



République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد  
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي  
Institut de Maintenance et de Sécurité Industrielle

**Département de Maintenance en Instrumentation**

## **Mémoire de fin d'étude**

Pour l'obtention du diplôme de Master

**Filière :** Maintenance en Instrumentation

**Spécialité :** Maintenance des automatismes et instrumentation industrielle

### **Thème**

**Régulation de niveau et débit d'un système hydraulique par automate – SIEMENS S7-300 et supervision via WINCC flexible**

Préparé par :

**DJEROUROU Yacer**

et

**ELKHALDI Oussama**

Devant le jury composé de :

| <b>Nom et prénom</b>   | <b>Grade</b> | <b>Etablissement</b> | <b>Qualité</b>   |
|------------------------|--------------|----------------------|------------------|
| BENARBIA Taha          | MCB          | IMSI-Univ .D'Oran 2  | <b>Encadreur</b> |
| KEDDAR Mohamed         | MAA          | IMSI-Univ .D'Oran 2  | <b>Président</b> |
| BENFEKIR<br>Abderrahim | MCB          | IMSI-Univ .D'Oran 2  | <b>Examineur</b> |

**REMERCIEMENTS**

*Tout d'abord, nous remercions Dieu tout puissant de nous avoir donné le courage et la patience durant toutes ces années d'études, et que grâce à lui, ce travail a pu être réalisé.*

*Merci à nos parents d'avoir su nous écouter et nous motiver, à nos amis(e), qui ont été là pour nous pendant les périodes de doute et de stress.*

*Nous remercierons particulièrement notre encadreur Mr. BENARBIA et pour son aide précieuse, ses conseils et pour le temps qu'il nous a consacré tout au long de ce travail.*

*Nous remercierions aussi toute personne ayant contribué de près ou de loin à l'accomplissement de ce modeste travail.*

*Nous tenons aussi à remercier les membres du jury d'avoir accepté de juger notre travail*

**NOTATIONS ET SYMBOLES**

| <i>Symboles</i>      | <i>Désignations</i>                     |
|----------------------|---|
| <b>W(t)</b>          | consigne                                |
| <b>X(t)</b>          | Mesure                                  |
| <b>Y(t)</b>          | Commande                                |
| <b>E(t)</b>          | Erreur (W-X)                            |
| <b>PO</b>            | Partie opérative                        |
| <b>PC</b>            | Partie commande                         |
| <b>K<sub>p</sub></b> | Gain proportionnel                      |
| <b>T<sub>i</sub></b> | Temps d'intégration                     |
| <b>T<sub>d</sub></b> | Temps dérivatif                         |
| <b>DISC</b>          | Discontinue                             |
| <b>Z(t)</b>          | perturbation                            |
| <b>P</b>             | Régulateur proportionnel                |
| <b>PI</b>            | Régulateur proportionnel Intégral       |
| <b>IHM</b>           | Interface homme machine                 |
| <b>CPU</b>           | Centrale processing unite               |
| <b>IN</b>            | Input                                   |
| <b>OUT</b>           | Output                                  |
| <b>RAM</b>           | Random access memory                    |
| <b>ROM</b>           | Read only memory                        |
| <b>TOR</b>           | Tout ou rien                            |
| <b>PID</b>           | Régulateur proportionnel IntégralDérivé |
| <b>API</b>           | Automate programmable industriel        |
| <b>FBD</b>           | Fonction bloc diagramme                 |

**LISTE DES FIGURES &  
LISTE DES TABLEAUX**

**LISTE DES FIGURES & LISTE DES TABLEAUX**

|   |                               |
|---|-------------------------------|
| Figure I.1 Echange des informations PC avec l'extérieur .....                             | 4                             |
| Figure I.2 les vannes de la régulation .....  | 6                             |
| Figure I.3 Architecture interne des API .....   | 7                             |
| Figure I.4 Différents modèles de L'A.P. I SIEMENS S7 .....                                | 9                             |
| Figure I.5 Automate S7-1200. [6] .....  | Error! Bookmark not defined.0 |
| Figure I.6 Module extensions .....  | Error! Bookmark not defined.0 |
| Figure I.7 Automate s7-300 .....  | Error! Bookmark not defined.1 |
| Figure I.8 Schéma représentant les adresses de chaque module .....                        | Error! Bookmark not defined.3 |
| Figure I.9 Module d'entrées/sorties analogique (SM331/SM332).....                         | Error! Bookmark not defined.4 |
| Figure I.10 interface de TIAPORTAL V13 .....  | 16                            |
| Figure I.11 Présentation d'un schéma CONT .....   | 18                            |
| Figure I.12 exemple Langage littéral structuré .....                                      | Error! Bookmark not defined.0 |
| Figure II.1 Système de commande .....   | 23                            |
| Figure II.2 Boucle ouverte .....  | Error! Bookmark not defined.3 |
| Figure II.3 Boucle fermée .....   | Error! Bookmark not defined.4 |
| Figure II.4 Représentation fonctionnelle d'une boucle de régulation .....                 | Error! Bookmark not defined.4 |
| Figure II.5 Performances d'un système de commande .....                                   | 26                            |
| Figure II.5 stabilité du système .....  | 26                            |
| Figure II.6 Démonstration du mode de glissement .....                                     | 27                            |
| Figure II.7 Le principe de base de la commande prédictive à base de modèle .....          | 28                            |
| Figure II.8 Principe de la commande adaptative .....                                      | 28                            |
| Figure II.9 Schéma bloc d'une commande directe du couple .....                            | 29                            |
| Figure II.10 La commande optimal .....  | 29                            |
| Figure II.11 la variation de la consigne et la mesure .....                               | 30                            |
| Figure II.12 Réponse en boucle ouverte du procédé à un échelon de commande Ay .....       | Error! Bookmark not defined.1 |
| Figure II.13 Mise en oscillation juste entretenues d'un système .....                     | Error! Bookmark not defined.2 |
| Figure II.14 Réponse indicielle du régulateur P .....                                     | Error! Bookmark not defined.3 |
| Figure II.15 Influence du temps d'intégration I .....                                     | Error! Bookmark not defined.4 |
| Figure II.16 Influence du temps de dérivation D .....                                     | 35                            |
| Figure II.17 Pont de Wheatstone .....   | 36                            |
| Figure II.18 Représentation des ensembles .....   | 37                            |
| Figure II.19 Structure générale d'une commande floue .....                                | 43                            |
| Figure III.0 Configuration de notre bloc FB 41 .....                                      | Error! Bookmark not defined.5 |
| Figure III.1 Bloc de calcul de la partie décroissante de la fonction d'appartenance ..... | 45                            |
| Figure III.2 Bloc de calcul de La partie croissante de la fonction d'appartenance .....   | Error! Bookmark not defined.6 |
| Figure III.3 SIMATIC S7-300 en détail .....   | 47                            |
| Figure III.4 Creation de projet 1 .....   | 47                            |
| Figure III.5 Création de projet 2 .....   | 48                            |
| Figure III.6 Station SIMATIC 300 (1) .....  | 49                            |
| Figure III.7 Station Simatic 300 (2) .....  | 49                            |
| Figure III.8 Les modules entrées et sorties TOR .....                                     | 50                            |
| Figure III.9 Les modules entrées et sorties analogiques .....                             | 51                            |
| Figure III.10 Remplie dans la table mnémonique .....                                      | Error! Bookmark not defined.2 |
| Figure III.11 Bloc d'organisation (OB) .....  | Error! Bookmark not defined.  |
| Figure III.12 La fonction mise à l'échelle (SCALE) .....                                  | Error! Bookmark not defined.3 |
| Figure III.13 La fonction mise à l'échelle (UNSCALE) .....                                | Error! Bookmark not defined.4 |

|  |                                       |
|--|---------------------------------------|
| Figure III.14 Modules fonctionnels d'un système de supervision.....                    | 55                                    |
| Figure III.15 Communication entre WINCC et automate programmable industriel (API)..... | 56                                    |
| Figure III.16 Vue d'ensemble du progiciel Win cc flexible.....                         | 57                                    |
| Figure III.17 Schéma P&ID de la station de niveau.....                                 | 57                                    |
| Figure III.18 Station de niveau PUL-2.....   | 58                                    |
| Figure III.19 Programme à créer.....   | 58                                    |
| Figure III.20 Programmation sous step7.....  | 59                                    |
| Figure III.21 Programmation sous step7 - 1.....  | 59                                    |
| Figure III.22 Programmation sous step7 - 2.....  | 59                                    |
| Figure III.23 Programmation sous step7 - 3.....  | <b>Error! Bookmark not defined.</b> 1 |
| Figure III.24 Programmation sous step7 - 4.....  | <b>Error! Bookmark not defined.</b> 1 |
| Figure III.25 Schéma fonctionnel de bloc PID (FB41).....                               | 64                                    |
| Figure III.26 Paramétrage du régulateur sous step7 (FB41).....                         | 64                                    |
| Figure III.27 Bloc de données de (FB41).....   | 65                                    |
| Figure III.28 Table mnémonique de programme.....                                       | 66                                    |
| Figure III.29 Configuration des modules analogique.....                                | 66                                    |
| Figure III.30 Vue d'accueil du système de contrôle de processus PUL-2.....             | 67                                    |
| Figure III.31 Runtime du système de contrôle de processus PUL-2.....                   | 68                                    |

|  |            |
|--|------------|
| <b>Tableau I.1</b> les types de vannes .....   | <b>6</b>   |
| <b>Tableau II.1</b> fonction et instructions d'action de langage IL [12] .....       | <b>32</b>  |
| <b>Tableau II.2</b> Tableau pour le calcul des paramètres PID en boucle ouverte..... | <b>33</b>  |
| <b>Tableau II.3</b> Réglages préconisés par Ziegler et Nichols en boucle fermée..... | <b>5</b>   |
| <b>Tableau III.1</b> Influence des actions des régulateurs PID.....                  | <b>355</b> |
| <b>Tableau III.2</b> Description des différents éléments de l'API S7 300 .....       | <b>60</b>  |
| <b>Tableau III.3</b> Paramètres d'entrée .....                                       | <b>62</b>  |
| <b>Tableau III.4</b> Paramètres de sorties .....                                     | <b>63</b>  |





**SOMMAIRE**

## SOMMAIRE

|  |                               |
|--|-------------------------------|
| Introduction Générale .....  | Error! Bookmark not defined.  |
| <b>Chapitre 1 : Généralités sur les automates SIEMENS.</b> .....   | <b>2</b>                      |
| <b>I.1 Introduction :</b> .....  | <b>3</b>                      |
| <b>I.2 Structure d'un système automatisé :</b> .....   | <b>3</b>                      |
| <b>I.2.1 Partie commande :</b> .....   | <b>3</b>                      |
| <b>I.2.2 La partie supervision :</b> .....   | <b>3</b>                      |
| <b>I.2.3 La partie opérative :</b> .....   | <b>3</b>                      |
| <b>I.3 Automate programmable industriel (A.P.I)</b> .....  | <b>7</b>                      |
| <b>I.3.1 Architecture interne des API :</b> .....  | <b>7</b>                      |
| <b>I.3.2 Fonctionnement de l'API :</b> .....   | <b>8</b>                      |
| <b>I.3.3 Critères de choix d'un automate</b> .....   | <b>8</b>                      |
| <b>I.4 Les différents modèles de l'API SIEMENS S7</b> .....  | <b>9</b>                      |
| <b>I.4.1 Présentation de l'API SIEMENS S7-1200 :</b> .....   | Error! Bookmark not defined.0 |
| <b>I.4.2 Présentations de l'automate S7-300 :</b> .....  | Error! Bookmark not defined.1 |
| <b>I.5 Logiciel « Totally Interfrated Automation » :</b> .....   | <b>16</b>                     |
| <b>I.6 SIMATIC MANAGER – STEP7</b> .....   | <b>17</b>                     |
| <b>I.6.1 Fonctionnalités :</b> .....   | <b>17</b>                     |
| <b>I.6.2 Caractéristiques techniques :</b> .....   | <b>17</b>                     |
| <b>I.6.3 Avantage :</b> .....  | <b>17</b>                     |
| <b>I.7 Présentations des blocs de programmation :</b> .....  | <b>18</b>                     |
| <b>I.7.1 Les blocs d'organisation (OB)</b> .....   | <b>18</b>                     |
| <b>I.7.2 Les fonctions (FC) et blocs fonctionnels (FB)</b> .....   | <b>18</b>                     |
| <b>I.8 Langage de programmation :</b> .....  | <b>18</b>                     |
| <b>I.8.1 Ladder (LD) :</b> .....   | <b>18</b>                     |
| <b>I.8.2 Langage liste d'instruction (IL)</b> .....  | <b>19</b>                     |
| <b>I.8.3 Langage Grafcet :</b> .....   | <b>19</b>                     |
| <b>I.8.4 Langage littéral structuré (ST)</b> .....   | <b>20</b>                     |
| <b>2 Chapitre 2 : Étude détaillé sur les différente techniques et méthodologie de régulation et asservissement peuvent s'effectués par API</b> ..... | <b>21</b>                     |
| <b>II.1 Introduction</b> .....   | Error! Bookmark not defined.2 |
| <b>II.2 Conception d'un système de commande</b> .....  | Error! Bookmark not defined.2 |
| <b>II.2.1 Asservissement :</b> .....   | Error! Bookmark not defined.2 |
| <b>II.2.2 Régulation :</b> .....   | Error! Bookmark not defined.2 |
| <b>Dans un asservissement, la grandeur réglée devra suivre rapidement les variations de la consigne</b> .....  | Error! Bookmark not defined.2 |
| <b>II.3 Boucle ouverte et boucle fermée</b> .....  | Error! Bookmark not defined.3 |

