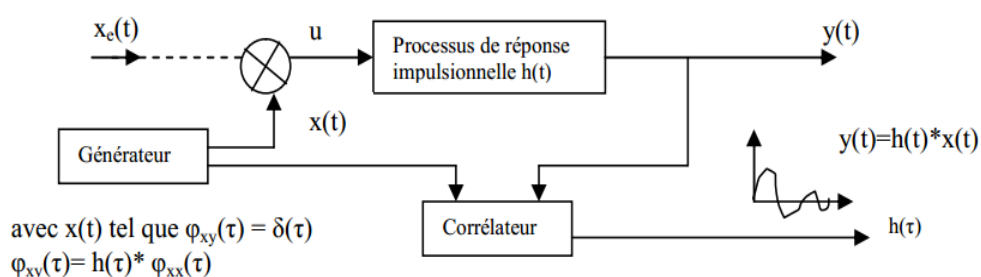
Si on compare dans l'intervalle défini, un signal  $x(t)$  avec lui-même décalé de  $\tau$ , on conçoit que cela permette de voir en quoi le signal à une instant donnée est lié à ce qui s'est passé un instant  $\tau$  avant.

$$\varphi_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0 - \frac{T}{2}}^{t_0 + \frac{T}{2}} x(t)x(t + \tau) dt \quad (\text{II.1})$$

#### ♦ Inter-corrélation

Si on compare deux signaux  $x(t)$  et  $y(t)$ , par exemple l'excitation et la réponse d'un système linéaire, on peut se demander, si de la connaissance de l'histoire de  $x(t)$  et de  $y(t)$  pendant le même intervalle de temps  $T$ , on peut déduire une relation entre les deux signaux. Si le signal  $x(t)$  est tel que sa fonction de transfert est assimilable à une impulsion de Dirac, alors  $\varphi_{xx}(\tau) = h(\tau)$ .

$$\varphi_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0 - \frac{T}{2}}^{t_0 + \frac{T}{2}} x(t)y(t + \tau) dt \quad (\text{II.2})$$



**Figure.II.5.** Structure d'identification par réponse impulsionnel

Notons que la transformée de Fourier d'une corrélation temporelle, a pour résultat une corrélation fréquentielle (densité spectrale). La corrélation peut être utilisée, par exemple, pour des études vibratoires (en mécanique). Cette méthode est toutefois sensible aux perturbations.

Les diverses méthodes non paramétriques (analyse fréquentielle, réponse indicielle, impulsionnelle...) peuvent aider à choisir la structure du modèle. Néanmoins, les méthodes qui prennent en compte un modèle paramétrique du processus, sont préférées, car elles sont plus simples à mettre en œuvre et à utiliser. [8]

## II.4.2. Méthodes paramétriques

Il existe deux principales classes de méthodes paramétriques

- Les méthodes paramétriques graphiques ou déterministes
- Les méthodes paramétriques statistiques (études stochastiques)

### II.4.2.1. Les méthodes paramétriques graphiques (déterministes)

Les méthodes graphiques sont très utiles pour la détermination rapide d'un modèle. La plupart des techniques existantes sont basées sur une réponse du procédé soumise à un échelon en entrée. De ce fait, ces méthodes sont souvent simples à mettre en œuvre sur les installations. Elles permettent d'obtenir des réponses sur le comportement de l'asservissement et ainsi de mieux appréhender le problème à résoudre. [8]

L'objectif est d'obtenir des modèles paramétriques à partir de modèles non paramétriques, type « réponse en échelon » ou « réponse en fréquence ».

#### ❖ Méthode de strejc

Un processus de type intégrateur est caractérisé par le fait qu'à une variation en échelon applique à son entre correspond une variation en rampe de sa sortie lors du régime permanent

On trace le droite D2 passant par l'origine parallèle à l'asymptote D1 pour une fonction de transfert de la forme

$$H(p) = \frac{Ke^{-rp}}{p(\tau p + 1)^n} \quad (\text{II.3})$$



On calcule le rapport  $\frac{AB}{AC}$

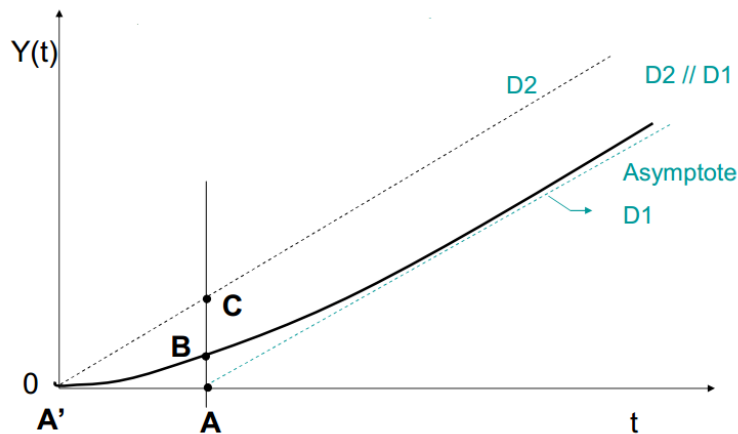


Figure.II.6. Identification par méthode de strejc

- ♦ Si n entier alors  $\tau = \frac{AA'}{n}$  est r est nul
- ♦ Si n n'est pas entier on détermine le nouveau rapport  $\frac{AB}{AC}$  correspondant à la partie entier de n pour cela déplacé D2 parallèlement à D2 par D1 pour obtenir ce nouveau rapport .le temps mort r correspond alors à la translation effectuée par D2

Calculer  $\tau$  à partir de

$$AA' = r + \tau n$$

Le gain dynamique

$$K = \frac{AC}{AA'}$$

N	1	2	3	4	5	6	7	8
$\frac{AB}{AC}$	0.368	0.271	0.224	0.195	0.175	0.161	0.149	0.139

Tab.II.1. Ordre de système par la méthode de strej

❖ **Méthode de Broida**

Le modèle propose est de 1<sup>er</sup> ordre avec un retarder pur soit fonction de transfert

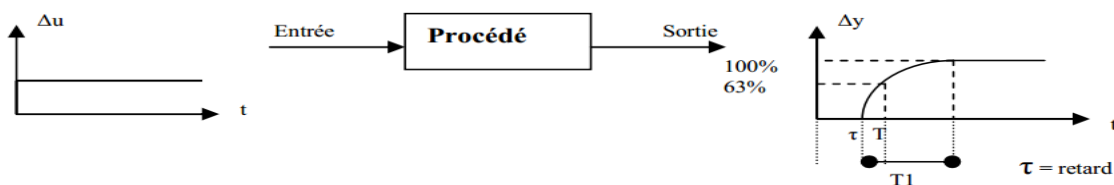
$$G(p) = \frac{K e^{-rp}}{1 + \tau p} \tag{II.4}$$

Soit  $t_1$  et  $t_2$  les temps au soit quelles la réponse expérimental atteint respectivement 28% et 40% de la valeur finale les paramètres du modèle de Broida soit donne comme suit

$$\tau = 5.5(t_2 - t_1)$$

$$r = 2.8t_1 - 1.8t_2$$

Nous illustrons ces méthodes ci-dessous

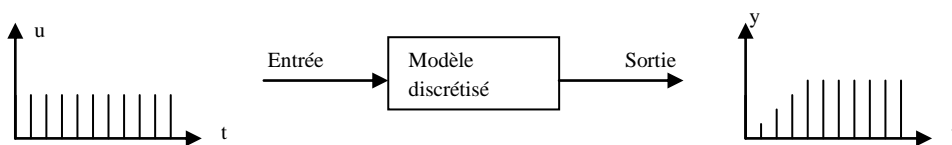


**Figure.II.7.**Méthode d'identification de Broida des modèles continus

Cette méthode peut être étendue à l'identification des modèles échantillonnés

$$H(q^{-1}) = q^{-d} \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1}} \quad (\text{II.5})$$

Discrétisation :



**Figure.II.8.**Méthode d'identification de Broida des modèles échantillonnés

Les méthodes que nous venons de décrire sont simples, et ne nécessitent que peu de matériel, en revanche elles :

- sont limitées dans le choix des modèles,
- nécessitent des signaux tests d'amplitude importante (rarement tolérés par les installations Industrielles),
- sont sensibles aux perturbations (et il n'est pas possible ici de modéliser les perturbations),
- il n'y a pas de procédures explicites de validation des modèles,
- En boucle ouverte, les procédures d'identification que nous venons de décrire, sont assez imprécises,

- La méthode proposée à partir du modèle de Strejc, ou de celui de Broida, permet de déduire l'ensemble des paramètres du modèle. En boucle fermée le mode opératoire utilisé consiste à placer le système à la juste instabilité (déphasage de  $-\pi$  et gain de boucle de  $-1$  : ce sont des valeurs exactes. Si la B.F. comporte un intégrateur, il doit en être tenu compte).

#### II.4.2.2. Les méthodes paramétriques statistiques

Il s'agit de mettre en place des algorithmes d'estimation automatique des paramètres des modèles échantillonnés des procédés en utilisant un ordinateur numérique. Le traitement des données entrées / sorties peut se faire à l'aide d'algorithmes non récursifs (Traitement en bloc des fichiers de données obtenues sur un horizon de temps) ou récursifs (Traitement pas à pas des données), opérant avec des signaux d'excitation extrêmement faibles. Le second type d'algorithme, présente l'avantage d'être « temps réel » et nécessite moins de ressources mémoire en terme de calculs. [8]

L'identification paramétrique statistique est généralement basée sur l'erreur de prédiction, qui est l'erreur entre la sortie du procédé  $y(t)$ , et la sortie  $\hat{y}(t)$  prédite par le modèle. Cette erreur de prédiction est utilisée par un Algorithme d'Adaptation Paramétrique (A.A.P.), qui à chaque instant d'échantillonnage, va modifier les paramètres du modèle afin d'en minimiser l'erreur. L'entrée appliquée au système est généralement une Séquence Binaire Pseudo Aléatoire (S.B.P.A.) de très faible niveau, engendrée par le ordinateur. [8]

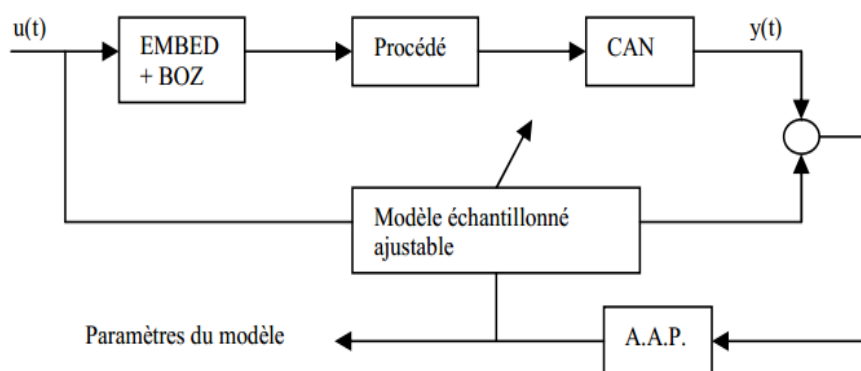


Figure.II.9. Structure d'identification paramétrique statique

Cette approche moderne de l'identification

- élimine les défauts mentionnés précédemment et permet un suivi des paramètres en temps réel,
- permet de modéliser (et supprimer si besoin) les perturbations et bruits capteurs, et permet un traitement aisé des signaux (analyse spectrale des signaux).

## II.5. Les algorithmes de l'identification

### II.5.1. Algorithme du moindre carré simple

La méthode de moindre carré a l'avantage d'une certaine simplicité mais en pratique elle a l'inconvénient de donner bien souvent une estimation biaisée du vecteur des paramètres. Dans ce qui suit on suppose que le bruit du système : bruit interne et bruit de mesure est ramené en sortie suivant l'hypothèse de linéarité (bruit additif) L'erreur entre la sortie réelle et la sortie du modèle  $e(k)$  Et les paramètres  $\theta$  soit alors des variables aléatoires caractérisée par leurs paramètres statistiques (valeur moyenne et la variance) . [10]

Les calculs suivant seront fondés sur le modèle présenté en figure

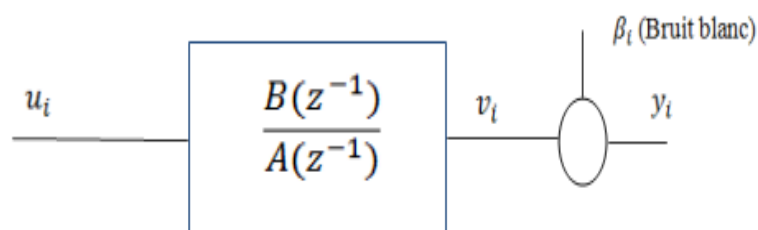


Figure.II.10. Structure de représentation de modèle

La mise en équation du système donne

$$Av_i = Bu_i \quad (\text{II.6})$$

$$y_i = v_i + \beta_i \quad (\text{II.7})$$

$$A(y_i - \beta_i) = Bu_i \quad (\text{II.8})$$

$$Ay_i = Bu_i + A\beta_i \quad (\text{II.9})$$

Posons

$$e_i = A\beta_i \quad (\text{II.10})$$

$e_i$  sont les résidus de l'estimation. On obtient finalement

$$Ay_i = Bu_i + e_i \quad (II.11)$$

Sous forme matricielle, on obtient

$$\begin{pmatrix} y_N \\ y_{N-1} \\ y_{N-2} \\ y_{N-3} \\ \vdots \\ \vdots \\ y_{n+2} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} -y_{N-1} & -y_{N-2} & \dots & -y_{N-n} & u_N & u_{N-p} \\ -y_{N-2} & -y_{N-3} & \dots & -y_{N-n-1} & u_{N-1} & u_{N-p-1} \\ -y_{N-3} & \vdots & \dots & \vdots & u_{N-2} & u_{N-p-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots u_{N-p-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & -y_{n+1} & \dots & \vdots & \vdots & \vdots \\ -y_{n+1} & -y_n & \dots & \vdots & \vdots & \vdots \\ -y_n & -y_{n-1} & \dots & -y_1 & u_{n+1} & u_{n-p} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_m \end{pmatrix} + \begin{pmatrix} e_N \\ e_{N-1} \\ e_{N-2} \\ \vdots \\ \vdots \\ e_{n+2} \\ e_{n+1} \end{pmatrix} \quad (II.12)$$

Soit

$$Y = \phi\theta + e \quad (II.13)$$

Le critère J est

$$J = \sum e^2 = e^T e \quad (II.14)$$

D'où on en déduit :

$$\hat{\theta} = (\phi^T \phi)^{-1} \phi^T Y \quad (II.15)$$

➤ **Calcul du biais de l'estimateur**

$$Y = \theta\phi + e \quad (II.11)$$

Et

$$\begin{aligned} E[\hat{\theta}] &= E[(\phi^T \phi)^{-1} \phi^T Y] \\ &= E[\theta + (\phi^T \phi)^{-1} \phi^T e] \\ &= \theta + E[(\phi^T \phi)^{-1} \phi^T e] \end{aligned} \quad (II.16)$$

L'estimateur non biais si :  $E[\hat{\theta}] = \theta$       Donc si  $E[(\phi^T \phi)^{-1} \phi^T e] = 0$

Il faut donc que :

- $\phi$  et  $e$  soient non corrélée
- $e$  soit centre

### II.5.2. Méthode de la Matrice Instrumentale

Puisque le biais de l'estimateur des moindres carrés est biaisé à cause de la corrélation entre  $\phi$  et  $e$ , on se propose de déterminer une autre matrice qui permette le calcul de  $\theta$  tout en évitant cette corrélation et donc le biais de l'estimateur. [10]

Posons

$$\hat{\theta} = (Z^T \phi)^{-1} Z^T y \quad (\text{II.17})$$

Où  $Z$  est la matrice instrumentale.

En poursuivant le calcul

$$\hat{\theta} = (Z^T \phi)^{-1} Z^T (\phi \theta + e) \quad (\text{II.18})$$

$$\hat{\theta} = \theta + (Z^T \phi)^{-1} Z^T e \quad (\text{II.19})$$

Pour que  $E[\hat{\theta}] = \theta$

Il faut :  $E[Z^T \phi] \neq 0$  et  $E[Z^T e] = 0$

Plusieurs choix de  $Z$  sont possibles, en voici deux : le premier choix consiste à faire deux campagnes de mesures avec la même entrée  $u_i$ , le plus souvent consécutives (décalées dans le temps). Nous avons alors la possibilité de déterminer deux matrices  $\phi$  soit  $\phi_1$  et  $\phi_2$ .