|         |  |                               |
|---------|--|-------------------------------|
| II.3.1  | Boucle ouverte (B.O.).....   | Error! Bookmark not defined.3 |
| II.3.2  | Boucle fermée (B.F.).....  | Error! Bookmark not defined.3 |
| II.4    | Constituants d'une chaine de régulation.....   | Error! Bookmark not defined.4 |
| II.4.1  | Consigne « W ».....  | Error! Bookmark not defined.4 |
| II.4.2  | Sortie « Y ».....  | Error! Bookmark not defined.4 |
| II.4.3  | Retour « M ».....  | Error! Bookmark not defined.4 |
| II.4.4  | Perturbation « P ».....  | Error! Bookmark not defined.4 |
| II.4.5  | Comparateur .....  | Error! Bookmark not defined.4 |
| II.4.6  | Régulateur.....  | Error! Bookmark not defined.4 |
| II.4.7  | Actionneur .....   | Error! Bookmark not defined.4 |
| II.4.8  | Procédé.....   | 24                            |
| II.5    | Performances des systèmes réglés.....  | 25                            |
| II.5.1  | Rapidité .....   | 25                            |
| II.5.2  | Précision.....   | 25                            |
| II.5.3  | Stabilité .....  | 25                            |
| II.6    | Les commandes avancées .....   | 30                            |
| II.6.1  | Méthode de Ziegler Nichols en boucle fermée .....  | Error! Bookmark not defined.1 |
| II.6.2  | Régulateur à action proportionnelle (P).....   | Error! Bookmark not defined.3 |
| II.6.3  | Régulateur à action intégrale (I).....   | Error! Bookmark not defined.3 |
| II.6.4  | Régulateur dérivateur pur (D).....   | Error! Bookmark not defined.4 |
| II.7    | Les différentes techniques et méthodologie de régulation et asservissement peuvent s'effectués par API<br>35             |                               |
| II.7.1  | Régulation Tout Ou Rien .....  | 36                            |
| II.7.2  | La Régulation PID.....   | 36                            |
| II.7.3  | Régulation analogique .....  | 39                            |
| II.7.4  | Régulation numérique .....   | 39                            |
| II.7.5  | La logique floue .....   | 40                            |
| II.7.6  | Régulation par la logique floue .....  | Error! Bookmark not defined.  |
| II.7.7  | Principe de la commande floue .....  | Error! Bookmark not defined.1 |
| II.8    | Remarque.....  | Error! Bookmark not defined.2 |
| II.9    | Conclusion.....  | Error! Bookmark not defined.3 |
| 3       | Chapitre 3 : Proposition d'une régulation discontinue et régulation PID de débit et niveau d'un système hydraulique..... | Error! Bookmark not defined.4 |
| III.1   | Introduction.....  | 4Error! Bookmark not defined. |
| III.2   | Automate S7-300 (CPU314) .....   | 45                            |
| III.3   | Création d'un projet et configuration Matérielle dans STEP 7.....  | 46                            |
| III.3.1 | Création d'un projet dans SIMATIC Manager .....  | 46                            |
| III.3.2 | Configuration matériels(hardware).....   | 48                            |

---

|                     |  |                               |
|---------------------|--|-------------------------------|
| III.3.3             | Modules d'E/S tout ou rien (TOR) .....                                       | 48                            |
| III.3.4             | Modules d'E/S analogiques .....  | 49                            |
| III.3.5             | Création de la table des mnémoniques.....                                    | 49                            |
| III.3.6             | Les blocs de programmations.....   | Error! Bookmark not defined.0 |
| III.4               | Progiciel de conception et configuration d'IHM WINCC flexible.....           | 52                            |
| III.4.1             | Présentation générale.....   | 53                            |
| III.4.2             | Les principales étapes suivies pour créer notre application sous WINCC ..... | 54                            |
| III.4.3             | Communication entre WINCC et automate programmable industriel (API).....     | 54                            |
| III.4.4             | Eléments du progiciels WINCC flexible.....                                   | 54                            |
| III.4.5             | Création de l'interface WINCC .....  | 55                            |
| III.5               | Réalisation de projet : .....  | 56                            |
| III.5.1             | Création de programme de régulation discontinue.....                         | 56                            |
| III.6               | Création de programme de régulation PID .....                                | Error! Bookmark not defined.0 |
| III.6.1             | Programmation de la régulation PID en utilisant le bloc FB 41 intégré .....  | Error! Bookmark not defined.0 |
| III.6.2             | Fonctionnement de bloc PID (FB41) .....                                      | Error! Bookmark not defined.0 |
|                     | .....  | Error! Bookmark not defined.1 |
| III.6.3             | Paramétrage du régulateur PID .....  | 62                            |
| III.6.4             | Mise en œuvre d'un régulateur PID sous step7 .....                           | 63                            |
| III.6.5             | La programmation de régulation PID sous step7 .....                          | 64                            |
| III.7               | Configuration des modules analogique .....                                   | 65                            |
| III.7.1             | Module d'entrée/sorties analogique.....                                      | 66                            |
| III.8               | Création d'un vue dans Wincc Flexible .....                                  | 67                            |
| III.8.1             | WinCC flexible Runtime.....  | 67                            |
| III.8.2             | Interface de commande PID et Discontinue de la station PUL-2/EV .....        | 67                            |
| III.9               | Conclusion.....  | Error! Bookmark not defined.  |
| Conclusion Générale | .....  | 70                            |

## RESUME

Dans ce mémoire on va présenter des généralités sur les Automate SIMATIC Et donne une vision à quoi sert l'automatisme et comment fonctionne et exposer quelque modèles d'automate comme un exemple S7-1200

S7-300 a été présenté en générale avec ses caractéristiques, ses composants et son fonctionnement et ses performances. Par ailleurs, nous avons discuté l'utilisation des automates dans l'industrie, à titre exemple la régulation comme nous allons faire présentation en générale c'est quoi la régulation et comment fonctionne et aussi comment concevoir une chaine de régulation et aussi les équipements nécessaire pour attribue le bon procès.

Le but de ce mémoire est la réalisation d'un modèle de régulation PID et DISCONTINUE a un procès pour voir les performances et les résultats à l'aide des logiciels utilisé Step7 et WINCC FLEXIBLE.

En outre on va parler aussi de la configuration matériel des automates et de logiciel et les étapes à suivre pour faire la régulation d'un procédé pour donner à la fin la meilleure solution à l'industrie concernant ce sujet.

## ABSTRACT

In this memory we will present the generality on the SIMATIC PLC And give a vision of what the automation is for and how function either operative part or control part we will give some model of PLC as an example S7-1200

S7-300 is presented in general their characteristic and their components and how function and performance of each. Also we will talk about the use of this PLC in the industry we take the example of the regulation as we will make presentation in general what is the regulation and how function and also how Constituents of a chain of regulation and also the equipment necessary to attribute the good process.

The purpose of this memory is to make a PID regulation and DISCONTINUED has a trial to see the performances and the results using the software used Step7 and WINCC FLEXIBLE

We will also talk about the hardware configuration of the PLCs and software and the steps followed to make the regulation of a process to give at the end the best solution to the industry concerning this subject.

## ملخص

في هذه المذكرة ، سنقدم عمومية SIMATIC PLC ونقدم رؤية لما هو أنظمة التشغيل الآلي وكيف تعمل إما الجزء التشغيلي أو جزء

التحكم ، وسنقدم بعض نماذج PLC كمثال S7-1200 و S7-300 ي تم عرضها بشكل عام خصائصها ومكوناتها وكيفية وظيفة وأداء كل

منها. وسنتحدث أيضاً عن استخدام PLC في الصناعة ونأخذ مثال التنظيم حيث سنقدم عرضاً بشكل عام ما هو التنظيم وكيفية العمل وكذلك

كيفية مكونات سلسلة التنظيم وكذلك المعدات اللازمة لإسناد العملية الجيدة . الغرض من هذه المذكرة هو إجراء تجربة تنظيم PID و

" TOR " الرؤية الأداء والنتائج باستخدام البرنامج المستخدم STEP7 و WINCC FLEXIBLE سنتحدث أيضاً عن تكوين الأجهزة الخاصة ب PLCs والبرمجيات والخطوات المتبعة لجعل تنظيم عملية إعطاء في النهاية أفضل حل للصناعة فيما يتعلق بهذا الموضوع.

## INTRODUCTION GENERALE

L'automatisme fait partie des sciences de l'ingénieur les plus développées de nos jours. Elle a pour objectif principal de procurer une certaine autonomie aux systèmes les laissant prendre les décisions adéquates aux différents stades de leur évolution. Cette science peut être considérée comme étant le noyau de toutes les sciences de l'ingénieur et cela du fait, qu'elle est en permanente interaction avec les autres disciplines telle que l'informatique, l'électricité et la mécanique etc., chose qui lui a permis de faire de grands pas vers l'avant en bénéficiant du progrès de chacune de ces disciplines. L'automatisation des systèmes est une chose indispensable dans l'industrie moderne et cela du fait, qu'elle permet de :

- Réduire les frais de main d'œuvre ;
- Eviter les travaux dangereux et pénibles ;
- Assurer une meilleure qualité du produit ;
- Réaliser des opérations impossibles à contrôler manuellement ;
- Commander à distance : augmenter les performances du système de production ;
- Améliorer la sécurité de l'installation industrielle et du personnel.

Ces derniers temps, on assiste au phénomène de recours à l'automatisation des processus de la majorité des entreprises et cela après s'être rendu compte que l'automatisation constitue la réponse efficace à la rude compétition industrielle.

Notre travail consiste à présenter une étude sur la régulation de niveau et débit d'un système hydraulique par automate siemens s7-300 et supervision via **WINCC** flexible.

C'est dans ce contexte que cette idée de contribuer à la conception de ce système qui sera l'objet de notre projet de fin d'étude. Pour ce faire, nous avons décomposé notre travail en trois principales parties, chacune constitue un chapitre.

**Chapitre 1** : Etat de l'art sur les automate programmable siemens de S7-300 à s7- 1200 , avec une étude détaillée de tous ce qui est fonctionnement , composition, programmation, et logiciel SIMATIC Manager et Tia-portal.

**Chapitres 2** : Etude détaillée sur les différente techniques et méthodologie de régulation et asservissement peuvent s'effectué par API.

**Chapitre 3**: Proposition d'une régulation discontinue et régulation PID de débit et niveau d'un system hydraulique et comparaison entre les performances de chaque type de régulation. Simulation par **S7-300 SIMATIC** manager et développement d'une interface par Wincc flexible seront abordés en dernière phase.

Enfin, une conclusion générale qui couronne ce travail de mémoire.

**CHAPITRE I : GENERALITES SUR  
LES AUTOMATES SIEMENS.**



## **I.1- INTRODUCTION**

Durant ces dernières années, nous assistons à un développement énorme dans le monde industriel notamment dans les techniques de commande, et cela grâce à l'apparition de nouvelles techniques s'appuyant sur des systèmes très puissants et très flexibles : les automates programmables (API). Les API ont depuis leur apparition, poussé l'industrie à s'orienter vers une nouvelle stratégie de commande se basant sur des programmes informatiques (logique programmée) éclipsant ainsi les méthodes anciennes se basant sur des relais électromagnétiques et des systèmes pneumatiques pour la réalisation des parties de commande (logique câblée).

L'automate programmable industriel (A.P.I.). La partie commande assure le traitement logique des informations et échange des informations avec l'extérieur du système pilote et surveillant dont elle reçoit les consignes et à qui elle fournit des comptes rendus visuels ou sonores.

## **I.2- STRUCTURE D'UN SYSTEME AUTOMATISE :**

### **I.2.1- PARTIE COMMANDE**

La partie commande : PC La Partie Commande regroupe l'ensemble des composants permettant le traitement des informations reçues de la PO et des ordres envoyés par la PR. Elle :

- \*/ Elabore les ordres à partir de ces informations et informe l'opérateur de l'état du système ;
- \*/ Pilote le fonctionnement du système automatisé. Ses autres fonctions sont :

- \*\*/ Dialoguer avec l'opérateur via la partie relation ;
- \*\*/ Acquérir des données en provenance des capteurs de la partie opérative.
- \*\*/ Traiter les données acquises en entrée:

\*\*\*/ Surveillance (sûreté de fonctionnement, diagnostic.) ;

\*\*\*/ Régulation du comportement des systèmes de la partie opérative. Une fois que la PC a terminé son traitement, elle transmet des ordres aux pré-actionneurs de la PO et des informations aux composants de signalisation de la PR (la partie relation /supervision)[4]

### **I.2.2- PARTIE SUPERVISION**

La Partie Relation/Supervision (PS) est l'interface homme machine (IHM). Elle est équipée d'organes permettant : La mise en/hors énergie de l'installation, La sélection des modes de marche, La commande manuelle des actionneurs, Le départ des cycles de fonctionnement, L'arrêt d'urgence, D'informer l'opérateur de l'état de l'installation : voyants lumineux, afficheurs, écrans vidéos, klaxons [4].

### **I.2.3- PARTIE OPERATIVE**

La Partie Opérative (PO) agit sur la matière à partir d'ordres envoyés par la partie commande, et renvoie à cette dernière des informations sur son état ou sur l'environnement. Sa fonction globale est d'apporter de la valeur ajoutée à la matière première [4].

Les principales fonctions assurées par la partie opérative sont :

- Convertir un ordre de la partie commande en signaux exploitables pour la partie opérative ;
- Adapter le niveau de puissance pour transmettre l'ordre au système concerné dans la partie opérative ;
- Transformer la puissance fournie pour créer une action dans la partie opérative ;
- Agir sur la matière première pour donner de la valeur ajoutée ;
- Donner un retour d'information de l'action réalisée dans la partie opérative vers la partie relation.