En posant

$$\hat{\theta} = (\phi_2^T \phi_1)^{-1} \phi_2^T y_2 \quad (\text{II.20})$$

Il n'y a plus de corrélation entre  $\phi_2$  et  $y_1$  (le bruit est stochastique et ergodique) le biais est bien nul.

La deuxième proposition consiste à créer les données nécessaires à la création de la matrice  $\phi_2$  à l'aide d'un modèle de type moindre carré simples issu de la première campagne de mesure.

### II.5.3. Méthode de moindre carré récursive

L'estimation de paramètres par la méthode des moindres carrés simples présente un inconvénient majeur, la nécessité de calculer l'inverse d'une matrice, ce qui est long et parfois impossible sur un microcontrôleur. On se propose de déterminer une forme récursive de cette estimation. [10]

Les avantages d'une formulation récursive tiennent essentiellement en deux points.

- La possibilité de traiter un plus grand nombre de données que dans le cas de la formulation directe (pas de pseudo-inverse à calculer), notamment dans le cas de l'implantation sur un microcontrôleur.
- Dans le cas des systèmes variant dans le temps, la forme récursive permet de "suivre" les paramètres du système. Dans ce cas, l'identification en ligne est généralement suivie d'une commande qui elle aussi "s'adapte" aux paramètres courants, on entre alors dans le domaine des commandes auto adaptatives.

Comme dans tout problème récursif on s'intéresse d'abord à la boucle, ensuite comment on en sort et enfin comment on y entre. Supposons que nous possédions une estimation des paramètres à l'instant N :  $\theta_N$

$$\theta_N = (\phi_N^T \phi_N)^{-1} \phi_N^T Y_N \quad (\text{II.21})$$

A l'instant suivant la nouvelle estimation est

$$\theta_{N+1} = (\phi_{N+1}^T \phi_{N+1})^{-1} \phi_{N+1}^T Y_{N+1} \quad (\text{II.22})$$

$$\text{Avec } \phi_{N+1} = \begin{bmatrix} \phi_{N+1} \\ \phi_N \end{bmatrix}, Y_{N+1} = \begin{bmatrix} y_{N+1} \\ Y_N \end{bmatrix} \quad \text{et } E_{N+1} = \begin{bmatrix} e_{N+1} \\ E_N \end{bmatrix}$$

$$\phi_{N+1} = [-y_N - y_{N+1} \dots - y_{N-n_a+1} \quad u_{N+1} + u_{N+2} \dots u_{N-n_b+1}] \quad (\text{II.23})$$

$\theta_{N+1}$  S'écrit alors :

$$\begin{aligned} \hat{\theta}_{N+1} &= (\phi_{N+1}^T \phi_{N+1})^{-1} \phi_{N+1}^T Y_{N+1} \\ \hat{\theta}_{N+1} &= (\phi_N^T \phi_N + \phi_{N+1}^T \phi_{N+1})^{-1} (\phi_N^T Y_N + \phi_{N+1}^T Y_{N+1}) \end{aligned} \quad (\text{II.24})$$

En utilisant le lemme d'inversion matricielle

$$(Q + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (\text{II.25})$$

On obtient

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1}(y_{N+1} - \phi_{N+1}^T \hat{\theta}_N) \quad (\text{II.26})$$

$$K_{N+1} = P_N \phi_{N+1}^T (I + \phi_{N+1}^T P_N \phi_{N+1})^{-1} \quad (\text{II.27})$$

$$P_{N+1} = P_N - K_{N+1} \phi_{N+1}^T P_N \quad (\text{II.28})$$

Cette méthode présente les même avantage que la méthode moindre carre non réursive, mais aussi le même inconvénient l'estimateur est biaise .Aussi on lui préfère généralement d'autres méthodes telles que la méthode de moindre carré étendu réursive.

#### II.5.4. Méthode de moindre carré étendu réursive [11]

Cet algorithme est utilisé dans l'identification de modèle de type ARMAX de la forme (modèle « procédé+perturbation »)

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})e(t) \quad (\text{II.29})$$

Cette identification a pour objectif de modéliser le procédé ainsi que la perturbation afin d'obtenir une erreur de prédiction nulle. L'erreur de prédiction est telle que

$$\varepsilon(t+1) = e(t+1) = y(t+1) - \hat{y}(t+1) \quad (\text{II.30})$$

On minimise le critère au sens des « moindres carrés » avec maintenant le vecteur des Paramètres estimés et le vecteur des observations

$$\varphi_{N+1} = \begin{bmatrix} -y_N - y_{N+1} \dots - y_{N-n_a+1} & u_{N+1} + u_{N+2} \dots u_{N-n_b+1} & \varepsilon_N \\ + \varepsilon_{N+1} \dots \dots \varepsilon_{N-n_c+1} \end{bmatrix} \quad (\text{II.31})$$

$$\theta_N^T = \begin{bmatrix} -a_N - a_{N+1} \dots - a_{N-n_a+1} & b_{N+1} + b_{N+2} \dots b_{N-n_b+1} & c_N \\ + c_{N+1} \dots \dots c_{N-n_c+1} \end{bmatrix} \quad (\text{II.32})$$

Dans cette méthode on remarque que La taille de  $\varphi_{N+1}$  et  $\theta_N$  augmente par rapport au moindres carrés, cette méthode donne une estimation du paramètres du modèle non biaisé.

#### II.6. Validation du modèle

La procédure d'identification des paramètres choisit le « meilleur » modèle compte tenu des données disponibles, de la structure de modèle proposée, du critère de performance et de l'algorithme d'identification utilisés [9]. Ce « meilleur » modèle est-il suffisamment bon? De manière plus spécifique:

- est-il suffisamment en accord avec les données expérimentales?
- est-il suffisant pour le but escompte?
- d'écrit-il le «vrai» système?



La validation du modèle peut s'effectuer grâce à diverses approches complémentaires :

- La première est de soumettre le modèle identifié à l'entrée des mesures pour visualiser la sortie et la comparer à la mesure réelle (simulation graphique par exemple). Par ce principe simple, on peut d'un seul coup d'œil dire si le modèle est faux. A contrario, il n'est pas évident d'affirmer par cette simple étape de la pertinence du modèle.
- La deuxième approche est de soumettre le modèle et le système réel (dans les conditions nominales) aux mêmes excitations afin de visualiser si l'identification était suffisamment précise pour décrire le comportement dynamique complet du système réel. Cette étape n'est hélas pas toujours aisée à réaliser.
- La troisième approche est de vérifier la validité du modèle par des tests sur les mesures. L'un de ceux-ci est basé sur le blanchissement de l'erreur de prédiction (méthode utilisées pour les MCR, MCE, MVR). On effectue le calcul tel que :

$$RN(i) = \frac{R(i)}{R(0)} = \frac{\frac{1}{N} \sum_t \varepsilon(t) \varepsilon(t-1)}{\frac{1}{N} \sum_t \varepsilon(t)^2} \quad (\text{II.33})$$

Avec  $i=1,2,\dots,\max(n_a, n_b + d)$

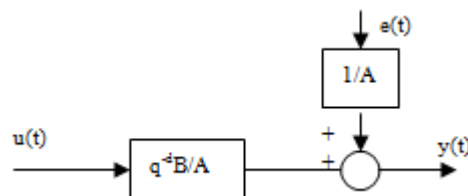
En général un modèle est acceptable du point de vue du test si :

$$|RN(i)| < \frac{2.17}{\sqrt{N}}$$

## II.7. Quelques structures « procédé + perturbation » largement utilisées en identification [11]

**ARX** (Auto régressive à variable exogène)

$$A(q^{-1})y(t) = q^{-d}B(q^{-d})u(t) + e(t) \quad (\text{II.34})$$



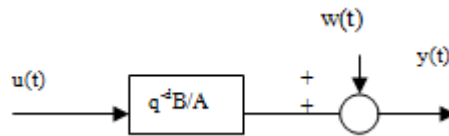
**Figure.II.11.** Structure de modèle ARX

$e(t)$  est un bruit blanc.

On peut utiliser la méthode des moindres carrés récurrents avec cette structure.

**Erreur de sortie**

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + A(q^{-1})w(t) \tag{II.35}$$



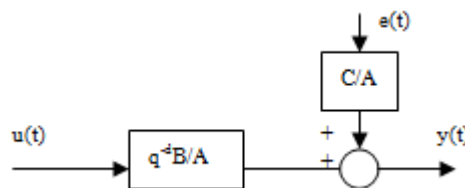
**Figure.II.12.** Structure de modèle Erreur de sortie

w(t) est une perturbation non modélisée.

Méthodes utilisées avec cette structure : Moindres carrés récursifs, variable instrumentale, erreur de sortie.

**ARMAX** (Auto régressive à moyenne ajustée et variable exogène)

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})e(t) \tag{II.36}$$

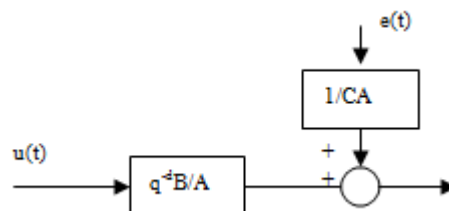


**Figure.II.13.** Structure de modèle ARMAX

Méthodes utilisées avec cette structure : moindres carrés étendus, maximum de vraisemblance récursif, erreur de sortie avec modèle de prédiction étendue.

**ARARX :**

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + \frac{1}{C(q^{-1})}e(t) \tag{II.37}$$



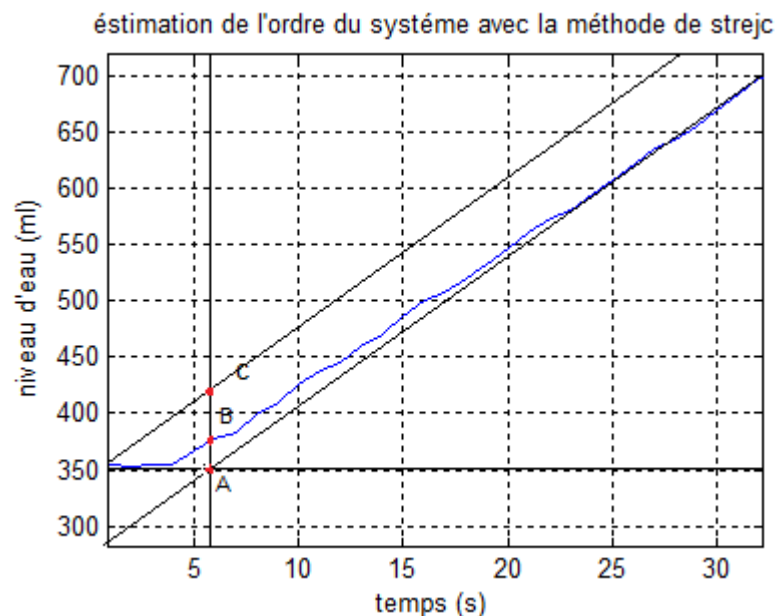
**Figure.II.14.** Structure de modèle ARARX

Méthodes utilisés avec cette structure : Moindres carrés généralisés.

## II.8. Estimation de l'ordre du système

Un des objectifs de l'identification est d'obtenir le modèle le plus simple qui vérifie les critères de validation. Une première approche pour estimer l'ordre du système, c'est-à-dire les valeurs  $m$ ,  $n$  et  $d$  qui sont respectivement l'ordre des polynômes  $B(q^{-1})$ ,  $A(q^{-1})$  et le retard, est d'étudier les représentations non paramétriques du système. [9]

Une estimation grossière du retard et de l'ordre du dénominateur peut être obtenue de la réponse indicielle ou impulsionnelle. Pour estimer l'ordre de notre système nous utilisons la méthode de Strejc, nous excitions le système avec un échelon puis nous traçons leur réponse indicielle. Nous avons obtenu le résultat suivant



**Figure.II.15.** Estimation de l'ordre du système avec la méthode de strejc

D'après la méthode de strejc

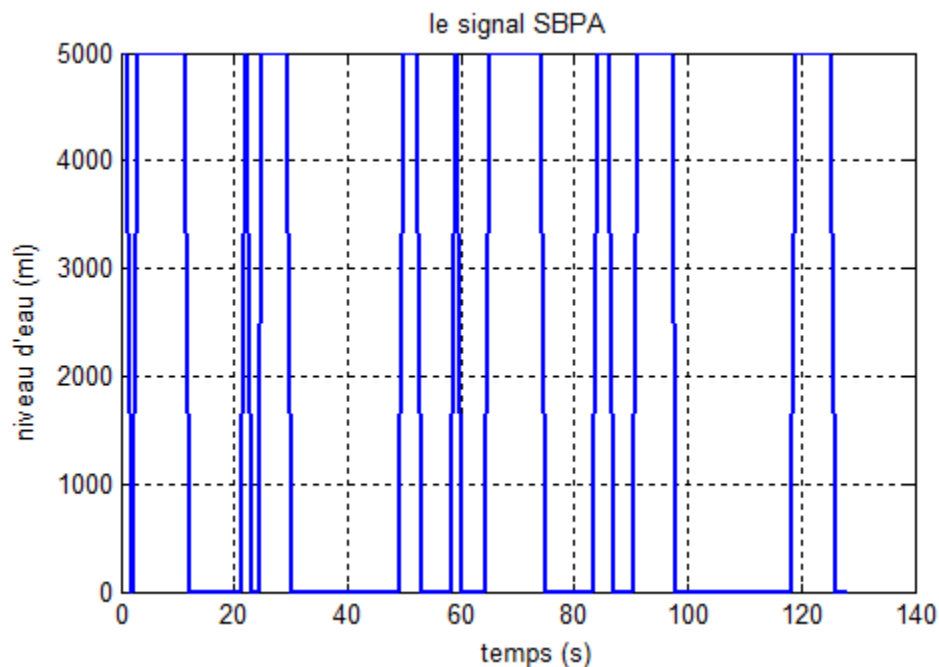
$$\frac{AB}{AC} = \frac{375-353}{412-353} = \frac{22}{60} = 0.367. \Rightarrow n = 1.$$

Le système étudié est un intégrateur du premier ordre sans retard pure.

## II.9. Simulation MATLAB des algorithmes d'identification récursif

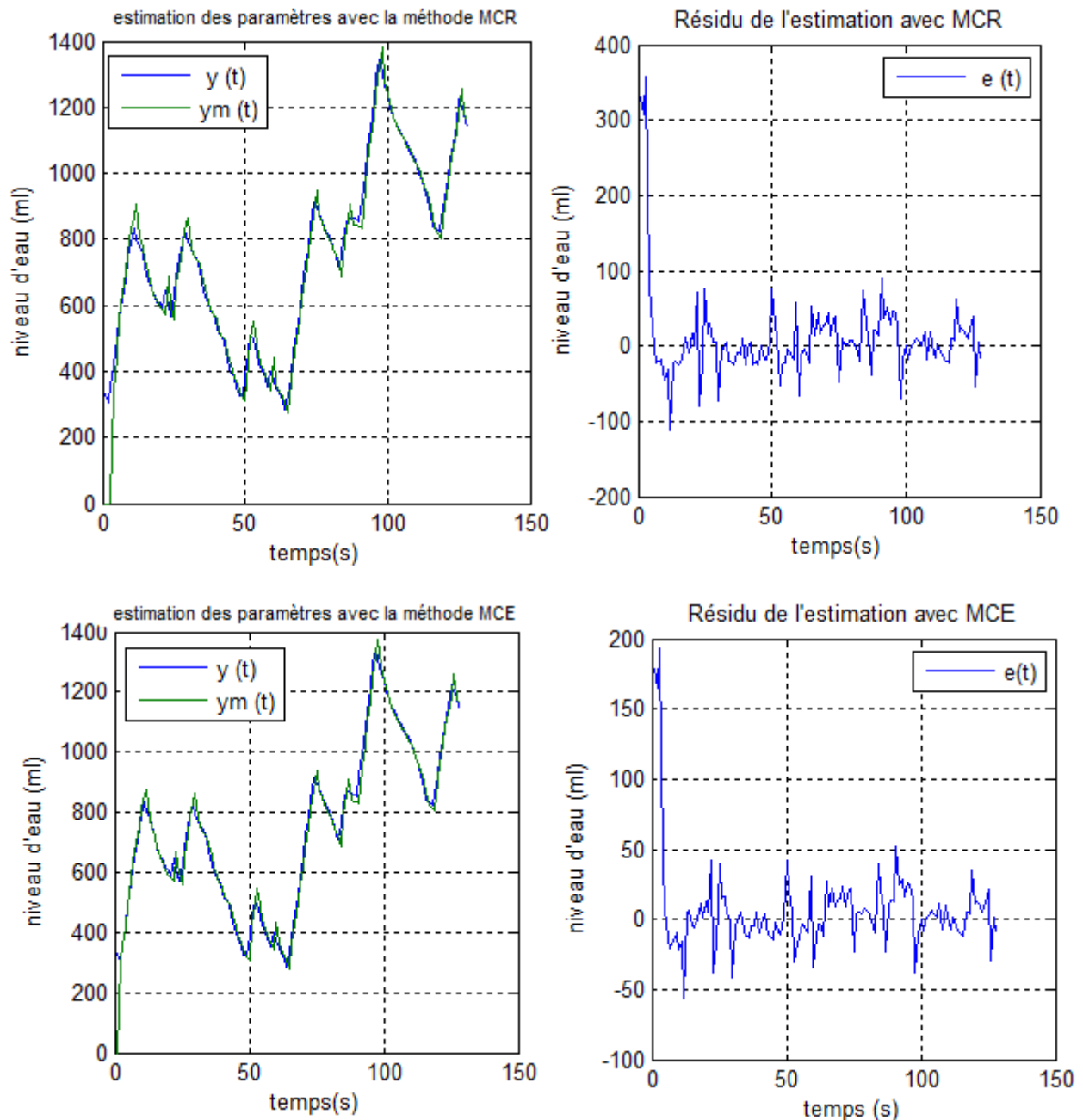
Pour identifier le système hydraulique que sur lequel notre travail a été fait, nous utilisons deux méthodes d'identification, la première méthode est la méthode des moindres carrés récursifs (MCR), et à cause du biais de l'estimation des paramètres du modèle obtenu par cette méthode, il faut utiliser une deuxième méthode, c'est la méthode du moindres carrés étendu (MCE), avec cette méthode on obtient une estimation non biaisée.