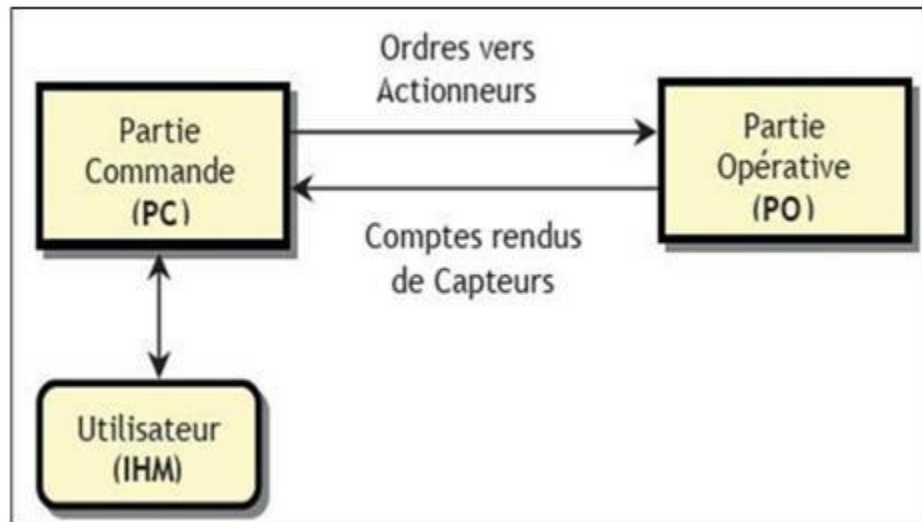


Figure I.1 : Echange des informations PC avec l'extérieur

### I.2.3.1- ACTIONNEURS

Les actionneurs sont des convertisseurs électromécaniques conçus pour mettre en mouvement des systèmes mécanique à partir de commande électrique. Ce sont essentiellement des moteurs et des vérins. Ils produisent de l'énergie mécanique à partir d'énergie électrique, hydraulique ou pneumatique mais toujours contrôlés par des signaux de commande électrique.[2]

Il y'a plusieurs types d'actionneurs, comme par exemple les :

**1- ACTIONNEURS PNEUMATIQUES :** Les actionneurs pneumatiques convertissent l'énergie de puissance pneumatique en énergie mécanique de translation, ils sont utilisés principalement pour des mouvements séquentiels simples [2].

**2- ACTIONNEURS HYDRAULIQUES :** Les actionneurs hydrauliques constituent l'outil indispensable pour convertir l'énergie hydraulique en énergie mécanique. Ils sont utilisés pour des mouvements requérant des forces très élevées à faible vitesse [1].

**3- ACTIONNEURS ELECTRIQUES :** Ils utilisent directement l'énergie électrique distribuée sur les machines. Ils peuvent être sous plusieurs formes : moteur électrique, électrovanne de débit, résistance de chauffage, tête de soudure, ... Les pré-actionneurs associés à ces actionneurs électriques sont principalement les contacteurs et variateurs de vitesse, entourés des sécurités nécessaires [1].

### I.2.3.2- CAPTEURS

Le capteur est un organe qui prélève des informations sur le comportement de la partie opérative et les transforme en une information exploitable par la partie commande. Une

information est une grandeur abstraite qui précise un événement particulier parmi un ensemble d'événements possibles [2]. Pour pouvoir être traitée, cette information sera portée par un support physique (énergie).

Les signaux sont généralement de nature électrique ou pneumatique.

Les différents capteurs rencontrés sont :

**1- CAPTEURS PASSIFS** : Ils ont besoin dans la plupart des cas d'apport d'énergie extérieure pour fonctionner (ex.: thermistance, photorésistance, potentiomètre, jauge d'extensomètre appelée aussi jauge de contrainte). Ce sont des capteurs modélisables par une impédance. Une variation du phénomène physique étudié (mesuré) engendre une variation de l'impédance. Il faut leur appliquer une tension pour obtenir un signal de sortie

**2- CAPTEURS ACTIFS** : On parle de capteur actif lorsque le phénomène physique, qui est utilisé pour la détermination de la mesure, effectue directement la transformation en grandeur électrique. C'est la loi physique elle-même qui relie mesure et grandeur électrique de sortie. Un capteur actif fonctionne assez souvent en électromoteur et dans ce cas, la grandeur de sortie est une différence de potentiel. Le nombre des lois physiques permettant une telle transformation est évidemment limité, on peut donc recenser facilement les capteurs actifs (dont le nombre est fini). Toutefois, les domaines d'application sont eux très étendus [2].

### I.2.3.3- TYPES DE SORTIE

Les capteurs et leurs conditionneurs peuvent aussi faire l'objet d'une classification par types de sortie :

**1- CAPTEURS ANALOGIQUES** : La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues.

**2- CAPTEURS NUMERIQUES** : La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre un grand nombre de valeurs discrètes.

**3- CAPTEURS LOGIQUES** : Ou capteurs TOR. La sortie est un état logique que l'on note 1 ou 0. La sortie peut prendre ces deux valeurs. Caractéristiques du capteur :

\*/ L'étendue de mesure.

\*/ Rapidité.

\*/ Résolution et la Sensibilité

### I.2.3.4- VANNES

La vanne de régulation est utilisée comme organe de réglage dans différents types de boucles de régulation. Elle permet de contrôler le débit dans une canalisation, en modifiant les pertes de charges de celle-ci. Il existe plusieurs représentations d'une vanne.



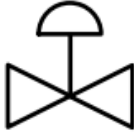
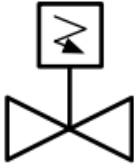

| Vanne simple  | Vanne manuelle  | Vanne Pneumatique   | Electrovanne   | Vanne pneumatique avec positionneur   |
|---|---|---|--|---|
|  |  |  |  |  |

Figure I.2 : les vannes de la régulation

Les principaux éléments constituant la vanne de régulation sont :

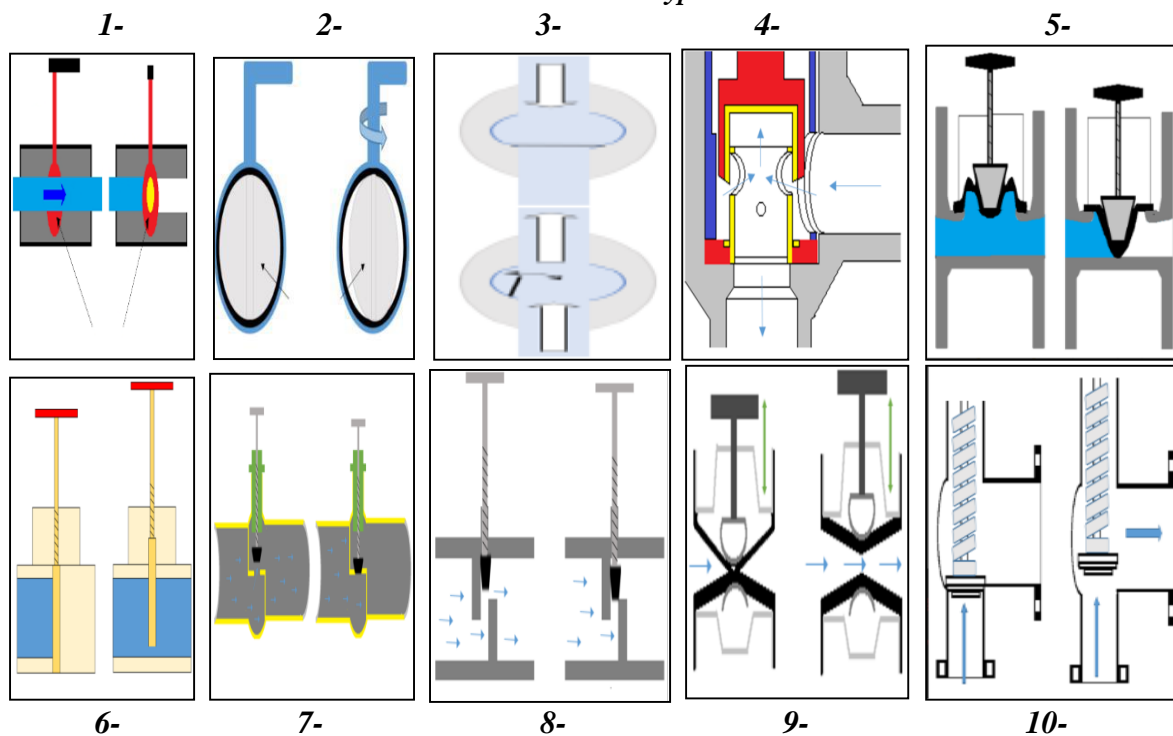
- Le **corps de vanne** : C'est l'élément qui assure le réglage du débit.
- Le **servomoteur** : C'est l'élément qui assure la conversion du signal de commande en mouvement de la vanne [3].

Et aussi d'un certain nombre d'éléments auxiliaires:

- Un contacteur de début et de fin de course ;
- Une recopie de la position ;
- Un filtre détendeur ;
- Un positionneur : il règle l'ouverture de la vanne en accord avec le signal de commande.

Les différents types de vannes :

Tableau I.1 : les types de vannes



1- La vanne à boisseau sphérique ; 2- La vanne à papillon ; 3- La vanne à clapet anti-retour ; 4- Les vannes d'étranglement ; 5- La vanne à diaphragme ; 6- La vanne à clapet ; 7- La vanne à soupape ; 8- Les vannes à pointeau ; 9- La vanne à pincement ; 10- La vanne de sûreté

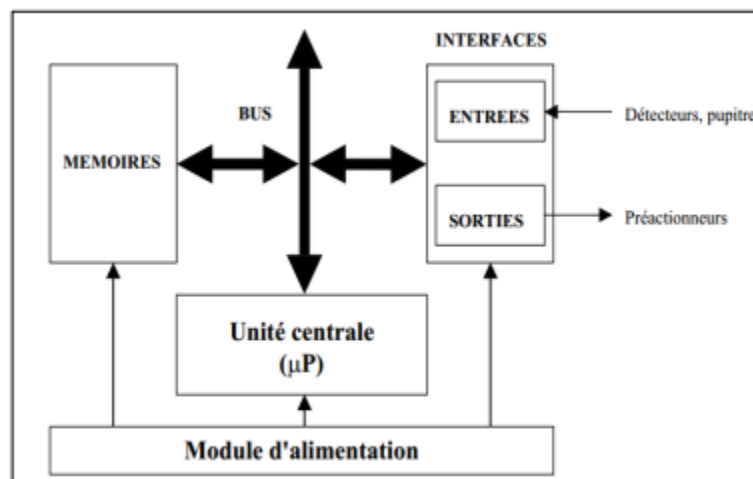
### I.3- AUTOMATE PROGRAMMABLE INDUSTRIEL (A.P.I)

**Par définition**, un automate programmable (A.P.I) est un appareil dédié au contrôle d'une machine ou d'un processus adapté à l'environnement industriel, constitué de composants électroniques,

Comportant une mémoire programmable par un utilisateur. L'API réalise des fonctions d'automatisme en étant directement connecté aux capteurs et pré- actionneurs afin d'assurer leur commande à partir d'informations logiques, analogiques ou numériques. L'API est programmable par un personnel qualifié et pas forcément informaticien et est destiné à piloter en temps réel des procédés industriels. A l'heure actuelle, l'API fait partie intégrante des procédés de fabrication modernes, il en est le « cerveau ». Le technicien est amené à concevoir, maintenir et dépanner ces automatismes industriels. [4].

#### I.3.1- ARCHITECTURE INTERNE DES API

L'architecture interne d'un API obéit au schéma donné sur la **Figure (I.3)** ci-dessous :



*Figure I.3 Architecture interne des API*

Détaillons successivement chacun des composants qui apparaissent sur ce schéma :

❖ **PROCESSEUR** : Constitue le cœur de l'appareil dans l'unité centrale. En fait, un processeur devant être automatisé, se subdivise en une multitude de domaines et processeurs partiels plus petits, liés les uns aux autres.

❖ **MODULES INTERFACE E/S** : Assurent le rôle d'interface entre le CPU et le processeur, en récupérant les informations sur l'état de ce dernier et en coordonnant les actions. Plusieurs types de modules sont disponibles sur le marché selon l'utilisation souhaitée : Modules TOR (Tout Ou Rien): l'information traitée ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type d'information délivrée par une cellule Photoélectrique, un bouton.

❖ **MEMOIRES** : Un système de processeur est accompagné par un ou plusieurs types de mémoires. Elles permettent de stocker :

- Le système d'exploitation dans des ROM ou PROM ;
- Le programme dans des EEPROM ;
- Les données système lors du fonctionnement dans des RAM. Cette dernière est généralement secourue par pile ou batterie

❖ **ALIMENTATION** : Assure la distribution d'énergie aux différents modules. L'automate est alimenté généralement par le réseau monophasé 230V-50 Hz. Mais d'autres alimentations sont possibles (120 AC ,110 AC etc. ). L'alimentation L : 24 V et 220V. La masse M : 0 V pour plus de précision sur les schémas de raccordements.[5]

❖ **LIAISONS DE COMMUNICATIONS** : Permettent la communication de l'ensemble des blocs de l'automate et des éventuelles extensions. Les liaisons s'effectuent avec :

- L'extérieur, par des bornes sur lesquels arrivent des câbles transportant le signal électrique.
- L'intérieur par des bus reliant divers éléments, afin d'échanger des données, des états et des adresses.

### 1.3.2- FONCTIONNEMENT DE L'API

L'automate programmable reçoit les informations relatives au système. Il traite ces informations en fonction du jeu d'instruction et modifie l'état de ses informations pour commander les prés actionneur.

- **Réception** : Nécessaire pour l'information d'entrées.
- **Traitement** : Notion de programme et de microprocesseur.
- **Jeu d'instructions** : Notion de stockage et de mémoire.
- **Commander** : Notion de sortie pour donner des ordres.

Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique).

On appelle scrutation, l'ensemble des quatre opérations réalisées par l'automate et le temps de scrutation est le temps mis par l'automate pour traiter la même partie de programme. Ce temps est de l'ordre de la dizaine de millisecondes pour les applications standards.

Le Temps de Réponse Total (TRT) est le temps qui s'écoule entre le changement d'état d'une entrée et le changement d'état de la sortie correspondant.

### 1.3.3 CRITERES DE CHOIX D'UN AUTOMATE

Le choix d'un automate programmable est en premier lieu le choix du :

**1- Nombre d'entrées / sorties** : Le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé.

**2- Type de processeur** : La taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettent le choix dans une large gamme.

**3- Fonctions ou modules spéciaux** : Certaines cartes (commande d'axe, pesage ...) permettent de soulager le processeur et doivent offrir les caractéristiques souhaitées (résolution, etc.).

**4- Fonctions de communication** : L'automate doit pouvoir communiquer avec les autres systèmes de commande et offrir des possibilités de communication avec des standards normalisés (Profibus) .[4]

#### I.4- DIFFERENTS MODELES DE L'API SIEMENS S7

Il existe différents types d'automates distingués principalement par leur forme. Ils sont illustrés dans la Figure I.4 ci-dessous :



**Figure I. 4 Différents modèles de L'A.P. I SIEMENS S7**

Le contrôleur S7-1200 (Figure I.4) offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils afin de répondre aux besoins en matière d'automatisation. Sa forme compacte, sa configuration souple et son important jeu d'instructions forment une solution idéale pour commander une variété importante d'applications [6].

Le CPU combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant. Une fois le programme chargé, le CPU contient la donnée logique nécessaire au contrôle et à la commande des appareils concernant le processus à contrôler.

Le CPU surveille les entrées et modifie-les sorties conformément à la logique du programme intégré, qui peut contenir des instructions booléennes, des instructions de comptage, des instructions de temporisation, des instructions mathématiques complexes ainsi que des commandes pour communiquer avec d'autres appareils intelligents.

Le CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les réseaux PROFIBUS, RS485 ou RS232 [6].

PROFINET est le nouveau standard de communication créé par PROFIBUS International pour mettre en œuvre des solutions d'automatisation intégrées et cohérentes, sur Ethernet industriel.

➤ **Modules d'extensions :**

La gamme S7-1200 offre divers modules de cartes enfichables pour accroître les capacités de la CPU avec des E/S supplémentaires ou d'autres protocoles de communication (Figure 6) [7].

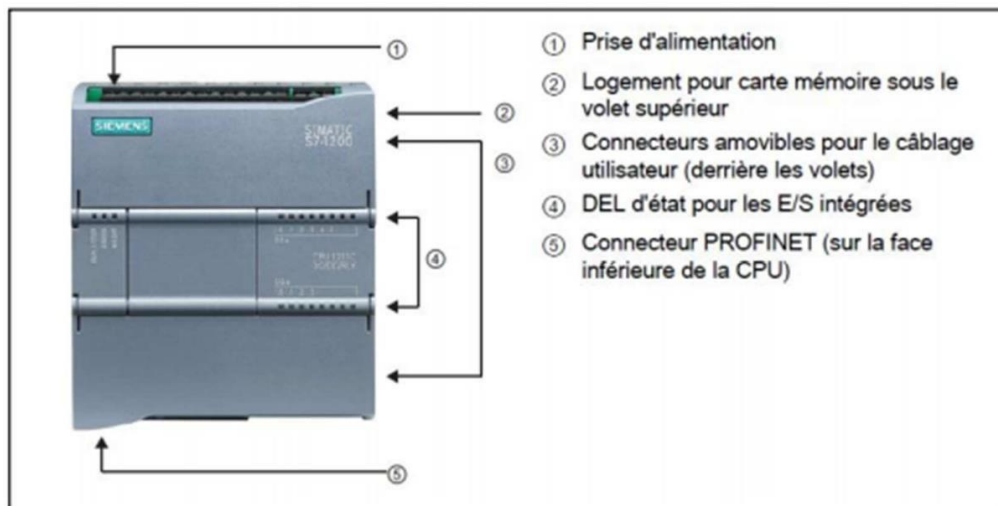


Figure I.5 Automate S7-1200. [6]

Il faudrait se référer aux caractéristiques techniques pour des informations détaillées sur un module spécifique :

- 1 : Module communication ;
- 2 : CPU ;
- 3 : Module d'E/S ;
- 4: Signal Board (SB), Communication Board.

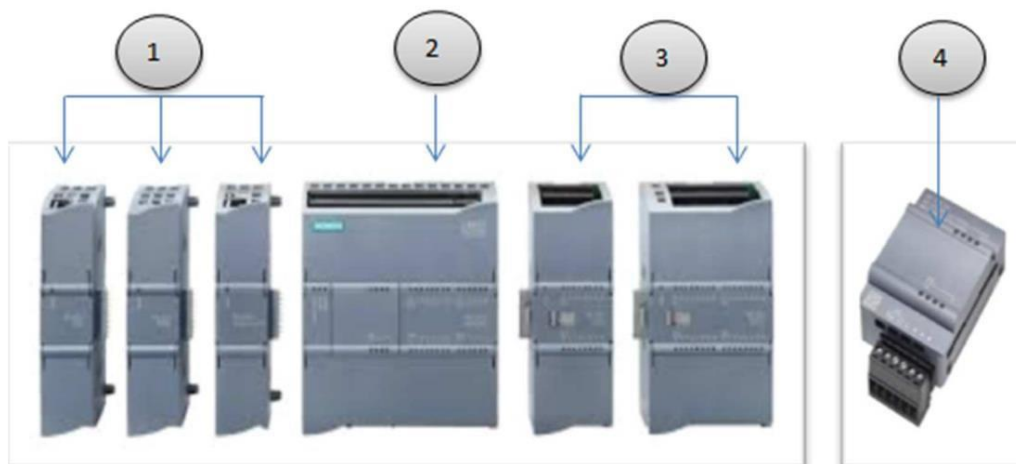


Figure I.6 : Module extensions



Pour un API S7-1200 on distingue :[6]

- Nombre de module d'extensions E/S : 8 ;
- Nombre Maximal des E/S : 16384 ;
- Nombre des modules de communications : 3 ;
- Module des signaux(SB).

#### I.4.2- PRESENTATIONS DE L'AUTOMATE S7-300 [8]

L'automate programmable industriel S7 – 300 fabriqué par SIEMENS, qui fait partie de La gamme SIMATIC S7 est un automate destiné à des tâches d'automatisation moyennes et hautes gammes. La configuration et le jeu d'instruction des API SIEMENS sont choisis pour satisfaire les exigences typiques et industrielles et la capacité d'extension variable permet une adaptation facile de l'appareil à la tâche considérée. L'automate lui-même est constitué d'une configuration minimale composée d'un module d'alimentation, de la CPU, du coupleur et de modules d'entrées/sorties.

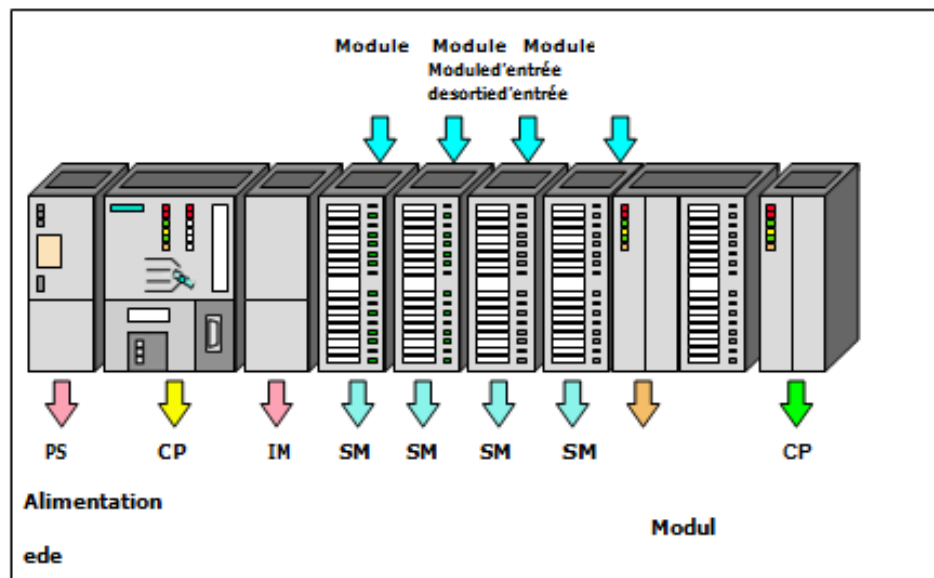


Figure 7 Automate S7-300

##### A) Modularité

Le S7 – 300 est de conception modulaire, une vaste gamme de module est disponible. Ces modules peuvent être combinés selon les besoins lors de la conception d'une solution d'automatisation. Les types des modules sont les suivants :

1. Modules d'alimentations (PS).
2. Unité centrale (CPU).
3. Coupleurs (IM).
4. Processeurs de communication (CP).
5. Modules de fonctionnements (FM).
6. Modules de signaux (SM).
7. Modules de simulation (SM374).

**B) Module d'alimentation**

Le module d'alimentation assure la conversion de tension du secteur (ou du réseau) en tension de (24V, 48V, 120V ou 230V) pour l'alimentation de l'automate et des capteurs et actionneurs en (24V, 48V, 120V ou 230V).

Il remplit aussi des fonctions de surveillance et signalisation à l'aide des LEDS. Il permet de sauvegarder le contenu des mémoires RAM au moyen d'une pile de sauvegarde ou d'une alimentation externe.

**C) Unités centrales (CPU)**

La CPU est le cerveau de l'automate car elle permet de:

- Lire les états des signaux d'entrées.
- Exécuter le programme utilisateur et commander les sorties.
- Régler le comportement au démarrage et diagnostiquer les défauts par les LEDS.
- Le S7-300 dispose d'une large gamme de CPU à différents niveaux de performance, on compte les versions suivantes :

- **CPU à utilisateur standard** : CPU313, CPU314, CPU315 et CPU316.

- **CPU avec fonction s'intégrées** : CPU312IFM et la CPU314 IFM.

- Les fonctions intégrées permettent d'automatiser à moindre coût des tâches qui ne nécessitent pas la performance d'un module de fonction (FM), la particularité de ces CPU est qu'elles sont dotées d'entrées /sorties TOR intégrées, des EEPROM intégrées et des fonctions s'intégrées qui sont les suivantes:

- Fonction intégrée fréquence mètre.
- Fonction intégrée compteur.
- Fonction intégrée compteur A/B.

**CPU avec interface PROFIL BUS DP (CPU315-2DP, CPU316-2DP, CPU318-2DP)[10]**

Elles sont utilisées pour la mise en place des réseaux, toutes ces CPU peuvent être utilisées comme maître DP ou esclave DP à l'exception de la CPU 318 - 2 DP où elle est utilisée uniquement comme un maître DP.

**D) Coupleur (IM)**

Les coupleurs sont des cartes électroniques qui assurent la communication entre les E/S (périphériques ou autre) et l'unité centrale. L'échange de l'information entre la CPU et les modules d'E/S s'effectue par l'intermédiaire d'un bus interne (liaison parallèle codée). Les coupleurs ont pour rôle le raccordement d'un ou plusieurs châssis au châssis de base.

Pour l'APIS7- 300, les coupleurs disponibles sont :

- IM365: Pour les couplages entre les châssis d'un mètre de distance au max ;
- IM360 et IM361: pour les couplages allant jusqu'à 10 mètres de distances.

### E) Modules de communication (CP)

Les modules de communication sont destinés aux tâches de communication par transmission en série. Ils permettent d'établir également des liaisons point à point avec :

- Des commandes robots.
- Communication avec des pupitres opérateurs.
- Des automates SIMATIC S7, SIMATIC S5 et des automates d'autres constructeurs.

### F) Modules de fonctions (FM)

Ces modules réduisent la charge de traitement de la CPU en assurant des tâches lourdes des calculs. On peut citer les modules suivants :

- FM354 et FM 357: Module de commande d'axe pour servomoteur.
- FM353 : Module de positionnement pour moteur pas à pas.
- FM355 : Module de régulation.
- FM350-1 et FM350 -2: Module de comptage.

### G) Modules de signaux (SM)

Ils servent d'interface entre le processus et l'automate. Ils existent des modules d'entrées TOR, des modules de sorties TOR ainsi que des modules d'entrées analogiques et des modules de sorties analogique. Les modules d'entrées/sorties sont des interfaces vers les capteurs et les actionneurs d'une machine ou d'une installation. Il existe différents modules d'entrées/sorties dont.

### H) Modules d'E/S TOR (SM321/SM322)[11]

Les modules d'entrées/sorties TOR constitue les interfaces d'entrées et de sorties pour les signaux toutou rien (TOR) de l'automate. Ces modules permettent de raccorder à l'automate S7 300 des capteurs et des actionneurs TOR les plus divers. En utilisant si nécessaire des équipements d'adaptation (conditionnement, conversion, etc.). Les modules d'entrées ramènent le niveau des signaux TOR externes, issus des capteurs, au niveau du signal interne du S7 - 300. Les modules de sorties transportent le niveau du signal interne du S7 - 300 au niveau du signal requis par les actionneurs ou Pré- actionneurs.

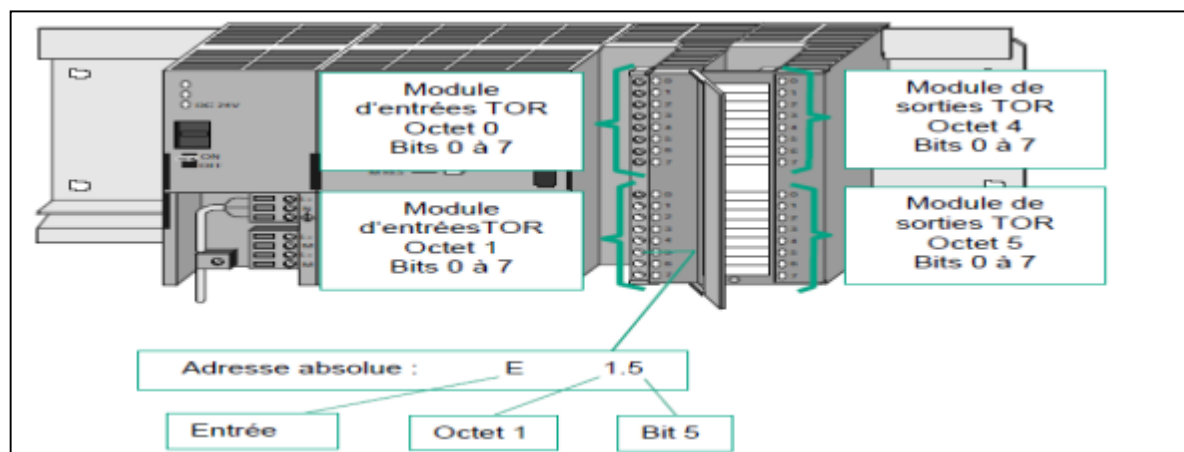


Figure I.8 : Schéma représentant les adresses de chaque module

**I- Modules d'E/S analogique (SM321/SM322) [10]**

Ces modules permettent de raccorder à l'automate des capteurs et actionneurs analogiques, les entrées et les sorties analogiques possèdent des convertisseurs analogique – numérique et numérique – analogiques d'une résolution de 11 bits + bit de signe, les images des valeurs analogiques sont représentées sur 16bits, pour les modules possédant néré solution inférieure à 15bits + bit de signe, les bits de poids plus faibles ne sont pas représentatifs.