L'identification nécessite un signal d'excitation riche en fréquence, pour cela nous excitons le système avec un signal SBPA de longueur 7, pour cet effet nous avons développé un programme en langage ST sous PL7, puis nous avons implanté ce programme sous l'automate Schneider.



**Figure.II.16.** Le signal d'excitation SBPA

Pour faire une simulation MATLAB des deux algorithmes d'identification, nous excitons le système avec le signal SBPA, puis nous enregistrons les couples des mesures (entrée, sortie) avec une base de temps de 1s, et à partir de l'option « transfert des données » intégrée dans le logiciel PL7, nous traçons les courbes du modèle obtenu et la sortie du système réelle sous MATLAB, nous avons trouvé les résultats suivants



**Figure.II.17.** Résultat de l'identification de système avec MATLAB

Nous remarquons que les résultats obtenus avec les deux méthodes sont proche de la sortie du système réel, mais cette vision graphique seule n'est pas suffisante pour dire que les modèles obtenus sont acceptables, il est nécessaire de faire le test de blancheur sur le résidu de l'estimation, pour valider les modèles nous développons un programme MATLAB pour faire ce test. Nous trouvons les résultats suivants

	MCR	MCE
N	128	128
R(0)	3548.3	1705.2
R(1)	2025.6	206.39
RN(1)	0.5708	0.1210

**Tab.II.2.** Résultats du test de blancheur avec MATLAB

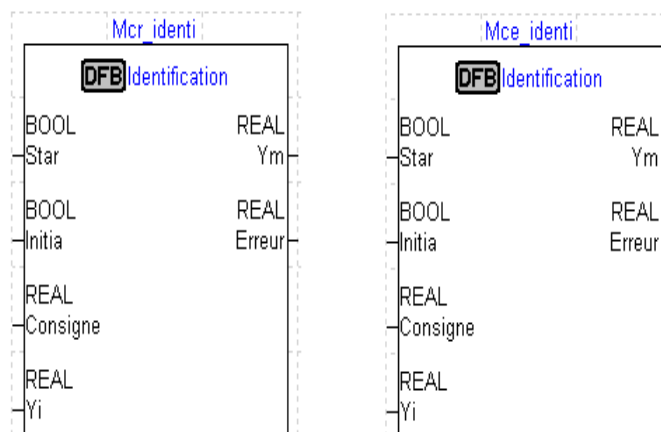
Le test de blancheur de l'erreur de prédiction que nous avons appliqué met en évidence que la méthode de MCR n'est pas acceptable, il est clair qu'elle est biaisée, par contre la méthode de MCE est acceptable car le résultat de leur test est en dessous de la limite théorique (0.197).

## II.10. Implantation des algorithmes d'identification sous l'automate programmable Schneider premium

Notre objectif dans ce chapitre est de faire une identification en ligne du système étudié, dans cette partie nous utilisons le logiciel PL7 pour développer des objets DFB (dérive factions blocks) sous l'automate, nous développons trois objets. Deux pour les algorithmes d'identification, et le troisième objet pour la validation du modèle obtenu.

### II.10.1. L'identification paramétrique du système

L'identification paramétrique est assurée grâce à deux objets DFB que nous avons développés pour le projet : le « MCR » (Moindres Carrés Récurifs), le « MCE » (Moindres carrés Etendus).



**Figure.II.18.** Les blocs DFB des deux algorithmes d'identification

### II.10.2. Le signal d'excitation

Pour exciter le système à identifier nous utilisons un signal SBPA. L'objet « SBPA » que nous avons implanté dans une DFB.

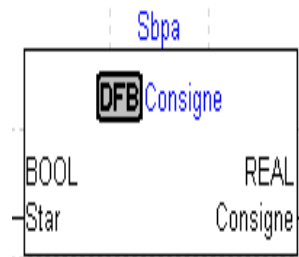


Figure.II.19. Bloc DFB de signal SBPA

### II.10.3. La validation du modèle

La validation du modèle identifié est assurée par un objet DFB que nous avons développé à cet effet, nous avons appelé cet objet « Rn\_i », il réalise le test de blancheur pour les deux méthodes d'identification développées pour le projet.

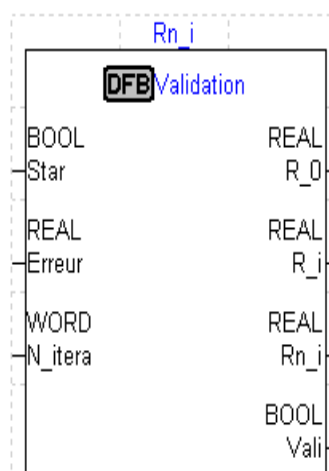
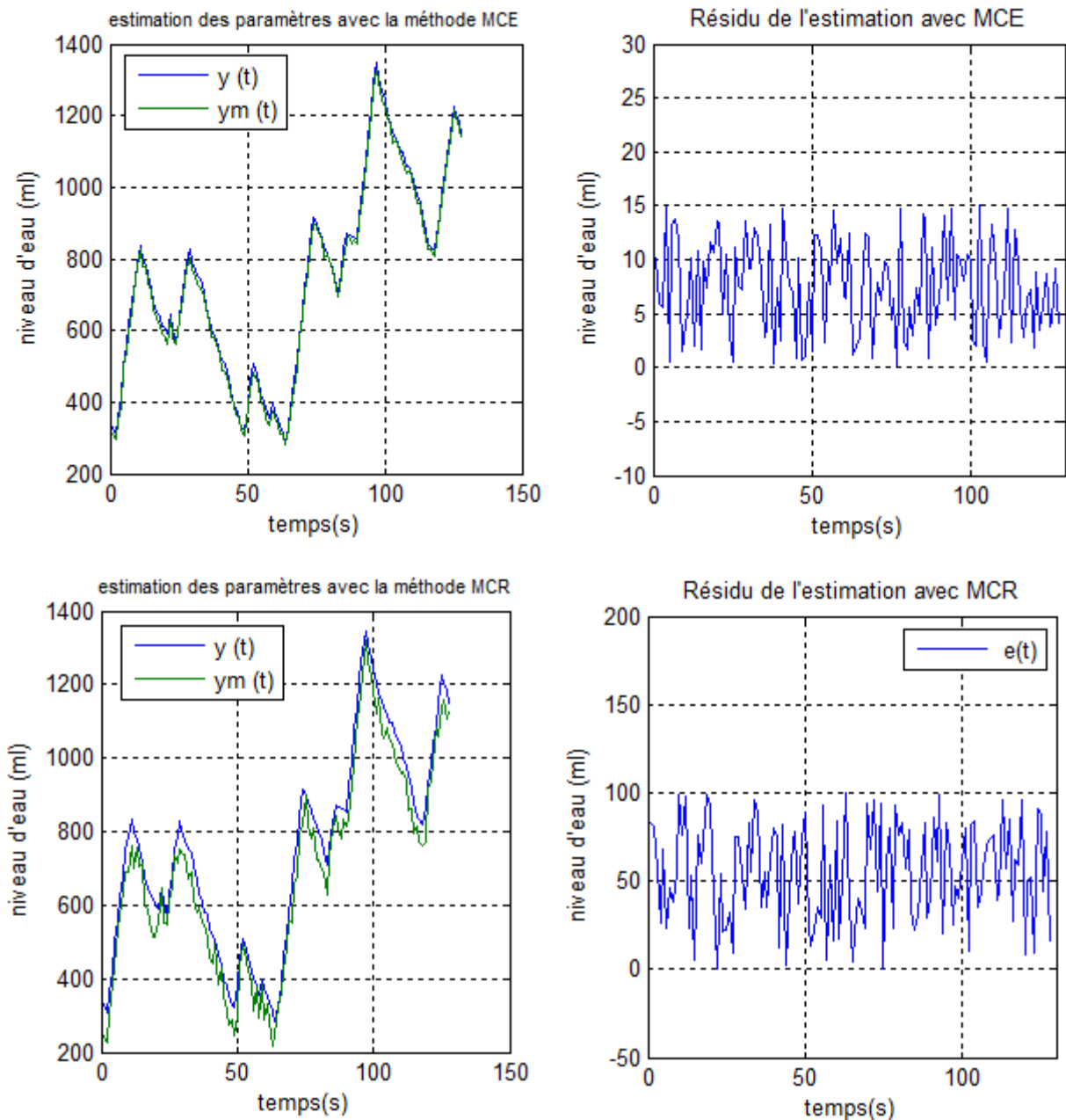


Figure.II.20. Bloc DFB de validation du modèle

Après l'implémentation de ces objets sous l'automate à partir le logiciel PL7, nous avons mis en marche le système puis nous enregistrons les résultats, on obtient les graphes suivants



**Figure.II.21.** Résultats de l'identification des deux algorithmes avec l'automate

Les graphes obtenus montrent le biais de la méthode MCR, il y a une divergence entre la sortie du modèle et la sortie réelle du système, le biais de l'estimateur est éliminé dans la méthode MCE, nous avons trouvé qu'il y a une convergence entre les deux graphes, ces résultats sont confirmés avec les résultats du test de blancheur qui est représenté dans le tableau suivant



	MCR	MCE
N	128	128
R(0)	2875.14	1579.65
R(1)	2425.18	134.5
RN(1)	0.8434	0.085

**Tab.II.14.** Résultats du test de blancheur avec automate

Après l'identification du système avec les deux méthodes, nous avons trouvé les modèles suivant :

$$y_{MCR\_PL7}(t) = \left( \frac{0.0137q^{-1}}{1 - 1.0001q^{-1}} \right) u(t) \quad (\text{II.38})$$

$$y_{MCE\_PL7}(t) = \left( \frac{0.0365q^{-1}}{1 - 1.00007q^{-1}} \right) u(t) + \left( \frac{0.0041q^{-1}}{1 - 1.00007q^{-1}} \right) e(t) \quad (\text{II.39})$$

## II.11. Conclusion

Avec le développement des techniques de la commande des systèmes dynamique, il est nécessaire de trouver un modèle mathématique qui représente la dynamique du système à commander. Ce modèle est utilisé soit pour faire une simulation sur le système, soit pour l'utiliser dans la commande du système lui-même, si le système est difficile à modéliser à partir de leur lois physiques, il y'a une deuxième méthode pour obtenir le modèle mathématique, cette méthode est l'identification des systèmes.

Dans ce chapitre nous avons fait l'identification en ligne du système hydraulique étudié, nous avons utilisé la méthode des moindres carrés récursif pour estimer les paramètres du modèle, et à cause du biais de cette estimation, nous abordons une deuxième méthode de l'identification, c'est la méthode de moindre carré étendu récursif, nous avons simulé les deux méthodes avec le logiciel MATLAB, puis nous avons implanté ces algorithmes sous l'automate programmable industriel Schneider.

Enfin on a trouvé que le biais de l'estimation est bien éliminé avec la méthode du MCE, et que les résultats de la simulation Matlab et de l'automate sont très proches.

**Chapitre III**  
**Régulation du système**  
**hydraulique**

### III.1. Introduction

Dans le domaine de l'industrie, la régulation des grandeurs physiques est nécessaire pour une bonne démarche des systèmes.

Dans ce chapitre nous avons fait une régulation de niveau pour les deux cuves placées dans le système étudié, pour cet effet nous utilisons deux régulateurs numériques, premièrement nous allons faire une simulation MATLAB de ces régulateurs, puis nous implémentons l'algorithme du régulateur PID sous l'automate programmable industrielle Schneider premium, enfin nous traçons les graphes de la consigne et la sortie du système réelle après réglage, nous avons trouvé que le système est bien réglé, et que les régulateurs sont acceptable ils sont réponde à notre besoin .

### III.2. Notions de système

Un système est une modélisation d'un procédé en fonctionnement. Il possède une ou plusieurs entrées, et une ou plusieurs sorties. Les entrées du système sont appelées variables exogènes; elles rassemblent les perturbations et les variables manipulées, commandes ou grandeurs de réglage. Elles sont souvent représentées de manière générique par la lettre  $u$  ou  $e$ . Elles sont reliées au procédé en tant que tel par un actionneur. [12]

Les sorties du système sont appelées variables contrôlées, mesures ou grandeurs réglées. Elles sont souvent représentées de manière générique par la lettre  $y$ . Le procédé est relié à la sortie du système par un capteur. [12]

### III.3. Classifications des systèmes

Les systèmes peuvent être classés en plusieurs catégories.

#### III.3.1. Classifications selon la nature de système

##### Systèmes à temps continus

Ce sont les systèmes qui existent naturellement. Pour ces systèmes, le temps  $t$  décrit la droite réelle

##### Systèmes à temps discret

Ce sont des systèmes pour lequel le temps  $k$  est une variable discrète (on se ramène généralement au cas où  $k$  décrit l'ensemble des nombres entiers). Sauf exception, ces

systèmes n'existent pas à l'état naturel (la majorité des systèmes physiques naturels sont à temps continu), mais étant donné que la plupart des contrôleurs utilisés en automatique sont calculés par des processeurs numériques, il est parfois intéressant de modéliser le système commandé comme un système à temps discret.

#### **Systèmes à événements discrets**

Systèmes dont le fonctionnement peut être modélisé par des événements discrets. Généralement, ces systèmes sont modélisés par des réseaux de Pétri, des GRAFCET (qui en sont des cas particuliers très répandus, notamment dans l'industrie) ou par les algèbres de diodes. Des exemples sont les réseaux ferroviaires, ou le fonctionnement d'une chaîne de montage.

#### **Systèmes hybrides**

Systèmes dont la modélisation nécessite l'utilisation des techniques liées aux systèmes continus et aux systèmes à événements discrets, par exemple : une boîte de vitesses de voiture.

### **III.3.2. Classifications selon les entrées et les sorties**

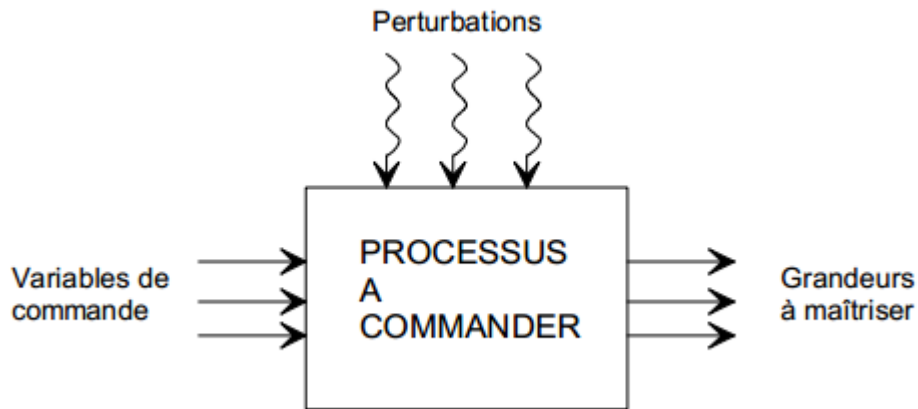
Quatre possibilités existent

- Le système à une entrée et une sortie, c'est un système monovariable ou SISO (Single Input Single Output),
- Le système a plusieurs entrées et plusieurs sorties, c'est un système multivariable ou MIMO (Multiple Input Multiple Output),
- Le système a une entrée et plusieurs sorties, système SIMO,
- le système a plusieurs entrées et une sortie, système MISO.

Néanmoins, ces deux derniers termes sont peu utilisés.

### **III.4. La commande des systèmes**

L'objectif principal d'une commande est de maîtriser l'évolution d'une ou plusieurs grandeurs physiques (température, pression, vitesse, PH ...) à partir d'une ou plusieurs variables de contrôle et ceci dans un environnement perturbé. [12]



**Figure.III.1.**Structure de la commande des systèmes

Nous nous trouvons dans la situation où les sorties dépendent des variables de contrôle et des perturbations, ces dernières sont de nature externe ou interne au système et généralement non mesurables.

### **III.5. La régulation des systèmes**

La régulation (ou asservissement) consiste à agir de façon à ce que une mesure soit égale à une consigne. Si l'on cherche à atteindre une consigne (de position ou de température), on parlera de poursuite ou asservissement ; si l'on cherche à éliminer des perturbations pour qu'une valeur reste constante (ex : garder la température intérieure de la voiture constante quelle que soit la température extérieure), on parlera de régulation. [13]

L'industrie utilise à foison des systèmes d'asservissement ou de régulation : que ce soit pour gérer le débit d'un fluide dans une conduite, la température d'un produit, la hauteur d'un niveau de cuve..., historiquement, les régulateurs n'étaient pas intégrés dans l'unité centrale des Automates Programmables Industriels, mais se présentaient sous forme de modules autonomes gérant leur environnement propre (acquisition, calcul, commande...).[13]

De plus en plus, les automates intègrent les régulateurs au sein de l'unité centrale. Soit sous la forme de module autonome émulant un régulateur externe au sein de l'UC (évitant ainsi la redondance de câblage qu'imposait l'utilisation de régulateur externe), soit sous la forme de blocs primitifs intégrables au sein du code au même titre qu'un bloc temporisateur.

### III.6. La régulation numérique des systèmes

L'implantation d'algorithmes de commande sur ordinateur, comparée à une réalisation purement analogique, offre de nombreux atouts, un avantage décisif réside dans une souplesse d'emploi exceptionnelle, il est très facile de modifier les paramètres, voire la structure des algorithmes de commande et de conduite, elles sont des méthodes sophistiquées, exigeantes quant au volume de calcul et de protection, en guise d'exemples les régulateurs polynomiaux à deux degrés de liberté. [14]

Elles aussi offerte une précision élevée résulte de l'élimination des dérives dues à des variations de températures et au vieillissement.

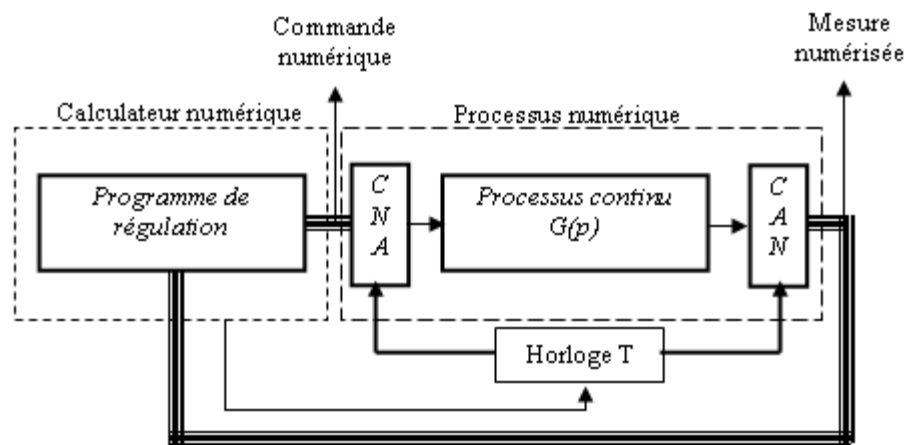


Figure.III.2. Structure d'une régulation numérique

Les régulateurs dans cette commande sont des algorithmes développés par un langage informatique concerné le calculateur utilisée, ils sont implémentés sous ces calculateurs et utilisés pour la commande des systèmes.

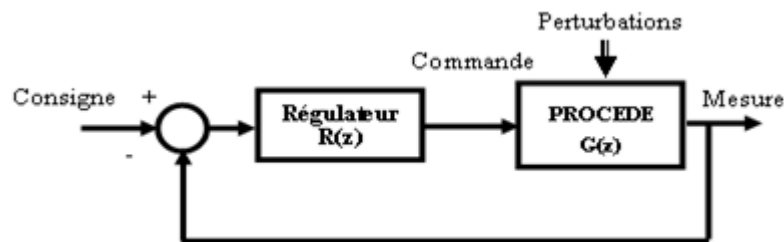
#### III.6.1. Synthèse des régulateurs numériques

##### III.6.1.1. Méthode directe de Synthèse des régulateurs numériques

Les méthodes directes (théoriques) sont très nombreuses et reposent sur la connaissance générale sur la connaissance d'un modèle discret précis du système à commander. Les performances réelles obtenues dépendent de la qualité du modèle et de son aptitude à représenter le mieux possible le procédé. [15]

Pour obtenir ce modèle, on peut partir du modèle continu qu'on discrétise en calculant ou  $G(p)$  désigne la fonction de transfert continue du procédé .l'autre méthode pour obtenir  $G(z)$  est de faire une identification directe assistée par calculateur numérique.

Parmi les méthodes directes, on propose de présenter la méthode du modèle .elle est basée sur la donnée d'un modèle en boucle fermée à atteindre répondant à un cahier des charges.



**Figure.III.3.**Structure de régulation en boucle fermé

La fonction de transfert en boucle ouverte est

$$T(z) = R(z)G(z) \quad (\text{III.1})$$

La fonction de transfert en boucle fermée est

$$F(z) = \frac{R(z)G(z)}{1 + R(z)G(z)} \quad (\text{III.2})$$

### Principe

Si la fonction de transfert en boucle fermée  $F(z)$  est donnée, c'est-à-dire qu'elle a été élaborée de manière à répondre au cahier des charges ,le régulateur  $R(z)$  est déterminé tout simplement par la relation suivante

$$R(z) = \frac{F(z)}{G(z)(1 - F(z))} \quad (\text{III.3})$$

Usuellement le comportement souhaité en boucle fermée est celui d'un système d'ordre un ou d'ordre deux avec un gain statique unitaire, ce qui permet d'assurer une précision statique parfait.

Une fois que la fonction de transfert en boucle fermée désirée est établie, on détermine l'expression du régulateur  $R(z)$  par la formule ci-dessus à partir de laquelle on déduit la loi de commande à programmer.

On note que l'expression de  $R(z)$  repose sur la connaissance de  $G(z)$  et on conclue à présent que le succès de cette méthode est conditionné par la précision avec le modèle  $G(z)$  a été obtenue.

### III.6.1.2. Méthode par transposition des régulateurs continus

Une possibilité de faire la synthèse d'un régulateur numérique est de concevoir d'abord un régulateur analogique satisfaisant dans le domaine continu et d'effectuer ensuite une transformation de la variable  $p$  en variable  $z$ . [15]

### III.6.2. Régulateur PID analogique

Les régulateurs proportionnel intégral (PI) et proportionnel intégral dérivé (PID), sont très répandus, car historiquement, ils faisaient appel pour leur réalisation à des techniques analogiques. A l'heure actuelle, bien que l'approche numérique soit prédominante, l'utilisation des PI, PID perdure, car elle est robuste et ne présuppose pas une connaissance précise de la dynamique du procédé à commander. [16]

Un régulateur Proportionnel Intégral Dérivé réalise la fonction

Dont la transformée de Laplace vaut

$$u(t) = K_p \left( \varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(t) \cdot dt + T_d \frac{d \cdot \varepsilon(t)}{dt} \right) \quad (\text{III.4})$$

$$\frac{u(p)}{\varepsilon(p)} = K_p \left( 1 + \frac{1}{T_i \cdot p} + T_d \cdot p \right) \quad (\text{III.5})$$

Le régulateur PID possède 2 zéros et un pôle nul, il peut compenser deux pôles du processus il sera utilisé préférentiellement pour réguler des processus modélisés par un second ordre.



### III.6.3. Les régulateurs PID discrets

#### III.6.3.1. Généralités

Le régulateur discret élabore une grandeur de commande discrète  $y^*(t)$  en Fonction de l'écart de réglage discret  $Y^*(t)$  du système à commander. Selon la Complexité du régulateur, la grandeur de commande à l'instant  $t=nT$  est formée en Fonction de la valeur de l'écart à cet instant, mais aussi aux instants précédents  $T(n-1)$ ,  $T(n-2)$ , ...

Comme dans le cas des régulateurs continus qui réalisent généralement la relation

$$y(t) = k_p \varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(t) dt + T_d \frac{d\varepsilon(t)}{dt} \quad (\text{III.6})$$

On crée des régulateurs discrets standards qui traduisent en valeurs discrètes cette expression. Les régulateurs les plus courants sont du type: P.I, P.D, P.I.D. Si l'action Proportionnelle P est des plus simples, les actions intégrales, dérivée méritent une certaine attention.

#### III.6.3.2. LES DIFFERENTES ACTIONS

##### ❖ Action Proportionnelle

Cette action est la plus classique. Elle consiste à appliquer une correction en rapport avec la différence instantanée entre la mesure et la consigne courante, Le rapport Erreur/Commande s'appelle le gain. [13]

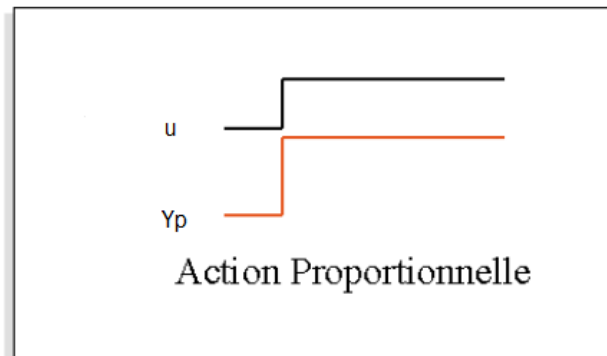
La limite de cette commande arrive à partir du moment où l'erreur devient petite ou si la consigne n'est pas constante ; en effet l'annulation de l'écart considère que la consigne à l'instant  $T+1$  sera le même qu'à l'instant  $T$ .

On a donc l'équation

$$u = K_p(uc - y) \quad (\text{III.7})$$

Dans le cas d'un régulateur inverse, l'équation sera

$$u = -K_p(uc - y) \quad (\text{III.8})$$



**Figure.III.4.** Réponse indiciel d'un régulateur P

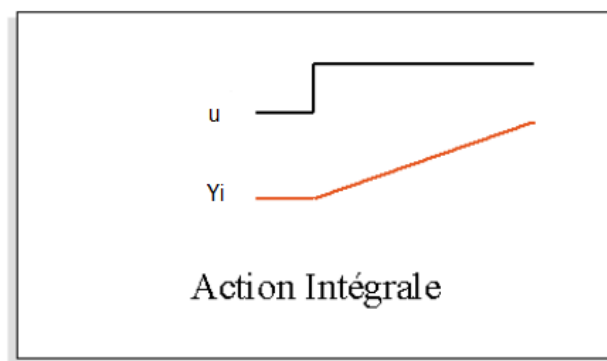
#### ❖ Action Intégrale

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

Cette action calcule sa contribution à la commande en n'utilisant pas l'écart instantané mais une mémoire des écarts intervenus lors des précédents échantillonnages. Ceci permet d'éliminer à la fois les erreurs statiques trop petites pour que l'action proportionnelle les contre et les erreurs de suivi de consignes en pente. [13]

La limite de cette commande intervient à partir du moment où le système ne réagit plus à la commande passée (blocage d'organe, limitation de sécurité, etc.). Dans ce cas précis, le régulateur va amplifier à l'infini sa contribution intégrale. On parle de saturation d'intégrale. La résolution de problèmes passe de l'utilisation d'un PID à intégrale externe.

La contribution intégrale se calcule en ajoutant tous les  $T_i$  secondes la valeur de l'entrée de l'intégrateur à l'ancienne valeur.



**Figure.III.5.** Réponse indiciel d'un régulateur I

**Equation discrète**

$$y^i(nT) = K_i \sum_{j=0}^n \varepsilon(jT) \tag{III.9}$$

$$K_i = \frac{T}{T_i} \tag{III.10}$$

**Equation récurrente**

$$y^i(nT) = y^i[(n - 1)T] + K_i \varepsilon(jT) \tag{III.11}$$

Pour plus de commodité, on posera, à l'instant considéré

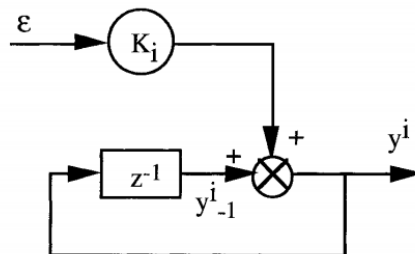
$$y^i = y^i_{-1} + K_i \varepsilon \tag{III.12}$$

**Fonction de transfert**

$$C_i(z) = \frac{y^i(z)}{\varepsilon(z)} = K_i \frac{z}{z - 1} = K_i \frac{1}{1 - z^{-1}} \tag{III.13}$$

**Diagramme structurel**

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



**Figure.III.6.** Diagramme structurel de régulateur I

❖ **Action dérivée D**

L'action dérivée se traduit par un terme proportionnel à la différence des écarts de réglage aux instants d'échantillonnage  $nT$  et  $(n-1) T$ .

Cette action surqualifie la commande à passer en la calculant à partir de la variation de l'erreur. Sur variation de l'erreur, l'action dérivée va intervenir mais s'annulera pour une

erreur constante. L'utilisation de l'action dérivée impose d'éliminer tous les bruits entachant la mesure sous peine de voir son système en mouvement permanent. [13]

L'action dérivée peut se calculer soit sur l'écart mesure consigne, soit sur la mesure. Le calcul sur la mesure se prête au cas de maintien et évite d'avoir une réaction trop grande en cas de variation de consigne.

Le réglage de la dérivée se fera en donnant  $T_d$  qui est un coefficient de multiplication de la valeur de la dérivée instantanée. La commande sera donc

$$y^d = T_d \frac{de}{dt} \tag{III.14}$$

$T_d$  est exprimé en unité de temps.

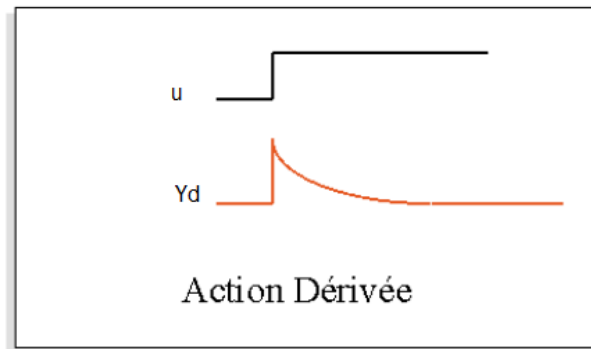


Figure.III.7. Réponse indicielle d'un régulateur D

**Equation discrète**

$$y^d(nT) = K_d[\varepsilon(nT) - \varepsilon[(n - 1)T]] \tag{III.15}$$

Avec  $K_d = T_d$

Soit

$$y^d = K_d(\varepsilon - \varepsilon_{-1}) \tag{III.16}$$

**Fonction de transfert**

$$C_i(z) = \frac{y^d(z)}{\varepsilon(z)} = K_d \frac{z - 1}{z} = K_d(1 - z^{-1}) \tag{III.17}$$

**Diagramme structurel**

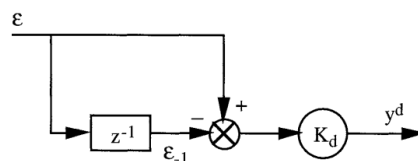


Figure.III.8. Diagramme structurel de régulateur D

**III.6.3.3. Le régulateur Proportionnel-Intégral P.I**

Le régulateur P.I est une combinaison d'un régulateur P et d'un régulateur I.