Les modules d'entrées analogiques (SM 331) convertissent un signal analogique issu des capteurs analogiques en un signal numérique. Ces modules ne comportent qu'un circuit de conversion analogique numérique CAN car la CPU de l'automate S7-300 ne peut lire que les valeurs analogiques binaires.

La conversion analogique/numérique concerne les entrées analogiques des étendues de tension ( $\pm 80\text{mV}$ ,  $\pm 250\text{ mV}$ ,  $\pm 500\text{ mV}$ ,  $\pm 5\text{ V}$ ,  $\pm 10\text{ V}$ ), de courant ( $\pm 10\text{ mA}$ ,  $\pm 3,2\text{ mA}$ ,  $\pm 20\text{ mA}$ ), de résistance ( $150\ \Omega$ ,  $300\ \Omega$  et  $600\ \Omega$ ) et de température, en un mot (de compléments à 2) au format de 8, 12 ou 16 bits, afin que l'automate puisse traiter ces variables par de la programmation (gestion d'alarme, contrôle, asservissement etc.).

Les modules de sorties analogiques (SM 332) réalisent la conversion des signaux numériques internes (du S7-300) en signaux analogiques destinés aux actionneurs ou pré actionneurs analogiques, ces modules comportent donc des circuits de convertisseurs numériques analogiques CNA. Le transfert des valeurs numériques vers le module s'effectue par multiplexage piloté par le processeur automate. La conversion des voies de sorties analogiques est réalisée séquentiellement, c'est-à-dire que les voies de sorties analogiques sont converties les unes après les autres.



Figure I.9 : Module d'entrées/sorties analogique (SM331/SM332)

**J- Modules de simulation (SM374)[11]**

Le module de simulation SM 374 est un module spécial qui offre à l'utilisateur la possibilité de tester son programme lors de la mise en service en cours de fonctionnement. Dans le S7- 300, ce module se monte à la place d'un module d'entrée ou de sortie TOR. Il assure plusieurs fonctions telles que la simulation des signaux de capteurs aux moyens d'interrupteurs. La signalisation d'état des signaux de sorties par des LEDS.

**K- Extension et mise en réseau de sous-réseaux [14]**

Conformément aux différentes exigences des niveaux d'automatisation (niveaux de conduite, de cellules, de terrain et de capteurs/actionneurs), SIMATIC offre les sous-réseaux suivants :

- Interface multipoint (MPI).
- PROFIBUS.
- Couplage point à point.
- Industriel Ethernet.
- Interface actionneur/capteur (ASI).

Cependant, il existe des modules où ils sont à la fois des modules d'entrées et sorties analogiques (FM334), ces modules réalisent les deux fonctions.

**L- Interface multipoint (MPI)**

MPI est un sous-réseau comportant une étendue réduite et un faible nombre de partenaires pour les niveaux de terrain et de cellule. MPI est une interface multipoint située dans le SIMATIC S7/M7 et C7. Elle est conçue comme interface PG et est destinée à la mise en réseau de quelques CPU ou à l'échange de faibles quantités de données avec les PG. MPI conserve toujours le dernier paramétrage concernant la vitesse de transmission, le nombre de partenaires et l'adresse MPI maximum, même après un effacement général, une défaillance de tension ou un effacement du paramétrage CPU.

**M- Profibus**

1. En tant que bus de terrain PROFIBUS-DP prévu pour un échange de données rapide et cyclique et PROFIBUS-PA pour le domaine à sécurité intrinsèque.

2. Plage de cellules en tant que PROFIBUS (FDL ou PROFIBUS-FMS) pour la transmission rapide avec des partenaires de la communication disposant des mêmes droits. Cependant, vous pouvez également réaliser PROFIBUS-DP et PROFIBUS-FMS au moyen des processeurs de communication (CP).

**N- Industriel Ethernet**

Réalisation par processeurs de communication (CP). Industriel Ethernet est le réseau prévu pour le niveau de conduite et de cellules dans le système de communication du SIMATIC, ouvert et indépendant du fabricant.

L'Industriel Ethernet est conçu pour une transmission rapide en cas d'importantes quantités de données et permet une mise en réseau entre différents sites grâce à des passerelles. Pour les CPU décrites dans le présent manuel, vous ne pouvez réaliser une liaison à L'Industriel Ethernet que via des processeurs de communication.

**O- Interface capteur/actionneur (ASI)**

Réalisation par processeurs de communication (CP). L'interface AS ou capteur/actionneur est un système de sous-réseau prévu pour le niveau de processus le plus bas dans les installations d'automatisation. Elle sert en particulier à mettre en réseau les capteurs et actionneurs binaires. La quantité de données est de 4 bits maximum par station esclave. Vous pouvez réaliser un

raccordement à l'interface capteur/actionneur uniquement au moyen de processeurs de communication dans le cas d'une CPU S7-300.

### **P- Montage identique MPI et PROFIBUS-DP**

Pour le montage d'un réseau MPI, nous vous recommandons d'utiliser les mêmes composants de réseau que pour le montage d'un réseau PROFIBUS-DP. Les mêmes règles de montage sont appliquées.

### **I.5- LOGICIEL « TOTALLY INTEGRATED AUTOMATION »**

TIA Portal ou Totally integrated automation est un environnement de développement, tout en un permettant de programmer non seulement des automates mais aussi des afficheurs industriels (HMI).[9] Le TIA Portal contient le Step7 (permettant la programmation d'automate) et le Wincc (permettant de programmer des afficheurs Siemens). Il intègre aussi la gestion des fonctionnalités motion, comptage. Step7 fait partie de la suite TIA Portal et permet seulement de programmer des automates. C'est comme la suite Office de Microsoft qui est constituée de Word, Excel, Powerpoint etc. Par comparaison on peut dire que TIA Portal est la suite Office et Word est Step7. Le souhait de SIEMENS est d'intégrer toutes leurs gammes de produits dans un seul logiciel [9].



*Figure I.10 : Interface de TIAPORTAL V13*

#### **➤ LE WINCC :**

Le SIMATIC WinCC dans le Totally Integrated Automation Portal (TIA Portal) fait partie d'un nouveau concept d'ingénierie intégré qui offre un environnement homogène pour la programmation et la configuration de solutions de commande, de visualisation et d'entraînement. Ce Framework d'ingénierie est une avancée fondamentale dans le développement de logiciels et représente le développement continu et conséquent de la philosophie TIA. WinCC dans le TIA Portal est le logiciel pour toutes les applications IHM allant de solutions de commande simples avec des Basic Panels aux applications SCADA pour systèmes multipostes basés sur PC. [10]

## I.6- SIMATIC MANAGER - STEP7 [11]

### 1.6.1 FONCTIONNALITES

L'atelier logiciel STEP 7 Professional fédère toutes les fonctions requises pour concevoir, configurer, programmer, tester, mettre en service et maintenir les systèmes d'automatisation SIMATIC.

STEP 7 Professional apporte une productivité de l'ingénierie sans précédent. Il concrétise tous les atouts du concept Totally Integrated Automation Interface utilisateur orientée objet, commune à tous les outils logiciels SIMATIC Industriel Software Base de données unique des projets garantissant la cohérence des applications même complexes Cohérence des réseaux de communication entre les composants d'une application

### 1.6.2 CARACTERISTIQUES TECHNIQUES

STEP 7 Professional intègre en particulier les outils suivants :

- Interface utilisateur SIMATIC Manager commune à tous les outils logiciels intégrés et optionnels Tous les langages de programmation pour automates programmables définis dans le standard CEI 61131-3: schémas contact, logigrammes, listes d'instructions, graphes séquentiels (S7-GRAPH) et langages structuré (S7-SCL). [12].

- Le logiciel de simulation automate S7-PLCSIM pour la mise au point de programmes sans disposer des automates cible outil configuration graphique des composants matériels et des réseaux de communication STEP 7 Professional comporte de nombreuses fonctions évoluées qui contribuent à l'efficacité du travail des automaticiens, par exemple:

- Programmation structurée, avec des blocs fonctions paramétrables et instanciables ;
- Vérification globale de la cohérence d'un programme ;
- Gestion multilingue de projet ;
- Interfaces d'import/export de code source des programmes, de la liste des noms symboliques de variables, de configurations matérielles ;
- Modèle objet STEP 7 pour réalisation de scripts destinés à automatiser le travail de l'automaticien.

### I.6.3- AVANTAGES

L'atelier logiciel STEP 7 Professional permet d'obtenir des gains de productivité importants de l'ingénierie durant toutes les phases d'un projet d'automatisation :

- Configuration plus rapide des systèmes grâce à des outils de configuration graphiques des composants et des réseaux ;
- Programmation plus efficace grâce à la possibilité de combiner dans des projets structurés, orientés objet, tous les langages de programmation standard selon CEI 61131-3 ;
- Phases de test plus courtes grâce aux outils de simulation et de mise au point intégrés ;
- Temps d'arrêts fortement réduits grâce à des moyens puissants de diagnostic de pannes et de maintenance à distance.

## I.7- PRESENTATIONS DES BLOCS DE PROGRAMMATION

La CPU fournit les types suivants de blocs de code qui permettent de créer une structure efficace pour le programme utilisateur.

### 1.7.1- BLOCS D'ORGANISATION (OB)

Ces blocs définissent la structure du programme. Certains OB ont des événements déclencheurs et un comportement prédéfini. Mais on peut également créer des OB à événements déclencheurs personnalisés.

### 1.7.2- FONCTIONS (FC) ET BLOCS FONCTIONNELS (FB)

Elles contiennent le code de programme qui correspond à des tâches ou combinaisons de paramètres spécifiques. Chaque FC ou FB fournit un jeu de paramètres d'entrée et de sortie pour partager les données avec le bloc appelant. Un FB utilise également un bloc de données associé :

- Appelé DB d'instance.
- Pour conserver les valeurs de données pour cette instance d'appel de FB.

Nous pouvons appeler un FB plusieurs fois et ce, avec un DB d'instance unique chaque fois. Utiliser des DB d'instance différents pour appeler le même FB n'affecte les valeurs de données dans aucun des DB d'instance. FC sont des blocs de code dans mémoire.

## I.8- LANGAGE DE PROGRAMMATION

### 1.8.1- LADDER (LD)

Un programme Ladder est un langage destiné pour la programmation des automates programmables industriels. Il ressemble à un schéma électrique, facile et compréhensive, il est présenté en forme logique, souvent utilisé dans la programmation. Il est composée de trois types d'éléments de *Langages* :

- **Les entrées** : ou contacts qui permettent de lire la valeur ou la variable. Il existe deux types d'entrées : Normalement ouvert, Normalement fermé ;
- **Les sorties** ou bobines qui permettent d'écrire la valeur ;
- **Les blocs fonctionnels** qui permettent de réaliser des fonctions avancées [12].

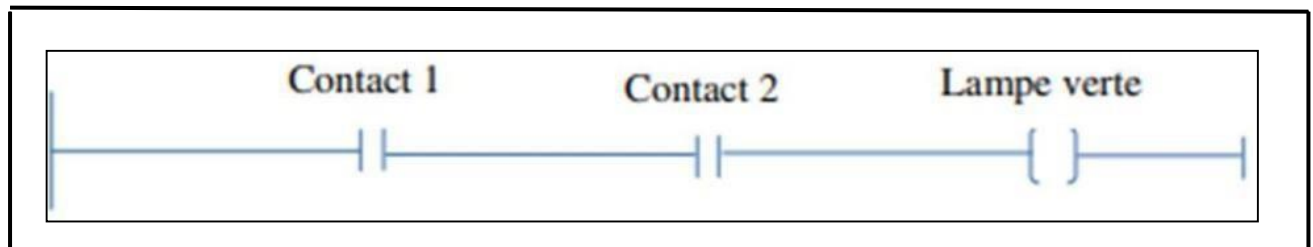


Figure I.11 : Présentation d'un schéma CONT



### I.8.2- LANGAGE LISTE D'INSTRUCTION (IL)

C'est un langage de programmation textuel proche de la machine. Dans cette programmation, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme. Pour faciliter la programmation, LIST a été complété par quelques structures de langage évolué (paramètres de blocs et accès structurés aux données).

*Tableau I.2 ; Fonction et instructions d'action de langage IL [12]*

| Désignation           | Fonctions  |
|-----------------------|--|
| LD                    | Le résultat est égal à l'opérande (load : lire la valeur).                     |
| LDN                   | Le résultat est égal à l'inverse de l'opérande (contact ouverture).            |
| AND                   | ET logique entre le résultat et précédent et l'état de l'opérande.             |
| ANDN                  | ET logique entre le résultat et précédent et l'état inverse de l'opérande.     |
| OR                    | OU logique entre le résultat et précédent et l'état de l'opérande.             |
| ORN                   | OU logique entre le résultat et précédent et l'état inverse de l'opérande.     |
| XOR,<br>XORN          | OU exclusif.   |
| Instructions d'action |  |
| ST                    | L'opérande associé prend la valeur de la zone de test.                         |
| STN                   | L'opérande associé prend la valeur inverse de la zone de test.                 |
| S                     | L'opérande associé est mis à 1 lorsque le résultat de la zone de test est à 1. |
| R                     | L'opérande associé est mis à 1 lorsque le résultat de la zone de test est à 1. |

### 1.8.3- LANGAGE GRAFCET

L'éditeur Grafcet (graph fonctionnel de commande étapes-transitions) permet de représenter graphiquement et de façon structurée le fonctionnement d'un automatisme séquentiel.

Autrement dit, le Grafcet est donc un langage graphique représentant le fonctionnement d'un automatisme par un ensemble :

- D'étapes auxquelles sont associées des actions.
- De transitions entre étapes auxquelles sont associées des conditions de transition (réceptivités).
- Des liaisons orientées entre les étapes et les transitions. [13].

#### I.8.4- LANGAGE LITTERAL STRUCTURE (ST)

L'éditeur de langage liste d'instruction permet l'écriture traitements logiques et numériques sous forme structurée. Cet éditeur dispose de fonctionnalités telles que :

- Les saisies et visualisation des opérandes sous forme de symboles et de repère ;
- L'aide à la saisie des instructions de la bibliothèque ;
- L'affichage en couleur des mots clefs du langage et des commentaires.