**Equation discrète**

$$y(nT) = K_p \varepsilon(nT) + K_i \sum_{j=0}^n \varepsilon(jT) \tag{III.18}$$

Comme précédemment, l'action intégrale peut s'exprimer par

$$K_i \sum_{j=0}^n \varepsilon(jT) = K_i \sum_{j=0}^{n-1} \varepsilon(jT) + K_i \varepsilon(nT) \tag{III.19}$$

Soit

$$y^i(nT) = y^i[(n - 1)T] + K_i \varepsilon(nT) \tag{III.20}$$

D'où

$$y(nT) = y^i[(n - 1)T] + (K_p + K_i) \varepsilon(nT) \tag{III.21}$$

$$y = y^i_{-1} + (K_p + K_i) \varepsilon \tag{III.22}$$

**Fonction de transfert**

$$C(z) = K_p + K_i \frac{z}{z - 1} = \frac{b_1 z + b_0}{z - 1} = \frac{b_1 + b_0 z^{-1}}{1 - z^{-1}} \tag{III.23}$$

Avec

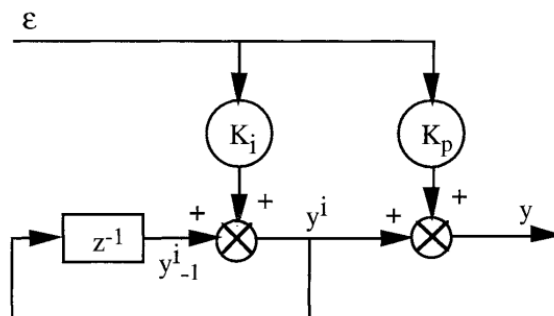
$$b_1 = K_p + K_i \quad \text{Et} \quad b_0 = -K_p$$

Le régulateur P.I est un système discret du premier ordre, ayant un pôle égal à 1.

Le régulateur P.I peut donc annuler l'écart de réglage, en régime établi, dû à une sollicitation en échelon de position.

**Diagramme structurel**

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



**Figure.III.9.** Diagramme structurel de régulateur PI

**III.6.3.4. Le régulateur universel P.I.D**

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

**Equation discrète**

$$y(nT) = K_p \varepsilon(nT) + K_i \sum_{j=0}^n \varepsilon(jT) + K_d [\varepsilon(nT) - \varepsilon[(n-1)T]] \quad (\text{III.24})$$

Soit

$$y^i(nT) = y^i[(n-1)T] + (K_p + K_i + K_d)\varepsilon(nT) - K_d \varepsilon[(n-1)T] \quad (\text{III.25})$$

**Fonction de transfert**

$$C(z) = K_p + K_i \frac{z}{z-1} + K_d \frac{z-1}{z} \quad (\text{III.26})$$

Soit

$$C(z) = \frac{b_2 z^2 + b_1 z + b_0}{z(z-1)} \quad (\text{III.27})$$

Avec

$$b_2 = K_p + K_i + K_d \quad , \quad b_1 = -(K_p - 2K_d) \quad , \quad b_0 = K_d$$

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

Ses pôles ne dépendent en aucune façon du réglage des actions Proportionnelle, Intégrale et Dérivée.

Les coefficients, n'interviennent que dans le numérateur de la transmittance et par conséquent que sur ses zéros.

**Diagramme structurel**

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

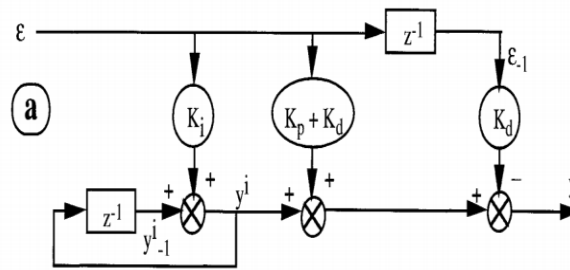


Figure.III.10. Diagramme structurel de régulateur PID

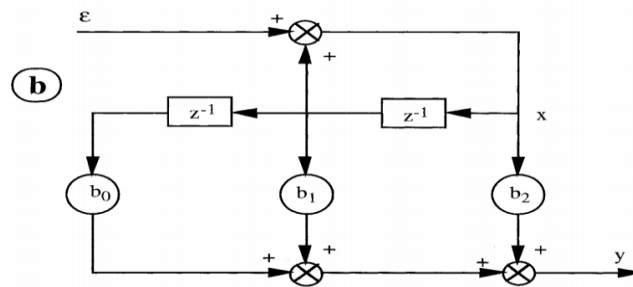


Figure.III.11. Diagramme structurel de régulateur PID

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

$$y = (K_p + K_i + K_d)\varepsilon + y_{-1}^i - K_d\varepsilon_{-1} \tag{III.28}$$

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

Données  $K_p$   
 $K_i$   
 $K_d$

.....  
 $K_{pid} = K_p + K_i + K_d \tag{III.29}$

$$X = y_{-1}^i \tag{III.30}$$

$$y = K_{pid}\varepsilon + X - K_d\varepsilon_{-1} \tag{III.31}$$

Sortir y

$$X = X + K_i\varepsilon \tag{III.32}$$

$$\varepsilon = \varepsilon_{-1}$$

Pour l'exécution de l'algorithme de réglage, on doit respecter la séquence prescrite. Tout de suite après le calcul de la grandeur de commande  $y$ , le calculateur de processus peut appliquer cette nouvelle valeur au système à régler. L'incrément de  $x$  par  $k_i \varepsilon$  et ensuite l'échange de  $\varepsilon$  à  $\varepsilon_{-1}$  peuvent se faire, en temps masqué, dans l'intervalle jusqu'au prochain instant d'échantillonnage.

Dans ce cas on doit imposer une valeur initiale adéquate à la grandeur auxiliaire  $x$ . En plus, on a aussi besoin d'une valeur initiale pour l'écart de réglage. En général, on choisira au début:  $x=0$  et  $\varepsilon_{-1}=0$ .

### III.6.3.5. Calcul des paramètres de régulateur PID

Nous allons désormais voir comment trouver les valeurs à attribuer aux trois coefficients ( $K_i$ ,  $K_p$ ,  $K_d$ ) de l'asservissement PID. Il existe deux façons de procéder, l'une par la modélisation, l'autre par l'expérimentation.

Modéliser le système consiste à déterminer par le calcul son comportement et de là déduire des valeurs plausibles pour les coefficients. L'approche par l'expérimentation signifie que l'on va utiliser une réponse réelle du système pour régler d'abord grossièrement puis finement les coefficients. [17]

#### III.6.3.5.1. Approche par les méthodes expérimentales

Le choix de procéder à la modélisation du système ou non est dicté par les contraintes inhérentes au système. Souvent la complexité des systèmes réels place la modélisation hors d'atteinte, mais dans certains cas, en chimie ou en mécanique par exemple, les règles qui régissent le système sont suffisamment simples pour que l'on puisse tenter de modéliser le système. [17]

D'autre part, il se peut que dans les cas où le système est particulièrement critique ou difficile d'accès (processus industriel lourd, système qui ne peut être mis hors service), il soit indispensable de modéliser le système afin d'avoir un jeu de coefficients suffisamment précis pour obtenir d'emblée un régulateur PID qui soit fonctionnel.

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

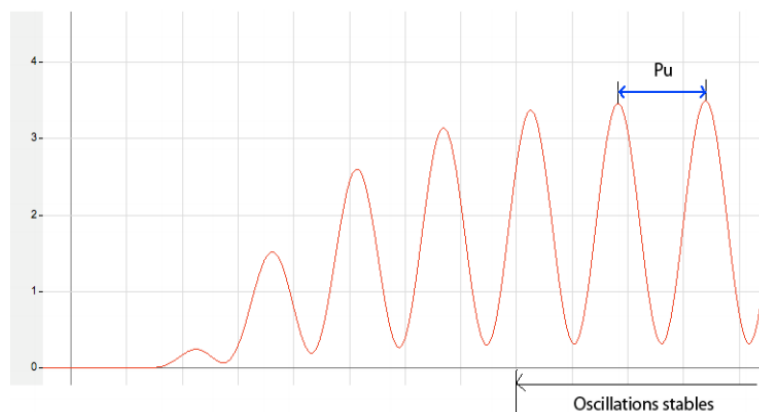


Nous présentons ici deux approches possibles pour la détermination des coefficients par expérimentation, une méthode avec l'utilisation du régulateur PID et l'autre qui ne l'utilise pas.

Nous indiquons aussi les avantages et inconvénients de ces deux méthodes. L'existence de la deuxième méthode est justifiée par le fait que dans certains régulateurs matériels, les valeurs des coefficients ne sont pas modifiables, c'est le cas si par exemple le régulateur est un circuit électronique et que les coefficients sont déterminés par les composants (résistors, condensateurs).

### III.6.3.5.2. Approche de Ziegler-Nichols pour les systèmes en boucle fermée (déjà réglés)

Le Principe est amené le système dans un état d'oscillations puis en déduire les valeurs des coefficients, on fixe  $K_i$  et  $K_d$  à 0 puis faire varier  $K_p$  jusqu'à obtenir des oscillations périodiques stables, c'est à dire non amorties et non amplifiées.



**Figure.III.12.** Approche de Ziegler-Nichols pour les systèmes en boucle fermée

Valeurs des coefficients

On note :  $K_u = K_{oscillations}$ ,  $P_u =$  Période des oscillations

Alors il faut choisir  $K_p = \frac{K_u}{1.7}$ ,  $K_i = \frac{P_u}{2}$ ,  $K_d = \frac{P_u}{8}$

#### Avantage

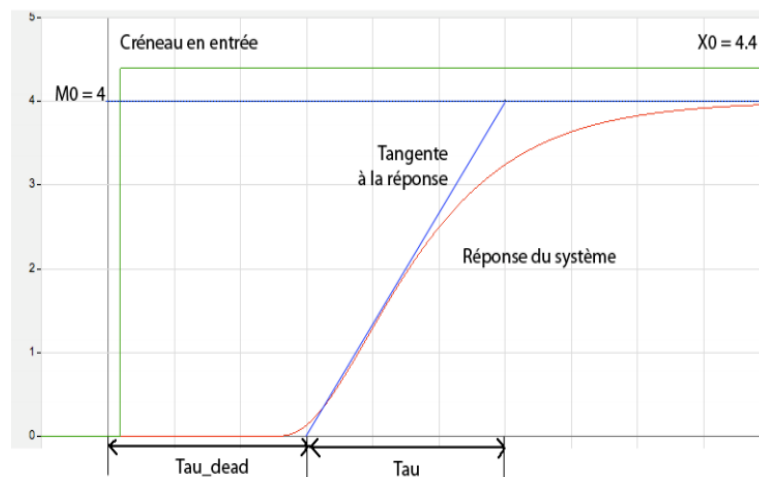
- La méthode est facile à mettre en œuvre physiquement et au point de vue calcul
- Elle peut être appliquée à un système déjà en production et permet une adaptation automatisée du régulateur pour s'adapter à l'évolution des paramètres intérieurs (usure) et extérieurs (environnement) au système.

**Inconvénients**

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

**III.6.3.5.3. Approche de Ziegler-Nichols pour les systèmes en boucle ouverte (pas encore réglés)**

On applique un créneau au système et on enregistre sa réponse.



**Figure.III.13.** Approche de Ziegler-Nichols pour les systèmes en boucle ouverte

On pose

$$K0 = \frac{X0}{M0} \cdot \frac{\tau}{\tau_{dead}}$$

on a alors  $Kp = 1.2 * K0$   $Ki = 2 * \tau_{dead}$  et  $Kd = 0.5 * \tau_{dead}$

**Avantages**

Cette méthode ne nécessite pas d'avoir un système déjà asservi, ni de deviner une valeur pour  $Kp$ .

**Inconvénients**

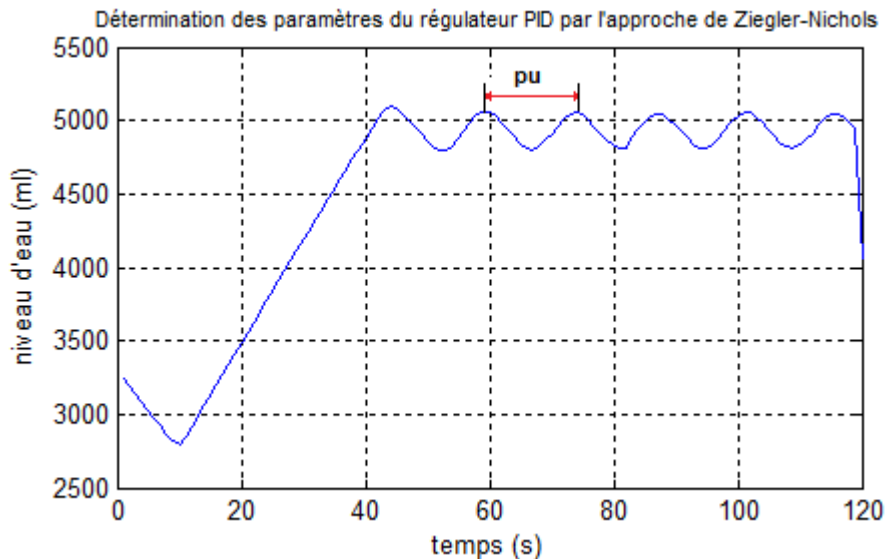
- Elle nécessite plus de matériel : enregistrement d'une courbe de réponse.
- Elle oblige à mettre le système "offline"

-Si la réponse du système est trop différente de la réponse montrée ci-dessus, cette méthode donne des valeurs qui peuvent ne pas convenir du tout (certains systèmes par exemple ne sont pas du tout réglables par des régulateurs PID).

### III.7. Simulation MATLAB des régulateurs utilisés pour le système

Notre objective dans ce chapitre est de faire une régulation de niveau d'une cuve, pour cet effet nous utilisons un régulateur de type PID.

Pour la régulation de ce système nous utilisons une structure parallèle de régulateur PID, en premier temps nous calculons les paramètres du régulateur PID à travers la méthode de **Ziegler-Nichols**, après les essais sur le système réel nous avons trouvé les résultats suivants :



**Figure.III.14.** Calcul de paramètres PID par la méthode de Ziegler-Nichols

Après plusieurs changements du paramètre proportionnel  $K_u$ , nous trouvons que le graphe de la sortie du système à des oscillations stable pour la valeur  $K_u=47$ .

D'après le graph, on a  $P_u=16$ , donc les paramètres du régulateur PID sont :

$$K_p = \frac{K_u}{1.7} = \frac{47}{1.7} \approx 28.$$

$$T_i = \frac{P_u}{2} = 8 \Rightarrow K_i = \frac{T}{T_i} = \frac{0.02}{8} = 0.0025.$$

$$T_d = 2.$$

Nous avons utilisé deux méthodes avec MATLAB pour la régulation du système, dans la première méthode nous calculons le régulateur numérique par la méthode directe de synthèse des régulateurs numériques. Pour la deuxième méthode nous utilisons l'algorithme de régulateur PID discret.

### III.7.1. Régulation de système par un régulateur numérique R(z)

Pour faire une simulation MTLAB de la régulation de notre système, nous utilisons le modèle mathématique calculé dans le chapitre 02 Pour la synthèse d'un régulateur numérique

$$G(z) = \left( \frac{0.0137z^{-1}}{1 - 1.0001z^{-1}} \right) u(z) \quad (\text{III.33})$$

$$G(z) = \frac{y(z)}{u(z)} = \frac{0.0137z^{-1}}{1 - 1.0001z^{-1}} \quad (\text{III.34})$$

Nous utilisons la méthode directe de synthèse des régulateurs numériques, nous proposons F(z) la fonction de transfert en boucle fermée désirée.

Elle admet un gain statique unitaire, donc garantissant une précision statique parfaite, elle n'assure pas un dépassement indiciel comme le montre la réponse à un échelon unitaire ci-dessous.

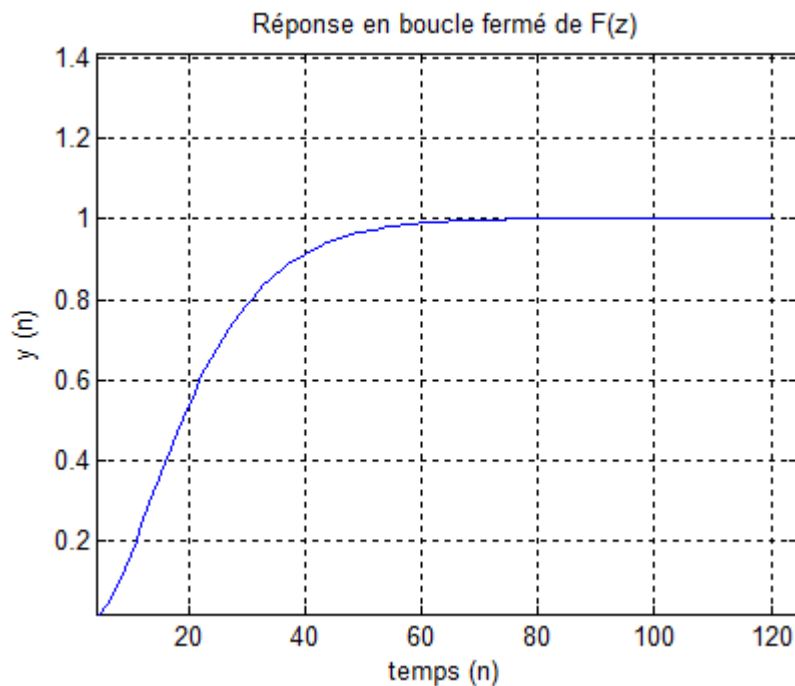


Figure.III.15.Réponse on boucle fermé de F(z)

Le régulateur  $R(z)$  s'obtient par

$$R(z) = \frac{F(z)}{G(z)(1 - F(z))} = \frac{0.0093z^{-2} + 0.419z^{-1} - 0.0001}{0.353z^{-2} - 1.24z^{-1} + 1} \quad (\text{III.35})$$

$$R(z) = \frac{u(z)}{e(z)} \quad (\text{III.36})$$

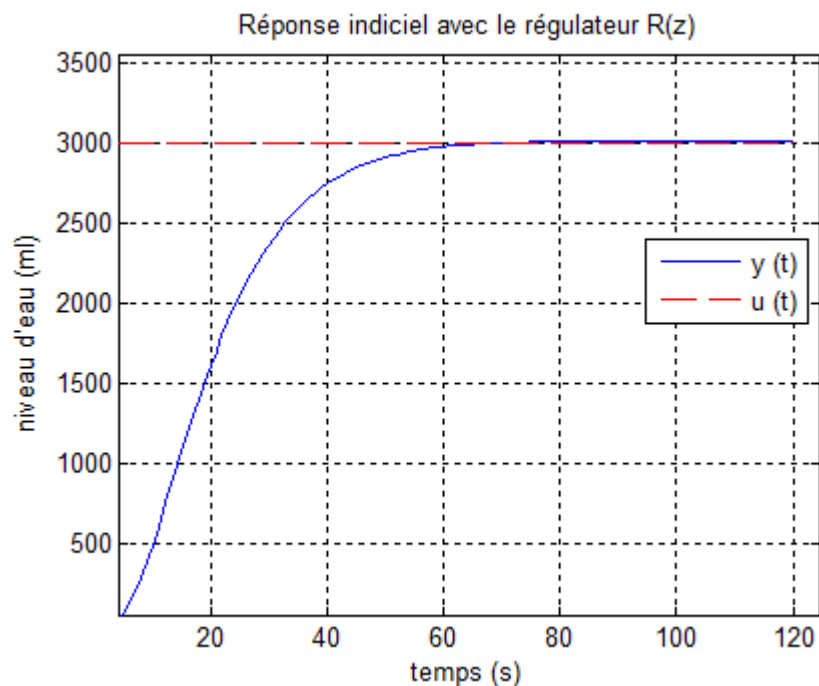
Donc

$$u(z)(0.353z^{-2} - 1.24z^{-1} + 1) = e(z)(0.0093z^{-2} + 0.419z^{-1} - 0.0001) \quad (\text{III.37})$$

La loi de commande qui découle de  $R(z)$  est

$$u(n) = 1.24u(n - 1) - 0.353u(n - 2) - 0.0001e(n) + 0.419e(n - 1) + 0.0093e(n - 2) \quad (\text{III.38})$$

Après le calcul de la loi de commande obtenue par le régulateur, nous avons fait une simulation MATLAB de ce régulateur, nous donnons un échelon d'amplitude de 3000 ml comme entrée du système, après la simulation nous obtenons le résultat suivant :



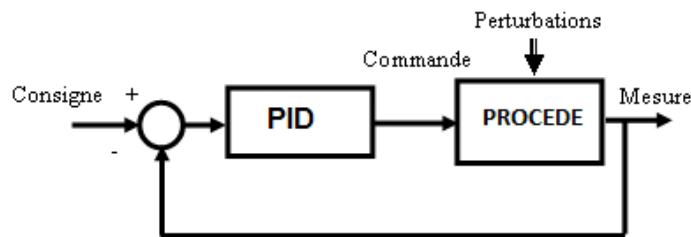
**Figure.III.16.**Réponse indicielle du système avec le régulateur  $R(z)$

Le résultat obtenu montrent que le système est bien réglé après un temps de réponse de 57 s, nous observons qu'il y'a pas de dépassement dans la réponse indiciel, donc on peut dire que le régulateur calculé est acceptable.

**III.7.2. Régulation PID du système**

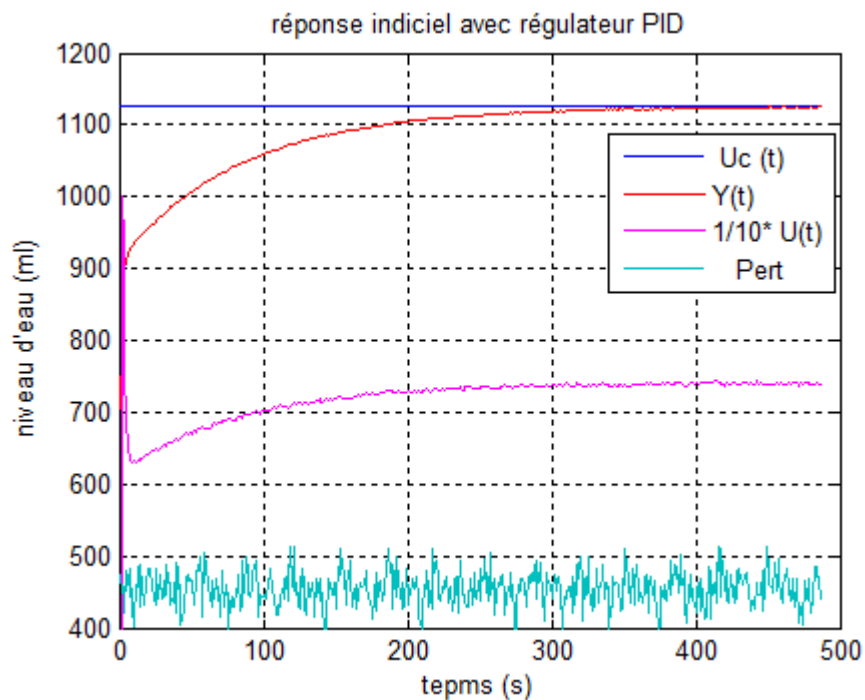
Dans cette simulation nous utilisons les paramètres de PID calculé précédemment, nous utilisons le modèle calculée dans le 2<sup>ème</sup> chapitre

$$y(z) = \left( \frac{0.0365z^{-1}}{1 - 1.0007z^{-1}} \right) u(z) + \left( \frac{0.0041z^{-1}}{1 - 1.0007z^{-1}} \right) e(z) \tag{III.39}$$



**Figure.III.17.**Structure de régulation PID du système

Nous excitons le système par une consigne de type échelon d'amplitude de 30 %, après la simulation MATLAB avec l'algorithme de régulateur PID on trouve les résultats suivant :



**Figure.III.18.**Réponse indiciel de système avec le régulateur PID

La réponse indicielle obtenue montre que le système est réglé après un temps de réponse de 5 Minutes, il n'y a pas de dépassement et d'oscillations, donc on peut dire que le régulateur est acceptable, il répond à notre besoin.

### III.8. Implémentation du régulateur PID sous l'automate programmable

Notre système est commandé par un automate programmable industrielle, pour régler le niveau d'eau dans les deux cuve.

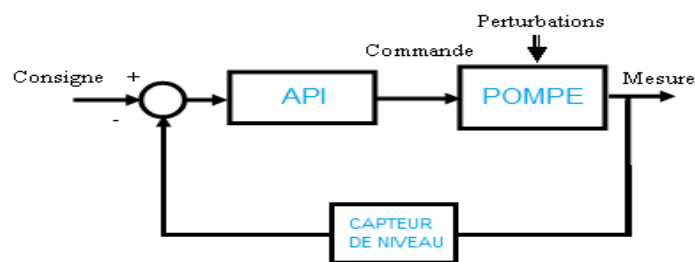


Figure.III.19. Structure de régulation de système en boucle fermée

Nous implémentons l'algorithme du régulateur PID utilisé dans la simulation sous l'automate, pour cet effet nous développons un bloc DFB avec PL7 qui s'appelle « Reg\_pid ».

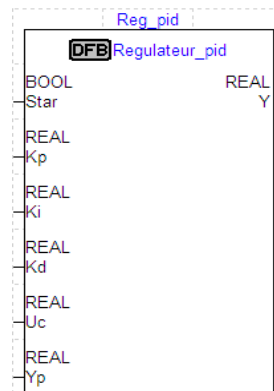


Figure.III.20. Bloc DFB de régulateur PID

#### III.8.1. Réponse indicielle du système réel

Après l'étalonnage des capteurs de pression qui capture le niveau d'eau dans les cuves, la commande est appliquée sur l'électropompe pour régler le niveau d'eau dans le réservoir 01, la consigne est un échelon d'amplitude de 30%, nous avons obtenu les résultats suivants

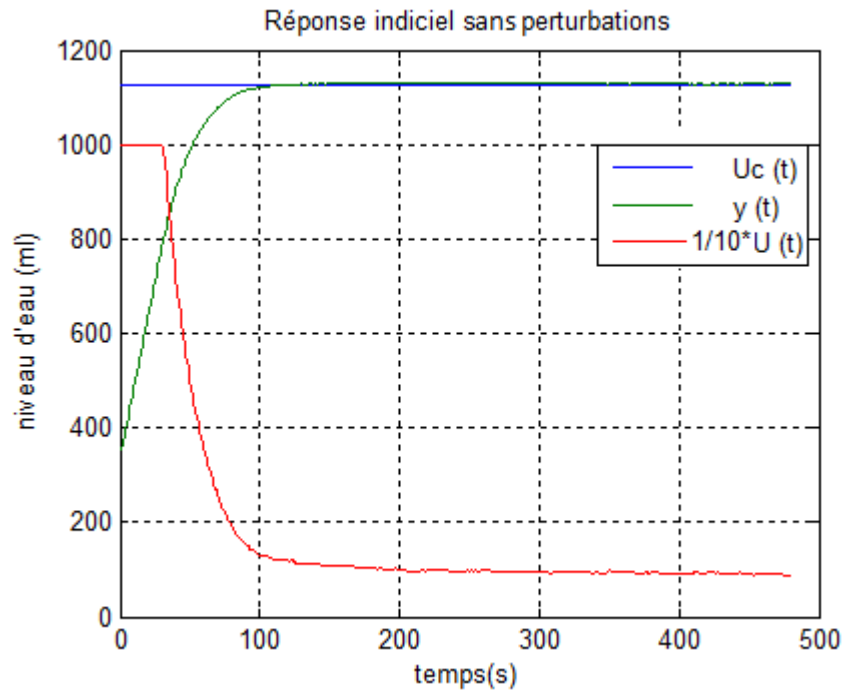


Figure.III.21. Réponse indiciel de système sans perturbations

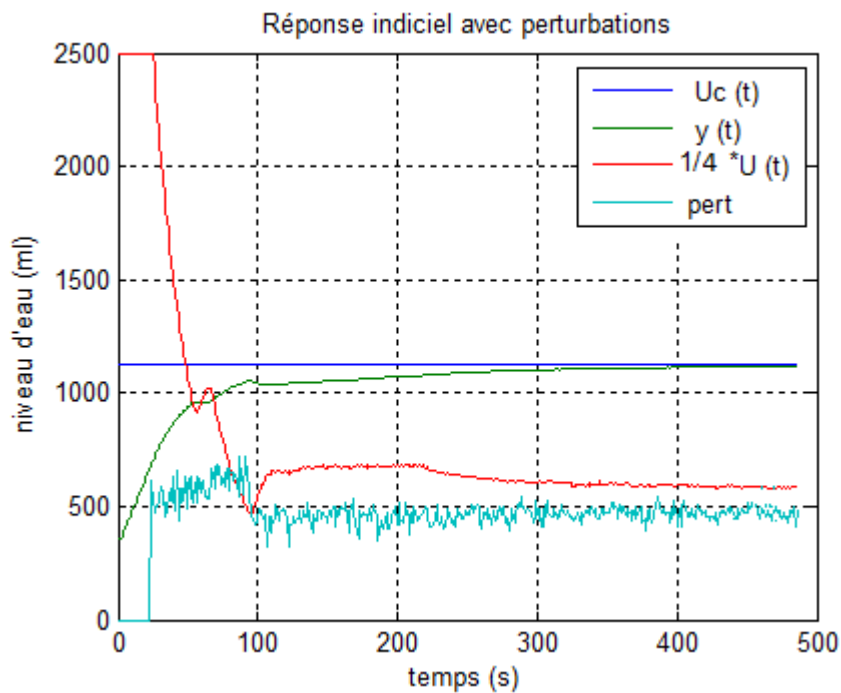


Figure.III.22. Réponse indiciel de système avec perturbations

Nous remarquons que dans le cas de la commande avec perturbations le temps de réponse est de 5 Minutes, et qu'il y a des petites osculations. Par contre dans le cas sans



perturbation le temps de réponse est diminué à 95 secondes, donc le système réagira rapidement par rapport au cas où il y a des perturbations.

**III.8.2. Réponse du système real avec une consigne aléatoire**

Pour tester si la sortie du système réglée est suivre la consigne désirée, nous excitons le système par un programme de consigne, pour cet effet nous développons un bloc DFB avec PL7 qui s'appelle « Prog\_consigne ».

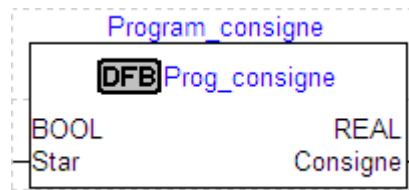


Figure.III.23. Bloc DFB de programme de consigne

Nous obtenons les résultats suivants

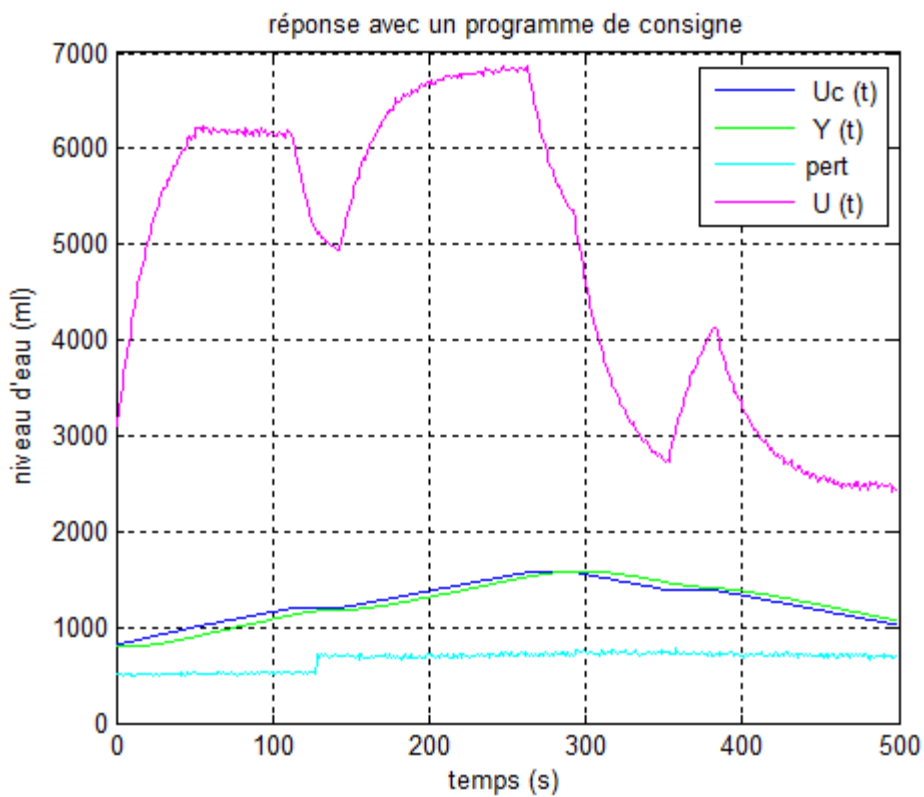


Figure.III.24. Réponse de système avec programme de consigne

Les résultats obtenus montrent que la sortie du système suit la consigne, donc le régulateur est bien paramétré et les graphes montrent la robustesse du régulateur PID.