Comprimés pour transmettre la puissance, ils peuvent être trouvés dans d'innombrables applications industrielles.

```
PROGRAM stexample
VAR
  x : BOOL;
END_VAR
x := TRUE;
REPEAT
  x := FALSE;
UNTIL x := FALSE;
END_REPEAT;
END_PROGRAM;
```

Figure I.12 : Exemple Langage littéral structuré

**CHAPITRE II : ÉTUDE DETAILLE SUR LES  
DIFFERENTE TECHNIQUES ET METHODOLOGIE  
DE REGULATION ET ASSERVISSEMENT PEUVENT  
S'EFFECTUE PAR API**

## II.1- INTRODUCTION

Dans la plupart des appareils dans des installations industrielles et domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations extrêmes ou internes influant sur ces grandeurs. Par exemple, la vitesse et la position des moteurs, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré.

Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage dit en boucle ouverte, permet d'obtenir et de maintenir la valeur demandée (par exemple : action sur un robinet d'eau).

Dans la majorité des cas, ce type de réglage n'est pas suffisant, car trop grossier ou instable. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglant.

La régulation automatique regroupe l'ensemble des moyens matériels et techniques mis en œuvre pour maintenir automatiquement (pas d'intervention Manuelle) une ou plusieurs grandeurs physiques (installation de production, robot, alimentation électronique stabilisée, ...etc.) en vue d'en imposer le comportement. Cette prise de contrôle s'effectue par l'intermédiaire de certains signaux (grandeurs physiques) qu'il est alors nécessaire de mesurer afin de déterminer l'action à entreprendre sur le système.

Lorsque des perturbations ou un changement de consigne se produisent, la régulation automatique provoque une action correctrice sur une autre grandeur physique, parmi les grandeurs d'entrée du procédé (grandeur réglant), afin de ramener la grandeur réglée vers sa consigne initiale (cas de perturbations) ou vers sa nouvelle consigne (cas de changement de consigne c'est-à-dire changement de point de fonctionnement).

## II.2- CONCEPTION D'UN SYSTEME DE COMMANDE

La commande d'un processus consiste à déterminer la commande appropriée, de manière à assurer aux variables à contrôler (sorties) un comportement défini. L'action de la commande est une action susceptible de changer l'état du système à commander. Ces commandes sont Délivrées par un organe de commande ; le processus et son organe de commande constituent le système de commande.

Le système de commande comprend un élément nécessaire qui est le régulateur qui effectue le calcul de la commande à appliquer au processus à partir de la consigne et de l'état du processus. Lorsqu'il y a un retour d'information de la grandeur observée sur le régulateur, on parle d'un asservissement du système ou d'une régulation du système. [15]

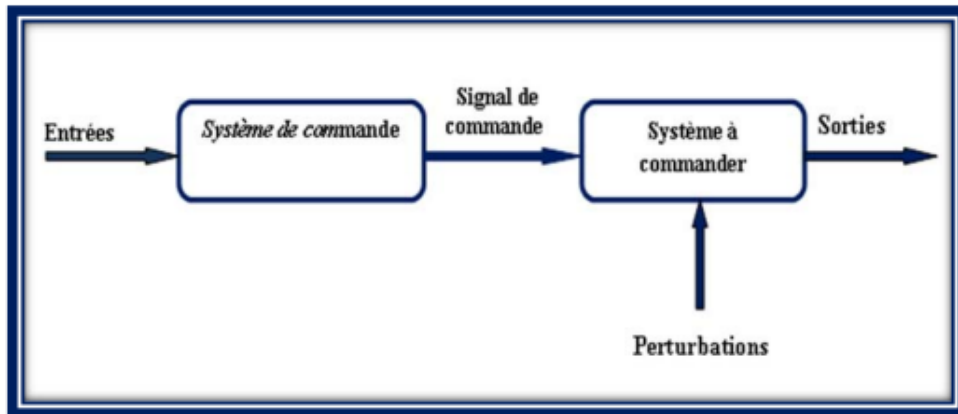


Figure II.1 Système de commande

❖ **ASSERVISSEMENT** : Dans une régulation, on s'attachera à maintenir constante la grandeur réglée d'un système soumis à des perturbations.

❖ **REGULATION** : Dans un asservissement, la grandeur réglée devra suivre rapidement les variations de la consigne [16].

## II.3- BOUCLE OUVERTE ET BOUCLE FERMEE

### II.3.1- BOUCLE OUVERTE (B.O.)

On parle de fonctionnement en boucle ouverte quand c'est l'opérateur qui contrôle l'organe de réglage. Il ne s'agit pas de régulation proprement dit, car cette technique n'utilise pas la mesure pour déterminer la commande du régulateur :

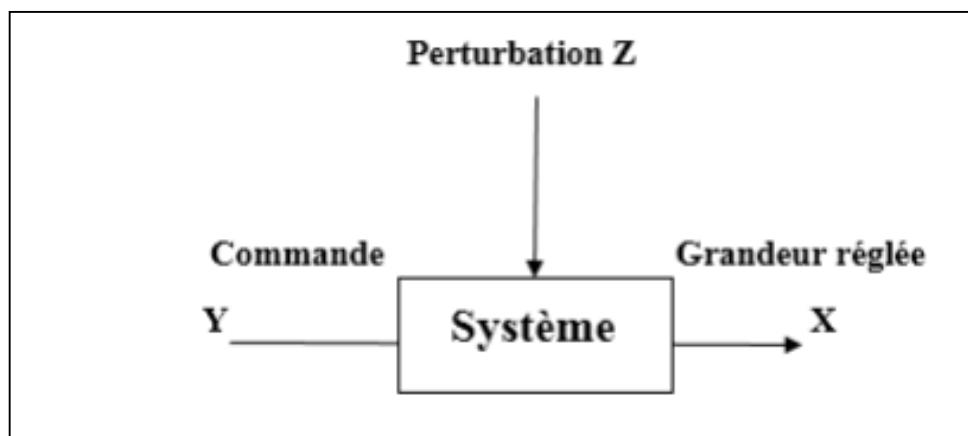


Figure II.2 : Boucle ouverte

### II.3.2- BOUCLE FERMEE (B.F.)

C'est le fonctionnement normal d'une régulation. Le régulateur compare la mesure de la grandeur réglée et la consigne et agit en conséquence pour s'en rapprocher. Le schéma ci-dessous représente la boucle fermée :

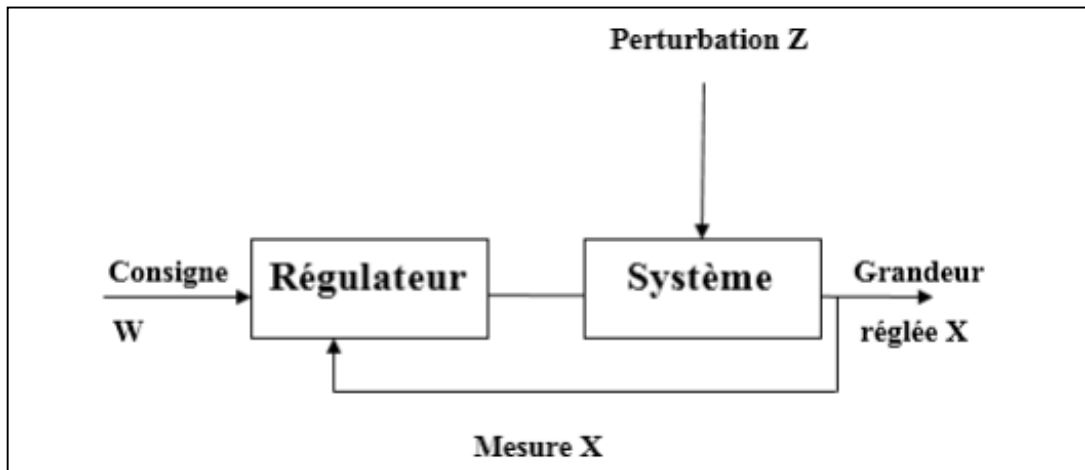


Figure II.3 : Boucle fermée

Exceptionnellement, le système de commande peut opérer en boucle ouverte à partir du seul signal de consigne. Mais la boucle fermée (contre réaction) est capable de :

- Stabiliser un système instable en B.O.
- Compenser les perturbations externes.
- Compenser les incertitudes internes au processus lui-même. [17]

#### II.4- CONSTITUANTS D'UNE CHAÎNE DE REGULATION

D'une manière générale, une boucle de régulation peut être représentée de la manière suivante :

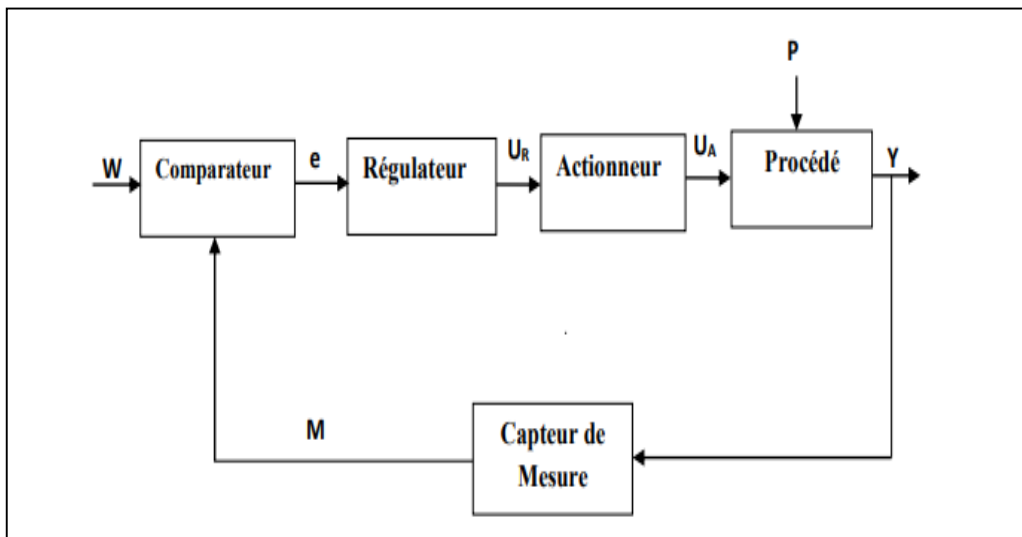


Figure II.4 : Représentation fonctionnelle d'une boucle de régulation

Cette organisation fonctionnelle représente la structure de base qu'on trouve dans tous les Systèmes asservis ou régulés. Elle fait intervenir deux chaînes : une chaîne d'action et une chaîne de retour ou d'informations.

- La chaîne d'action englobe tous les organes de puissance (nécessitant un apport extérieur d'énergie) et qui exécute le travail.

• La chaîne de retour ou de mesure : Analyse et mesure le travail effectué et transmet au comparateur une grandeur physique proportionnelle à ce travail. Elle comprend Généralement un capteur qui donne une mesure de la grandeur, qui est ensuite Amplifiée et transformée avant d'être utilisée.

❖ **CONSIGNE « W »** : La consigne est la grandeur qui doit commander la sortie, c'est-à-dire la valeur vers laquelle celle-ci doit tendre, pour finalement l'égaliser. La consigne n'a pas obligatoirement la même dimension que la sortie, mais doit être en accord avec la dimension de la mesure.

❖ **SORTIE « Y »** : Elle est le résultat de la régulation, c'est la variable que le système va influencer et/ou essayer de garder constante

❖ **RETOUR « M »** : Dans une boucle de régulation, la sortie est constamment contrôlée, il est ainsi possible de réagir à toute variation indésirable de celle-ci. La valeur mesurée (proportionnelle à la sortie) est appelée retour (ou mesure).

❖ **PERTURBATION « P »** : La perturbation est la grandeur qui influe de manière indésirable sur la sortie et qui l'éloigne de la valeur souhaitée (consigne). La seule existence d'une perturbation rend nécessaire la mise en œuvre d'une régulation statique.

❖ **COMPARATEUR** : C'est l'élément dans lequel la mesure actuelle de la sortie et la valeur de la consigne sont comparées. Dans la plupart des cas les deux grandeurs sont des tensions. La différence des deux grandeurs ainsi obtenue et appelée erreur « e ». La valeur de l'erreur va passer ainsi en entrée du régulateur pour y être traitée.

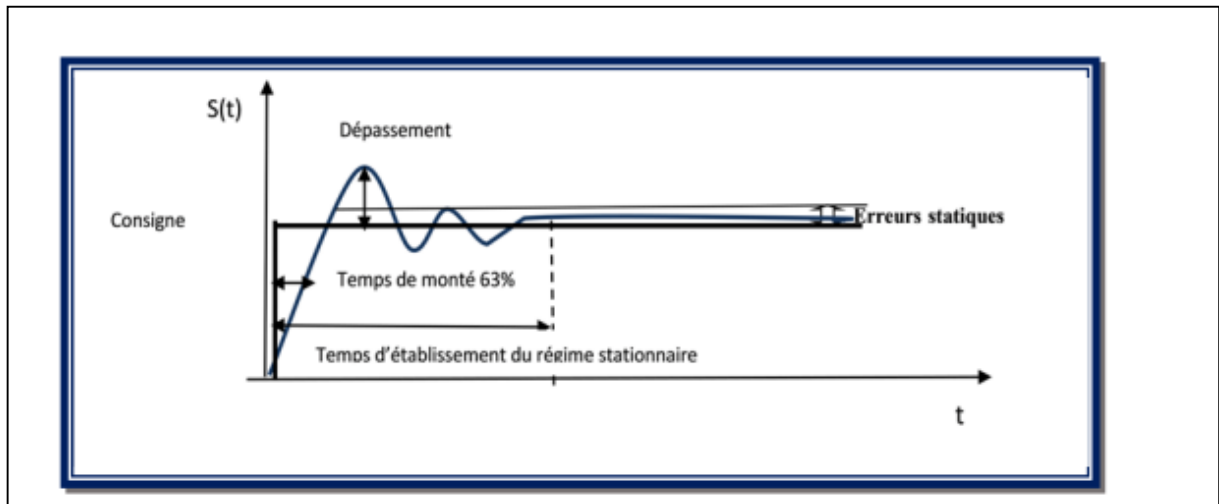
❖ **REGULATEUR** : Le régulateur est l'élément central d'une régulation. Il évalue l'erreur calculée par le comparateur, c'est-à-dire l'écart entre la sortie et la consigne, et en déduit à partir de celle-ci une valeur régulée « UR » ou valeur de correction à transmettre au procédé, afin de corriger la sortie. La façon (algorithme) avec lequel le régulateur calcule la valeur régulée à partir de l'erreur est la principale activité de la régulation.