### **III.9. Conclusion**

Les régulateurs PID sont les plus utilisés dans l'industrie, ils sont robustes, et répondent aux besoins des constructeurs des systèmes automatisés.

Dans ce chapitre nous avons fait la régulation de niveau d'eau dans les réservoirs, dans un premier temps nous calculons les paramètres du régulateur avec la méthode expérimentale **de Ziegler-Nichols**, puis nous avons utilisé les modèles mathématiques calculés dans le chapitre 02 pour faire une simulation Matlab des deux régulateurs numériques, nous avons trouvé des résultats acceptables, le système est réglé et il n'y a pas de dépassement dans leur réponse indicielle.

Enfin nous avons implémenté l'algorithme du régulateur PID sous l'automate programmable, ou nous avons tracé la sortie de système réglé pour deux types de consigne sur l'écran de supervision Magelis, les résultats obtenus montrent que la sortie du système suit la consigne, d'après ces résultats on peut dire que le régulateur PID est robuste il répond à nos besoins.

# *Conclusion générale*

## Conclusion générale

---

Notre but est d'identifier et réguler le débit et le niveau d'eau de deux réservoirs placés dans un système hydraulique. Ce système simule une station de pompage réelle.

Nous avons divisé notre mémoire en trois chapitres :

Dans le premier chapitre, nous avons présenté les systèmes automatisés, nous avons expliqué leurs différentes parties. La partie commande et la partie opérative, puis nous avons présenté le banc de régulation du niveau et du débit de l'eau, nous avons cité la partie opérative de ce banc (actionneurs, capteurs ...), et sa partie commande (automate programmable, variateur de vitesse ...). A la fin de ce chapitre nous avons fait une supervision du système hydraulique étudié, pour cet effet nous avons proposé quatre écrans de supervision.

Le deuxième chapitre est consacré à l'identification du système étudié. Premièrement nous avons estimé l'ordre du système par la méthode de Strejc et nous avons trouvé que le système est du 1<sup>er</sup> ordre, puis nous avons fait l'identification du système par deux méthodes d'identification paramétrique, la première méthode est la méthode des moindres carrés récursive MCR, nous avons trouvé que les résultats obtenus avec cette méthode sont biaisés, pour cela nous avons utilisé la deuxième méthode qui est la méthode des moindres carrés étendue. Avec cette méthode le biais est bien éliminé. Enfin nous avons cité les modèles mathématiques du système obtenus par ces deux méthodes.

Dans le troisième chapitre, nous avons fait la régulation de niveau d'eau dans les cuves. Premièrement nous avons fait une simulation MATLAB des deux régulateurs numériques ou nous avons calculé le premier régulateur par la méthode directe des synthèses de régulateur numérique. Le deuxième régulateur c'est le régulateur PID discret. Puis nous avons fait une régulation du système réel pour cela nous avons implémenté l'algorithme du régulateur PID dans l'automate. Nous avons trouvé que le système est bien réglé et que les résultats obtenus sont justes.

## REFERENCES BIBLIOGRAPHIQUES

- [1] PHILIPPE HOARAU, « *L'Automate Programmable Industriel* », TS MAI,
- [2] « *L. BERGOUGNOUX, « Automates Programmables Industriels* », POLYTECH' Marseille
- [3] G. MICHEL, « *Les A.P.I Architecture et application des automates programmables industriels* », Edition DUNOD, 1987.
- [4] «*Automates Modicon Premium TSX 57/PCX 57*», Catalogue Schneider Electric (2009)
- [5] «*Altivar 31* », Catalogue Schneider Electric (2009)
- [6] «*Gamme Magelis Graphique XBT-F / TXBT-F*», Catalogue Schneider Electric (2009)
- [7] «*manuelle de référence du logiciel PL7*», Catalogue Schneider Electric (2009)
- [8] Stéphane Colonges cours « *identification des systèmes en boucle fermé* » Janvier 2002.
- [9] Prof. D. Bonvin, Dr. A. Karimi « *Identification de systèmes dynamiques*» Lausanne Septembre 2007.
- [10] Gonzalo Cabodevila. Cours « *Identification des systèmes*» 2009.
- [11] I.D. LANDAU, « *Identification et commande des systèmes* », Hermès, 1993
- [12] <http://www.wikipédia.com>
- [13] cours «*La régulation*», Le magazine Schneider Electric de l'enseignement technologique et professionnel Juin 2004
- [14] ROLAND .LONGCHAMP « *commande numérique de systèmes dynamique cours d'automatique* » presses polytechniques et universitaire romandes 2006.
- [15] <http://www.specialautom.net>.
- [16] A. JUTARD M.BETEMPS « *Automatique –S.A.E* » INSA de Lyon 1998.
- [17] Daniel Ross .Etienne Deguine. Mickaël Camus cours « *Asservissement par PID* » mars 2010

# *Annexes*

### Annexe A

## Les programmes de simulation sous Matlab

### ✚ L'algorithme de moindré carré récurfive MCR

```
% nombres d'itérations %
N=128;
% les couples entrées sortie de système %
y= VarName2 ;
u= VarName1 ;
% les conditions initial %
theta=zeros(2,1);
f=1000*eye(2);
lam1=1;lam0=1;
phi=zeros(2,1);
j=[1 2];
% l'algorithme de MCR %
for t=1:N
    ym(t)=theta'*phi(j);
    e_apri(t)=y(t)-ym(t);
    e_apost(t)=e_apri(t)/(1+phi(j)'*f*phi(j));
    theta=theta+f*phi*e_apost(t);
    f=1/lam1*(f-((f*phi(j)*phi(j)'*f)/(lam1+phi(j)'*f*phi(j))));
    lam1=lam0*lam1+1-lam0;
    phi(1)=-y(t);
    phi(2)=u(t);
end
```

### ✚ L'algorithme de moindré carré étendu récurfive MCE

```
% nombres d'itérations %
N=128;
% les couples entrées sortie de système %
y= VarName2 ;
u= VarName1 ;
% les conditions initial %
theta=zeros(3,1);
f=1000*eye(3);
lam1=1;lam0=1;
phi=zeros(3,1);
j=[1 2 3];
% l'algorithme de MCE %
for t=1:N
    ym(t)=theta'*phi(j);
    e_apri(t)=y(t)-ym(t);
    e_apost(t)=e_apri(t)/(1+phi(j)'*f*phi(j));
    theta=theta+f*phi*e_apost(t);
    f=1/lam1*(f-((f*phi(j)*phi(j)'*f)/(lam1+phi(j)'*f*phi(j))));
    lam1=lam0*lam1+1-lam0;
    ym(t)=theta'*phi(j);
    phi(1)=-y(t);
    phi(2)=u(t);
    phi(3)=e_apost(t);
end
```

### ✚ L'algorithme de régulation du système par le régulateur numérique R(z)

```
% la consigne %
u=3000*ones(128,1);
% les conditions initial %
y(2)=0;uc(1)=0;uc(2)=0;e(1)=0;e(2)=0;
```

## Annexe A

---

```
% l'algorithme de régulateur %
for i=3:120
    %l'erreur entre la consigne et la sortie %
    e(i)=u(i)-y(i-1);
    % la commande uc %
    uc(i)=1.24*uc(i-1)-0.353*uc(i-2)-0.0001*e(i)+0.419*e(i-1)+0.0093*e(i-2);
    %la sortie de système %
    y(i)=1.0001*y(i-1)+0.0137*uc(i);
end
```

### L'algorithme de régulation du système par le régulateur PID

```
% la consigne %
u=VarName1;
% les paramètres de régulateur PID%
kp=28;
ki=0.0025;
kd=2;
%les condition initial%
y(1)=705;
yi(1)=0;
%la sommes des paramètres PID (utilisé dans l'algorithme de réglage)%
kpid=kp+ki+kd;
%les perturbation %
p=1000*rand(120,1);i=2;
% l'algorithme de régulateur PID %
for j=0:0.02:2.4
    x=yi(i-1);
    % l'erreur entre la consigne et la sortie %
    e(i)=u(i)-y(i-1);
    % la commande uc %
    uc(i)=kpid*e(i)+x-kd*e(i-1);
    if uc(i)>10000
        uc(i)=10000;
    end
    if uc(i)<0
        uc(i)=0;
    end
    % la sortie du système %
    y(i)=1.0007*y(i-1)+0.0365*uc(i)-0.0041*p(i);
    yi(i)=x+ki*e(i);
    i=i+1;
end
```

### L'algorithme de la validation des modèles obtenus par l'identification

```
% l'erreur entre la la sortie real et la sortie de modèle %
E=y-ym;
% la moyenne de l'erreur %
moy=mean(E);
err=E-moy;q=0;r=0;
for i=2:128
    q=q+err(i)*err(i);
    r=r+err(i)*err(i-1);
end
q=q+err(1)*err(1);
R_i=q/128;
R_0=r/128;
RN_i=R_i/R_0;
```



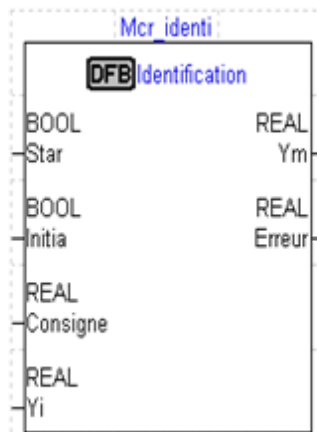
**Annexe B**

**Les programmes implémentés sous l'automate programmable**

❖ **Algorithme de l'identification par MCR**

**Description**

Objet qui réalise l'identification par MCR



Entrées	Format	Description
Star	BOOL	Démarre l'objet
initia	BOOL	Initialisation l'algorithme
consigne	REAL	La consigne
Yi	REAL	La sortie du système
Sorties	Format	Description
Ym	REAL	La sortie du modèle
Erreur	REAL	L'erreur entre Yi et Ym

🚦 L'algorithme de MCE avec langage ST sous PL7

```

IF Star THEN
    IF Initia THEN
        (* initialisation des valeurs *)
        Teta1p:=0.0; Teta2p:=0.0;
        G1p:=1000.0; G2p:=0.0; G3p:=0.0; G4p:=1000.0;
        Consigne:=0.0; Yi:=0.0; Ym:=0.0;
    ELSE
        (* calcul de la sortie du modèle Ym *)
    
```

```

Ym:=Teta1p*Yi+Consigne*Teta2p;

(* calcule les éléments de matrice de gain *)

G1:=G1p-(((G1p*Yi+G3p*Consigne)*G1p*Yi) + ((G1p*Yi+G3p*Consigne)* G3p* Consigne)) *
(1.0 / (1.0+Yi*(G1p*Yi+G3p*Consigne) + (Consigne* (G2 p*Yi+G4p * Consigne))));

G2:=G2p-(((G1p*Yi+G3p*Consigne)*G2p*Yi) + ((G1p*Yi+G3p*
Consigne)*G4p*Consigne))*(1.0/ (1.0+Yi* (G1p*Yi+G3p*Consigne) + (Consigne*
(G2p*Yi+G4p*Consigne))));

G3:=G3p-(((G2p*Yi+G4p*Consigne)*G1p*Yi) +
((G2p*Yi+G4p*Consigne)*G3p*Consigne))*(1.0/ (1.0+Yi*(G1p*Yi+G3p*Consigne) +
(Consigne*(G2p*Yi+G4p*Consigne))));

G4:=G4p-(((G2p*Yi+G4p*Consigne)*G2p*Yi) +
((G2p*Yi+G4p*Consigne)*G4p*Consigne))*(1.0/ (1.0+Yi*(G1p*Yi+G3p*Consigne) +
(Consigne*(G2p*Yi+G4p*Consigne))));

(*calcule les valeurs de téta *)

Teta1:=Teta1p+ ((G1*Yi+G2*Consigne)*(Y-(Yi*Teta1+Consigne*G2)));

Teta2:=Teta2p+ ((G3*Yi+G4*Consigne)*(Y-(Yi*Teta1+Consigne*G2)));

Consigne:=Consignt;

Yi:=-1.0*Y;

(*enregistrement des valeurs précédents *)

Teta1p:=Teta1; Teta2p:=Teta2;

G1p:=G1; G2p:=G2; G3p:=G3; G4p:=G4;

END_IF;

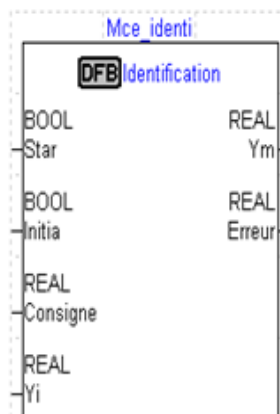
```

END\_IF;

## ❖ Algorithme de l'identification par MCE

### Description

Objet qui réalise l'identification par MCE



## Annexe B

Entrées	Format	Description
Star	BOOL	Démarre l'objet
initia	BOOL	Initialisation l'algorithme
consigne	REAL	La consigne
Yi	REAL	La sortie du système
Sorties	Format	Description
Ym	REAL	La sortie du modèle
Erreur	REAL	L'erreur entre Yi et Ym

### L'algorithme de MCE avec langage ST sous PL7

IF Star THEN

IF Initia THEN

(\* initialisation des valeurs \*)

Teta1p:=0.0; Teta2p:=0.0; Teta3p:=0.0;

G1p:=1000.0; G2p:=0.0; G3p:=0.0; G4p:=0.0; G5p:=1000.0; G6p:=0.0; G7p:=0.0; G8p:=0.0;  
G9p:=1000.0;

Erreur:=0.0; Ym:=0.0;

ELSE

Division: =1.0/ (1.0+ ((-1.0\*Yip\*(-1.0\*Yip\*G1p+ Consignp\*G4p +Erreur\*G7p)) +Consignp\*(-  
1.0\*Yip\*G2p + Consigne\*G5p + Erreur\*G8p) + Erreur \*(-1.0\*Yip\*G3p + Consignp\*G6p + Erreurp \*  
G9p)));

Ymp:=-1.0\*Teta1p\*Yip+Consigne\*Teta2p+Erreur\*Teta3p;

(\* calcule l'erreur entre la sortie du modèle et la sortie réel du système \*)

Erreur:=Yi-Ymp;

(\* calcule les éléments de la matrice du gain \*)

G1:=G1p-((-1.0\*Yip\*G1p\*(-1.0\*Yip\*G1p+Consignp\*G2p+Erreur\*G3p) + Consignp \* G4p \* (-  
1.0\*Yip\*G1p+Consignp\*G2p+Erreur\*G3p) + Erreur\*G7p\*(-1.0\* Yip\*G1p + Consignp\* G2p +  
Erreur \* G3p)) \*Division);

G2:=G2p-((-1.0\*Yip\*G2p\*(-1.0\*Yip\*G1p+Consignp\*G2p+Erreur\*G3p) +Consignp\*G5p\*(-  
1.0\*Yip\*G1p+Consignp\*G2p+Erreur\*G3p) +Erreur\*G8p\*(-1.0\*Yip\*G1p+Consignp\*G2p  
+Erreur\*G3p))\*Division);

G3:=G3p-((-1.0\*Yip\*G3p\*(-1.0\*Yip\*G1p+Consignp\*G2p+Erreur\*G3p) +Consignp\*G6p\*(-  
1.0\*Yip\*G1p+Consignp\*G2p+Erreur\*G3p) +Erreur\*G9p\*(-1.0\*Yip\*G1p +Consignp\*G2p  
+Erreur\*G3p))\*Division);

G4:=G4p-((-1.0\*Yip\*G1p\*(-1.0\*Yip\*G4p+Consignp\*G5p+Erreur\*G6p) +Consignp\*G4p\*(-  
1.0\*Yip\*G4p+Consignp\*G5p+Erreur\*G6p) +Erreur\*G7p\*(-1.0\*Yip\*G4p+Consignp\*G5p  
+Erreur\*G6p))\*Division);