❖ **ACTIONNEUR** : L'actionneur est en quelque sorte (l'organe exécutif) de la régulation. Il reçoit la variable régulée du régulateur. Cette variable d'entrée lui permet de savoir comment il doit influencer sur la sortie de la régulation.

❖ **PROCEDE** : Le procédé est le cœur du système régulé ou encore la partie originelle. C'est également la partie qui agit directement sur la valeur de sortie.

## II.5- PERFORMANCES DES SYSTEMES REGLES

Les performances des systèmes réglés définies dans un cahier des charges, sont illustrées par La figure II.5.

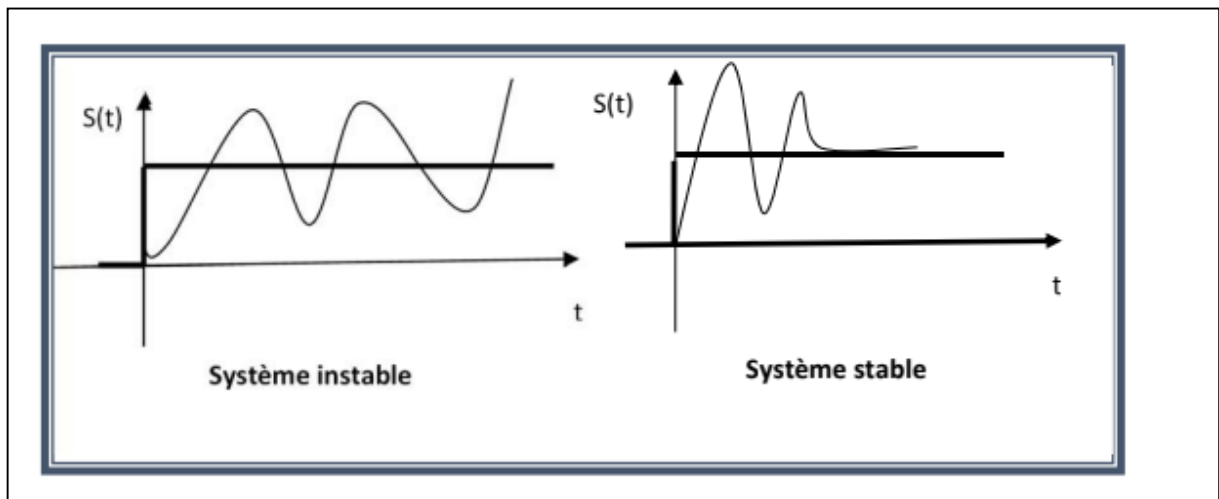


*Figure II.5 : Performances d'un système de commande.*

❖ **RAPIDITE** : La rapidité quantifie le temps de réponse du système. Elle correspond au temps de réaction de la sortie par rapport à la consigne. Le temps mis par la réponse pour ne plus dépasser 5% de la valeur finale. Ce temps est retenu Comme critère de rapidité 5%.

❖ **PRECISION** : La précision quantifié l'erreur lorsque l'équilibre est atteint, Avec l'entrée  $e(t)$  et la sortie  $s(t)$  de même nature. Autrement, un système est précis si la sortie suit la consigne en toutes circonstances avec un écart inférieur à la valeur définie dans un cahier des charges.

❖ **STABILITE** : On dit qu'un système est stable lorsque celui-ci tend à revenir à son état d'équilibre pour une consigne constante, la sortie doit être constante [18].



*Figure II.6 : Stabilité du système*

Toutes les méthodes étudiées jusqu'à présent de l'asservissement linéaire restent valables et efficaces jusqu'à ce que ces systèmes atteignent une complexité telle que l'on ne puisse plus se satisfaire de l'unique relation entrée-sortie (c'est-à-dire les systèmes mono-variables SISO : Single Input Single Output) pour les commander correctement.



De même, ces modèles deviennent difficiles à mettre en oeuvre lorsque les systèmes étudiés possèdent plusieurs entrées et plusieurs sorties cas d'un système multi-variable (MIMO : Multi Input Multi Output).

Les théories de commande avancées sont basées complètement sur les modélisations modernes sous la forme des variables d'état. La représentation d'état des systèmes est un outil puissant permettant de modéliser le fonctionnement de systèmes linéaires, en temps continu ou en temps discret et qui possède en outre, l'avantage de conserver la représentation temporelle des phénomènes.

**1- Commande par mode glissant :** La commande à structure variable (CSV) est par nature une commande non linéaire. La caractéristique principale des systèmes à structure variable est que leur loi de commande se modifie d'une manière discontinue. Les commutations de la commande s'effectuent en fonction des variables d'état, utilisées pour créer une "variété" ou "hyper surface" dite de glissement dont le but est de forcer la dynamique du système de correspondre avec celle définie par l'équation de l'hyper surface. Quand l'état est maintenu sur cette hyper surface, le système est dit en régime glissant.

Ainsi, tant que les conditions de glissement sont assurées, la dynamique du système reste insensible aux variations des paramètres du processus, aux erreurs de modélisation et à certaines perturbations. Ceci lui permet d'être particulièrement adaptée pour traiter les systèmes qui ont des modèles mal connus, soit à cause de problèmes d'identifications des paramètres, soit à cause de simplification sur le modèle du système. [19]

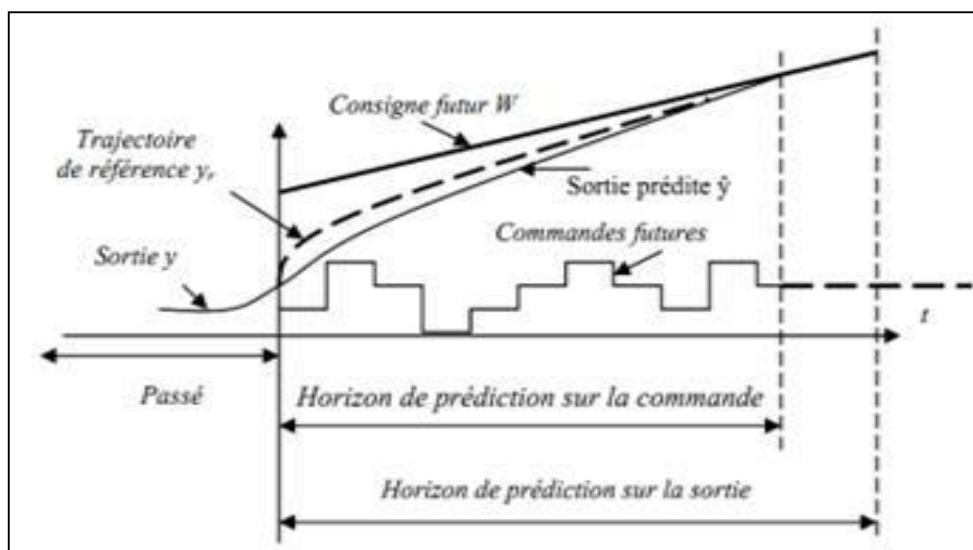


Figure II.7 : Démonstration du mode de glissement

**2- Commande prédictive :** La commande prédictive est une technique de commande avancée de l'automatique. Elle a pour objectif de commander des systèmes industriels complexes. Le principe de cette technique est d'utiliser un modèle dynamique du processus à l'intérieur du contrôleur en temps réel afin d'anticiper le futur comportement du procédé. [20]

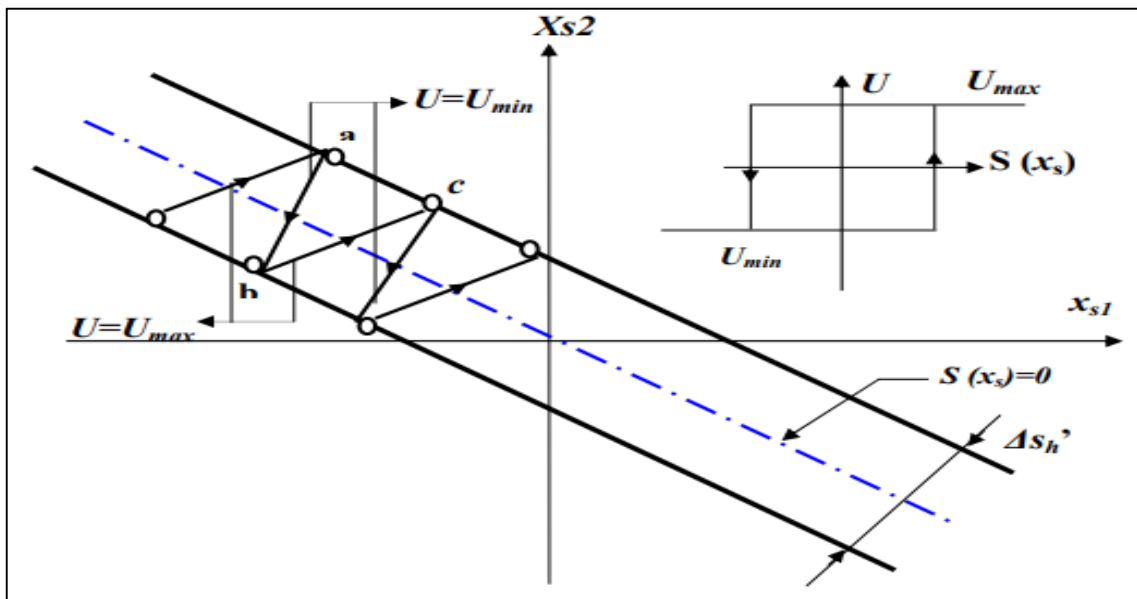


Figure II.8 : Le principe de base de la commande prédictive à base de modèle

**3- Commande adaptative :** L'adaptation d'un système à son environnement réside dans la possibilité de réagir face aux variations que peut subir cet environnement. La commande adaptative est une commande dont le but est de réagir à tout instant dans le sens désiré (en générale minimisation de l'erreur entre la consigne et la sortie) face aux variations que subit le système. [21]

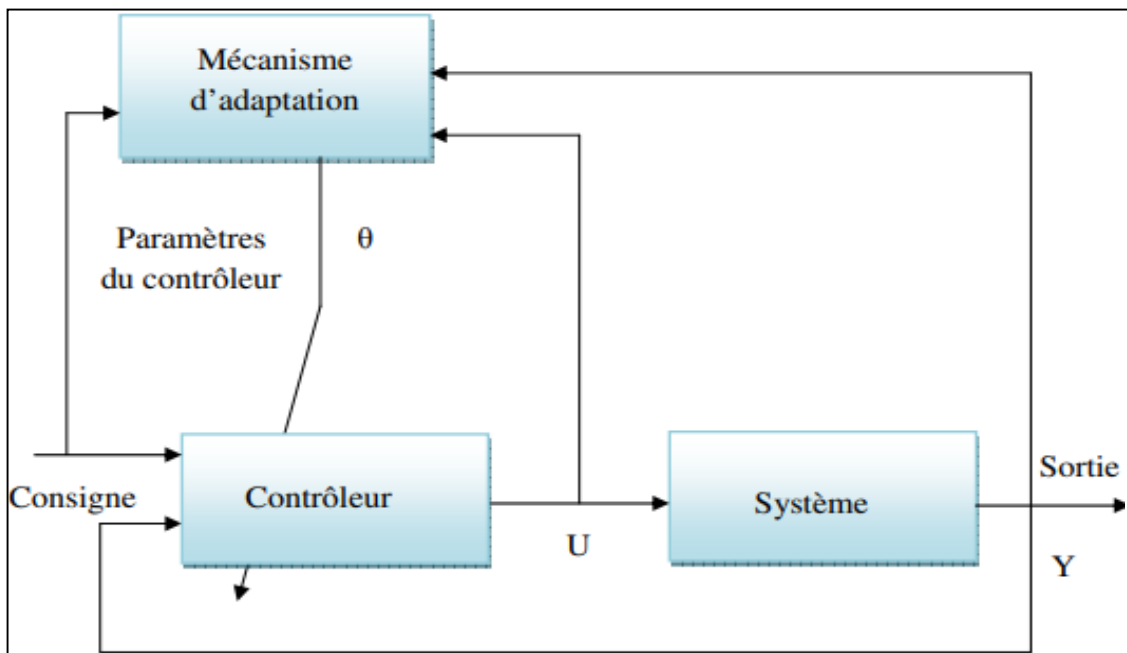


Figure II.9 : Principe de la commande adaptative

**4- Commande DTC :** La commande DTC (Direct Torque Control) est un type de commande vectorielle pour un onduleur de tension alimentant une machine asynchrone. Son objectif est de réguler le flux statorique et le couple électromagnétique sans disposer de mesures de flux ou de couple. Les seules mesures utilisées sont les courants alimentant le stator de la machine et la vitesse de rotation. Le flux et le couple sont entièrement estimés à partir de ces mesures. En ce qui concerne la connaissance de la machine, la valeur de la résistance statorique est nécessaire pour calculer le flux. Cette résistance est amenée à varier

dans le temps, et ce paramètre doit donc être parfaitement connue pour une telle commande. Dans ce chapitre, nous développons le principe de la commande direct du couple électromagnétique de la machine asynchrone. Pour cette stratégie de commande, nous expliquons le principe de réglage du flux et du couple électromagnétique se basant sur le modèle de l'onduleur de tension. Et on terminera avec l'étude de la commande directe du couple avec un régulateur PI de vitesse. Des résultats expérimentaux et de simulation sont présentés pour mettre en évidence les performances de la commande directe du couple avec et sans régulation. [22]