G5:=G5p-((-1.0\*Yip\*G2p\*(-1.0\*Yip\*G4p+Consignp\*G5p+Erreur\*G6p)+Consignp\*G5p\*(-  
1.0\*Yip\*G4p+Consignp\*G5p+Erreur\*G6p)+Erreur\*G8p\*(-1.0\*Yip\*G4p+Consignp\*G5p  
+Erreur\*G6p))\*Division);

```
G6:=G6p-((-1.0*Yip*G3p*(-1.0*Yip*G4p+Consignp*G5p+Erreurp*G6p)+Consignp*G6p*(-1.0*Yip*G4p+Consignp*G5p+Erreurp*G6p)+Erreurp*G9p*(-1.0*Yip*G4p+Consignp*G5p+Erreurp*G6p))*Division);
```

```
G7:=G7p-((-1.0*Yip*G1p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p)+Consignp*G4p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p)+Erreurp*G7p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p))*Division);
```

```
G8:=G8p-((-1.0*Yip*G2p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p)+Consignp*G5p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p)+Erreurp*G8p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p))*Division);
```

```
G9:=G9p-((-1.0*Yip*G3p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p)+Consignp*G6p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p)+Erreurp*G9p*(-1.0*Yip*G7p+Consignp*G8p+Erreurp*G9p))*Division);
```

(\* calcule les éléments de téta \*)

```
Teta1:=Teta1p+((-1.0*G1*Yip+G2*Consignp+G3*Erreurp)*Erreur);
```

```
Teta2:=Teta2p+((-1.0*G4*Yip+G5*Consignp+G6*Erreurp)*Erreur);
```

```
Teta3:=Teta3p+((-1.0*G7*Yip+G8*Consignp+G9*Erreurp)*Erreur);
```

(\* la sortie ym du modèle \*)

```
Ym:=-1.0*Teta1*Yip+Consignp*Teta2+Erreurp*Teta3;
```

(\* enregistrement des valeurs précédents \*)

```
Consignp:=Consigne; Yip:=Yi; Erreurp:=Erreur;
```

```
Teta1p:=Teta1; Teta2p:=Teta2; Teta3p:=Teta3;
```

```
G1p:=G1; G2p:=G2; G3p:=G3; G4p:=G4; G5p:=G5; G6p:=G6; G7p:=G7; G8p:=G8; G9p:=G9;
```

```
END_IF;
```

```
END_IF;
```

### ❖ Programme de signal programme de consigne

#### Description

Objet qui génère le programme de consigne



## Annexe B

Entrées	Format	Description
Star	BOOL	Démarre l'objet
Sorties	Format	Description
Consigne	REAL	Le signal programme de consigne

✚ L'algorithme de signal programme de consigne avec langage ST sous PL7

(\* programme de consigne \*)

IF Star THEN

IF %MW92<>2 THEN

%MW50:=0;

%MF94:=1200.0;

ELSE

%MW50:=%MW50+1;

IF %MW50<3000 THEN

%MF94:=%MF94+0.12666;

ELSIF %MW50>=3000 AND %MW50<6000 THEN

%MF94:=%MF94+0.1666666;

ELSIF %MW50>=6000 AND %MW50<7500 THEN

%MF94:=%MF94;

ELSIF %MW50>=7500 AND %MW50<13500 THEN

%MF94:=%MF94+0.1666666;

ELSIF %MW50>=13500 AND %MW50<15000 THEN

%MF94:=%MF94;

ELSIF %MW50>=15000 AND %MW50<18000 THEN

%MF94:=%MF94-0.1666666;

ELSIF %MW50>=18000 AND %MW50<19500 THEN

%MF94:=%MF94;

ELSIF %MW50>=19500 AND %MW50<25500 THEN

%MF94:=%MF94-0.1666666;

ELSE

%MW50:=0;

END\_IF;

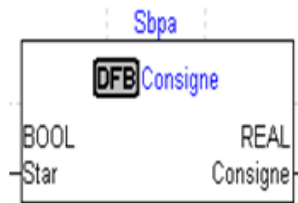
END\_IF;

END\_IF

### ❖ Programme du signal SBPA

#### Description

Objet qui génère le signal SBPA



Entrées	Format	Description
Star	BOOL	Démarre l'objet
Sorties	Format	Description
Consigne	REAL	Le signal SBPA

✚ L'algorithme de signal SBPA avec langage ST sous PL7

(\* le signal SBPA \*)

IF Star THEN

    IF RE %S6 THEN

        %MW0=%MW0+1;

        IF (%MW0>=3 AND %MW0<12) OR (%MW0=22) OR (%MW0>=25 AND %MW0<30)  
        OR (%MW0>=50 AND %MW0<53) OR (%MW0=59) OR (%MW0>=65 AND  
        %MW0<75) OR (%MW0>=84 AND %MW0<87) OR (%MW0>=91 AND %MW0<98) OR  
        (%MW0>=119 AND %MW0<126) THEN

            %MW1:= 5000;

        ELSE

            %MW1:=0;

        END\_IF;

    END\_IF;

IF %MW0>128 THEN

    %MW0:=0;

ELSE

    %MW0:=0;

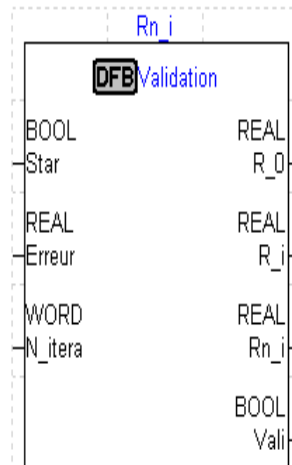
END\_IF;

END\_IF;

### ❖ Programme de validation des modèles

#### Description

Objet qui fait la validation des modèles d'identification



Entrées	Format	Description
Star	BOOL	Démarre l'objet
Erreur	REAL	Erreur entre Y et Ym
N_itera	REAL	Nombreur d'itérations
Sorties	Format	Description
R_0	REAL	La valeur de R_0
R_i	REAL	La valeur de R_1
Rn_i	REAL	La valeur de R_i
Vali	BOOL	Résultat de validation

#### 🔧 L'algorithme de validation des modèles avec langage ST sous PL7

(\* programme de validation des modèles d'identification \*)

```

IF Star THEN
    IF RE %S6 THEN
        %MW2:= %MW2+1;
    END_IF;
    IF %MW2<128 THEN
        R_i:=1/128*( R_i + Erreur*Erreup);
        R_0:=1/128*( R_0 + Erreur*Erreur );
        Erreup:=Erreur;
    ELSE

```

```

        Rn_i:= R_i/R_0;

    END_IF;

    IF Rn_i <0.191 THEN

        Vali:=1;

    ELSE

        Vali:=0;

    END_IF;

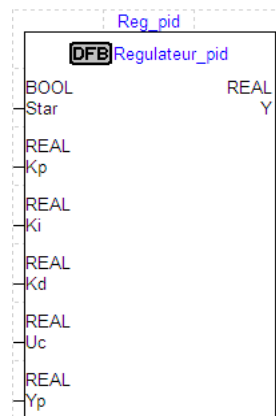
END_IF;

```

### ❖ Programme du régulateur PID

#### Description

Objet qui fait la régulation PID du système



Entrées	Format	Description
Star	BOOL	Démarre l'objet
Kp	REAL	paramètre proportionnelle
Ki	REAL	Temps d'intégration
Kd	REAL	Temps de dérivation
Uc	REAL	La consigne
Yp	REAL	Le grandeur à réglé (valeur de capteur)
Sorties	Format	Description
Y	REAL	La sortie du système après réglage



## Annexe B

---

(\*calculé des valeurs en pourcentage (étalonnage des capteurs) \*)

%MW69 :=(%MW77\*100)/10000;

%MF73:=%MF54;

%MF82:=INT\_TO\_REAL(%MW64)\*100.0/3750.0;

%MF86:=INT\_TO\_REAL(%MW77)\*100.0/10000.0;

(\* les adresses des touches fonction pour la supervision \*)

%M132:=%MW93:X0; %M133:=%MW93:X1; %MW150:X0:=%M134;

IF %M132 THEN

    SET %M134;

    RESET %MW93:X1;

END\_IF;

IF RE %M133 THEN

    RESET %M134;

    RESET %MW93:X0;

END\_IF;

IF %MW92=1 THEN

%MF45:=%MF45;

END\_IF;

IF %MW92=2 AND %M134 AND NOT %M133 THEN

%MF45:=%MF94\*100.0/10000.0;

END\_IF;

(\*la consign \*)

%MF0:=(%MF45\*3750.0)/100.0;

(\*l'algorithme du régulateur PID \*)

(\* les paramètres du régulateur \*)

%MF4:=28.0;       (\*kp\*)

%MF8:=1.0/300.0;   (\*ki\*)

%MF12:=2.0;       (\*kd\*)

%MF16:=%MF4+%MF8+%MF12;       (\*kpid\*)

(\* initialisation des valeurs \*)

%MF20:=INT\_TO\_REAL(%IW2.0);   (\*y initial\*)

%MF24:=0.0;       (\*yi initial\*)

%MF31:=0.0;       (\*e(i-1)\*)

%MF39:=%MF24 ;     (\*x=yi(i-1)\*)

## Annexe B

---

```
%MF35:=%MF0-%MF20;      (*e(i)*)
%MF54:=%MF16*%MF35+%MF39-%MF12*%MF31;      (*uc 'la commande PID' *)
IF %MF54>10000.0 THEN
%MF54:=10000.0;
END_IF;
IF %MF54<0.0 THEN
%MF54:=0.0;
END_IF;
IF %MW58=0 OR %MW58=3 THEN
%MW77:=REAL_TO_INT(%MF54);
ELSE
%MW77:=%MW59;
END_IF;
IF %MW58=1 THEN
%MW78:=REAL_TO_INT(%MF54);
ELSE
%MW78:=%MW60;
END_IF;
IF %MW58=2 THEN
%MW79:=REAL_TO_INT (%MF54);
ELSE
%MW79:=%MW61;
END_IF;
%MF20:=INT_TO_REAL(%IW2.0);  (*y(i) la sortie*)
%MF24:=%MF39+%MF8*%MF35;    (*yi(i)*)
%MF31:=%MF35;  (*e(i-1)*)
(* la fin de l'algorithme de régulateur PID *)
(* les sorties des actionneurs du système *)
%QW3.0:=%MW77;  (* adresse de variateur de vitesse pour commander l'électropompe *)
(* adresse des vannes proportionnels *)
%QW3.1:=%MW78; (*VM1*)
%QW3.2:=%MW79; (*VM2*)
%QW3.3:=%MW62; (*VM3*)
(*adresse des vannes TOR *)
```

## Annexe B

---

%Q5.0:=%M120; (\*EV1\*)

%Q5.1:=%M121; (\*EV2\*)

%Q5.2:=%M122; (\*EV3\*)

%Q5.3:=%M123; (\*EV4\*)

(\*commande de pompe de perturbations \*)

IF %QW3.3>2000 AND %QW3.0>500 AND %I4.0 THEN

SET %Q5.4;

ELSE

RESET %Q5.4;

END\_IF;

(\*adresses pour tracer les graphes sur Magelis\*)

%MW63:=REAL\_TO\_INT(%MF0); (\*la consign \*)

%MW64:=%IW2.0; (\*la sortie du système (niveau d'eau) \*)

%MW65:=REAL\_TO\_INT(%MF54); (\* la command Uc \*)

%MW66:=%IW2.3; (\* les perturbations \*)

📌 Programme d'enregistrement des mesures sous l'automate en langage ST sous  
PL7

(\* stockage des données \*)

IF RE %M150 AND %M170 AND %MW173<500

THEN INC %MW173;

%MW500[%MW173]:= %MW63; (\* la consign \*)

%MW1000[%MW173]:= %MW64; (\* la sortie du système y \*)

%MW1500[%MW173]:= %MW65; (\* la commande Uc \*)

%MW1500[%MW173]:= %MW66; (\* les perturbations \*)

END\_IF;

### Annexe C

#### Calcul de l'algorithme de l'identification MCR

$$\theta_N = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N$$

À l'instant suivant la nouvelle estimation est

$$\theta_{N+1} = (\Phi_{N+1}^T \Phi_{N+1})^{-1} \Phi_{N+1}^T Y_{N+1}$$

Avec

$$\Phi_{N+1} = \begin{bmatrix} \Phi_{N+1} \\ \Phi_N \end{bmatrix}; \quad Y_{N+1} = \begin{bmatrix} y_{N+1} \\ Y_N \end{bmatrix} \quad \text{et} \quad E_{N+1} = \begin{bmatrix} e_{N+1} \\ E_N \end{bmatrix}$$

$$\text{Ou } \varphi_{N+1} = [-y_N \quad -y_{N-1} \quad \dots \quad -y_{N-n+1} \quad u_{N+1} u_{N-p+1}]$$

$\theta_{N+1}$  s'écrit alors

$$\begin{aligned} \hat{\theta}_{N+1} &= (\Phi_{N+1}^T \Phi_{N+1})^{-1} \Phi_{N+1}^T Y_{N+1} \\ \hat{\theta}_{N+1} &= (\Phi_N^T \Phi_N + \varphi_{N+1}^T \varphi_{N+1})^{-1} (\Phi_N^T Y_N + \varphi_{N+1}^T y_{N+1}) \end{aligned}$$

En utilisant le lemme d'inversion matricielle

$$(Q + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

On obtient

$$\begin{aligned} \hat{\theta}_{N+1} &= ((\Phi_N^T \Phi_N)^{-1} - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T [1 \\ &\quad + \varphi_{N+1} (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T]^{-1} \varphi_{N+1} (\Phi_N^T \Phi_N)^{-1}) (\Phi_N^T Y_N + \varphi_{N+1}^T y_{N+1}) \end{aligned}$$

En posant  $a = 1 + \varphi_{N+1} (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T$

$$\begin{aligned} \hat{\theta}_{N+1} &= ((\Phi_N^T \Phi_N)^{-1} - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} \varphi_{N+1} (\Phi_N^T \Phi_N)^{-1}) (\Phi_N^T Y_N + \varphi_{N+1}^T y_{N+1}) \\ &= (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N + (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T y_{N+1} \\ &\quad - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} \varphi_{N+1} (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N \\ &\quad - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} \varphi_{N+1} (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T y_{N+1} \\ &= \hat{\theta}_N + (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T y_{N+1} - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} \varphi_{N+1} \hat{\theta}_N \\ &\quad - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} \varphi_{N+1} (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T y_{N+1} \\ &= \hat{\theta}_N - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} (\varphi_{N+1} \hat{\theta}_N) \\ &\quad + (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T (-a^{-1} \varphi_{N+1} (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T y_{N+1} + y_{N+1}) \end{aligned}$$

## Annexe C

---

$$\begin{aligned}
 &= \hat{\theta}_N - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} (\varphi_{N+1} \hat{\theta}_N) \\
 &\quad + (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} (-\varphi_{N+1} (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T + a) y_{N+1} \\
 &= \hat{\theta}_N - (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} (\varphi_{N+1} \hat{\theta}_N) + (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} y_{N+1} \\
 &\quad = \hat{\theta}_N + (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} (y_{N+1} - \varphi_{N+1} \hat{\theta}_N)
 \end{aligned}$$

On obtient finalement une formule de récurrence

$$\hat{\theta}_{N+1} = \hat{\theta}_N + (\Phi_N^T \Phi_N)^{-1} \varphi_{N+1}^T a^{-1} (y_{N+1} - \varphi_{N+1} \hat{\theta}_N)$$

Le terme  $(y_{N+1} - \varphi_{N+1} \hat{\theta}_N)$  représente l'erreur d'estimation à l'aide des paramètres précédents. L'équation est alors : la nouvelle estimation est l'ancienne estimation corrigée par un terme proportionnel à l'erreur d'estimation précédente, que l'on peut récrire sous la forme.

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1} (y_{N+1} - \varphi_{N+1} \hat{\theta}_N)$$

L'algorithme général est donc

$$\begin{aligned}
 \hat{\theta}_{N+1} &= \hat{\theta}_N + K_{N+1} (y_{N+1} - \varphi_{N+1} \hat{\theta}_N) \\
 K_{N+1} &= P_N \varphi_{N+1}^T (I + \varphi_{N+1} P_N \varphi_{N+1}^T)^{-1} \\
 P_{N+1} &= P_N - K_{N+1} \varphi_{N+1} P_N
 \end{aligned}$$