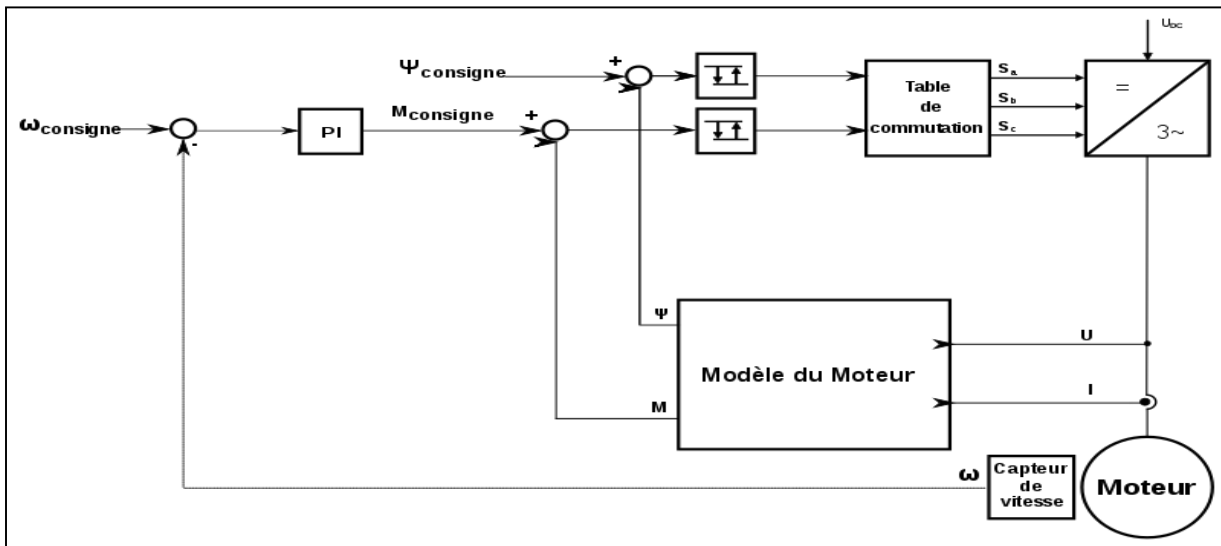


Figure II.10 : Schéma bloc d'une commande directe du couple

**5- Commande robuste :** Est un type de commande qui vise à garantir les performances et la stabilité d'un système face à des perturbations du milieu et les incertitudes du modèle. En effet, le modèle mathématique qui modélise un système réel est une représentation qui vise à approximer au mieux, avec des hypothèses simplificatrices, le système qu'on veut commander. Il existe donc un écart entre le comportement observé du système réel et son modèle interne. La commande robuste vise à déterminer une loi de commande qui soit capable de garantir des critères de performance et stabilité pour un système dont le modèle varie autour du modèle théorique ou nominal. Par la même approche, on peut rendre le système robuste face aux perturbations extérieures.[23]

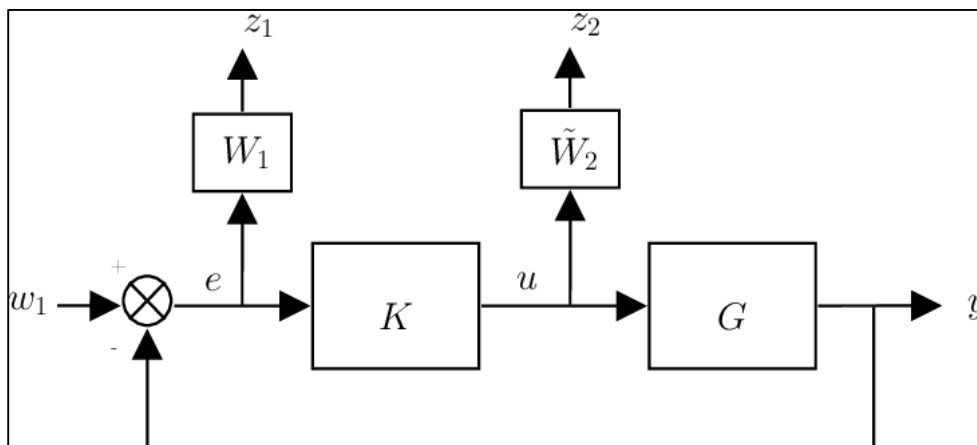


Figure II.11 : La commande robuste

**6- Commande optimale :** L'objectif principal de la commande optimale consiste à déterminer les signaux de commande qui vont entraîner un procédé (système) pour satisfaire certaines contraintes physiques et en même temps maximiser ou minimiser un critère de performance choisi (indice de performance ou fonction coût).

On s'intéresse à trouver une loi de commande optimale  $u^*(t)$  qui va conduire le système P d'aller de l'état initial vers un état final donné avec certaines contraintes sur les signaux de commande (contrôle) et les états et en même temps maximisant ou minimisant l'indice de

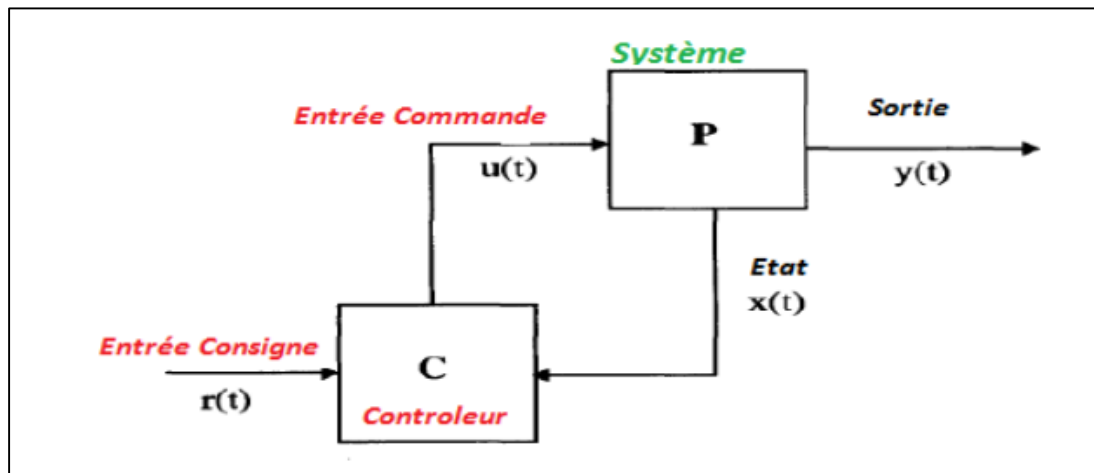


Figure II.12 : La commande optimale

**7- Commande PID :** P, I et D constituent des correcteurs dont l'association permet d'assurer une régulation optimale d'un procédé :

- L'action proportionnelle P permet d'accélérer la réponse du procédé proportionnellement à l'écart Mesure-Consigne. P donne la nervosité du régulateur PID ;
- L'action intégrale I complète l'action P en annulant l'erreur résiduelle sans modifier la réaction du régulateur PID ;
- L'action dérivée D permet de compenser les effets du temps mort du procédé et améliore la stabilité de la boucle en amortissant rapidement les oscillations provoquées par une perturbation ou un changement brusque de la consigne.

La loi de commande du régulateur PID est :

$$u(t) = K_p + K_i \int \varepsilon(t)dt + K_d \frac{d\varepsilon(t)}{dt} \quad (\text{V. 1})$$

$$C(s) = K_p + \frac{K_i}{s} + K_d \cdot s \quad (\text{V. 2})$$

Le temps mort  $\tau$  (ou retard  $T_r$ ) désigne l'intervalle de temps entre la variation de la consigne et la variation de la mesure. Les longueurs de canalisations induisent des temps morts.

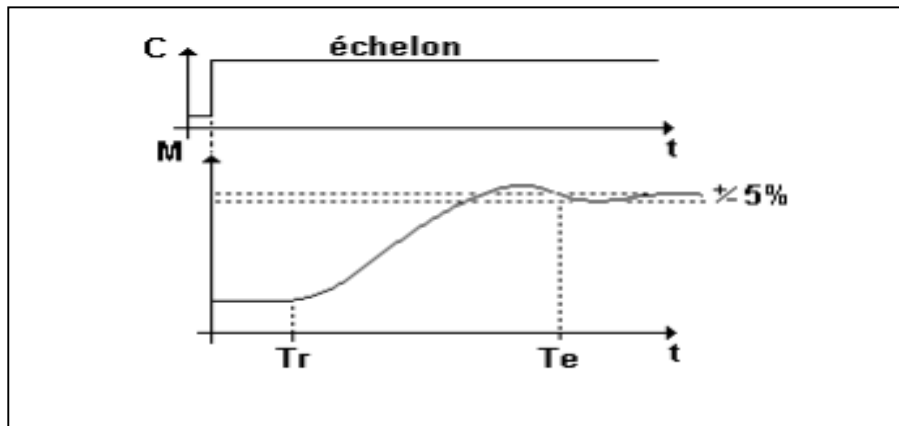


Figure II.13 : La variation de la consigne et la mesure

Le temps d'établissement  $T_e$  correspond à l'intervalle de temps entre la variation de la consigne et l'instant où la mesure entre dans un intervalle de  $\pm 5\%$  sans ne plus en ressortir. Les réglages du correcteur P, I et D permettent d'optimiser ce temps d'établissement. [25]

La synthèse des paramètres des actions PIDS :

❖ **Méthodes en boucle ouverte de Ziegler-Nichols [26]**

C'est la méthode la plus ancienne (1942). Elle a pour objectif la détermination du réglage d'un régulateur PID à partir de la réponse à un échelon du procédé. On dispose de la réponse  $X(t)$  (variation de la sortie) suite à un échelon d'entrée  $Y(t)=Ay$ .

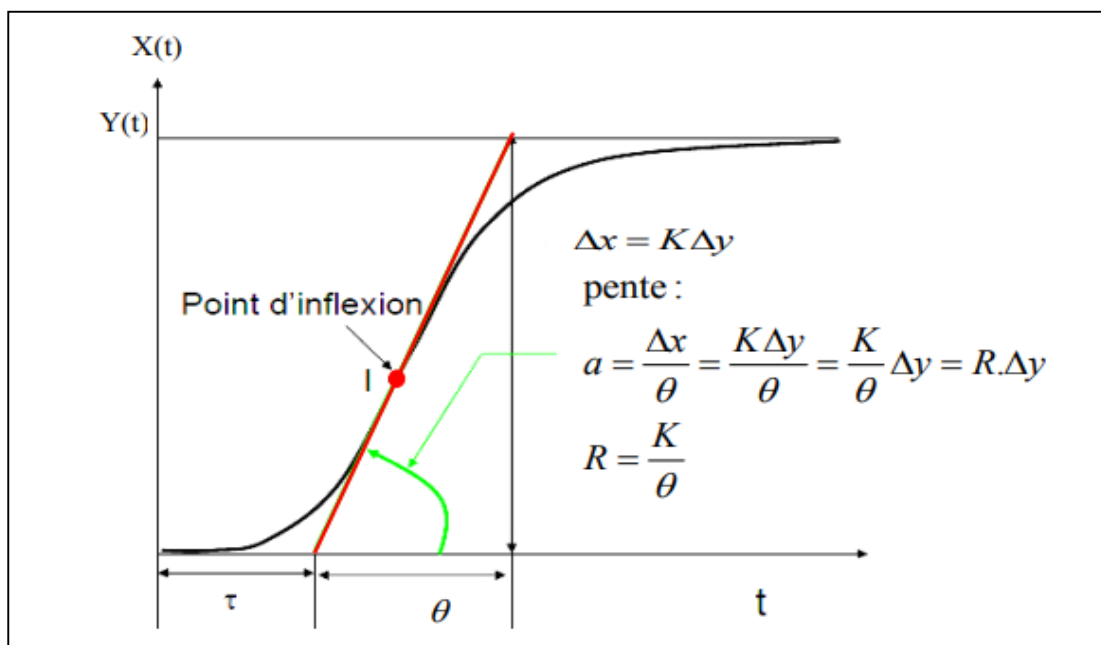


Figure II.4: Réponse en boucle ouverte du procédé à un échelon de commande  $Ay$

Les valeurs des paramètres PID en boucle ouverte selon Ziegler Nichols sont données sur le tableau ci-dessous.

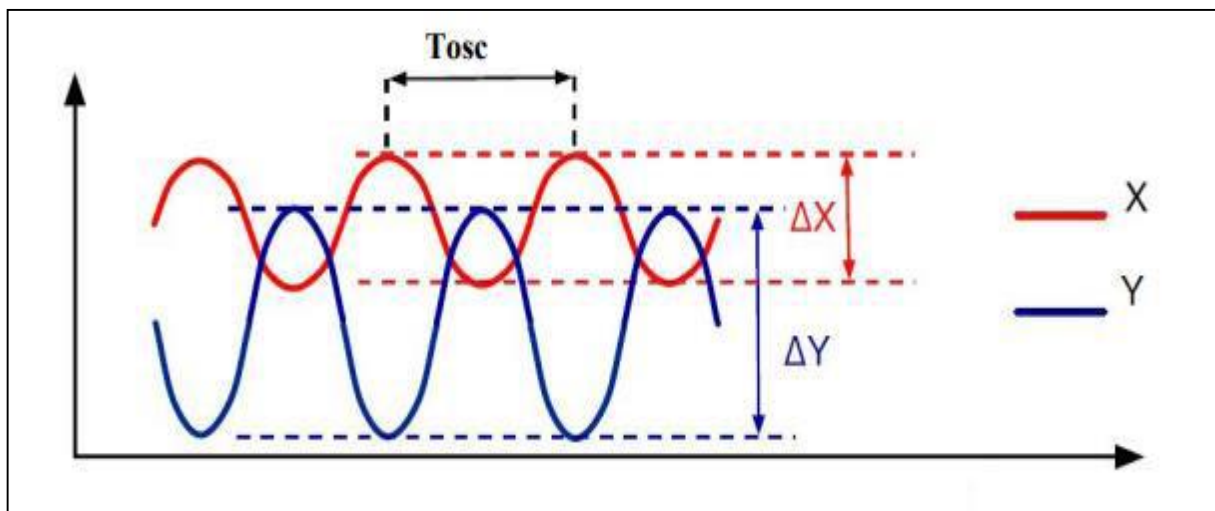
**Tableau II.1 : Calcul des paramètres PID en boucle ouverte**

| Type de régulateur                   | Gain Kp               | Ti        | Td        |
|--------------------------------------|-----------------------|-----------|-----------|
| Proportionnel(P)                     | $\frac{1}{R.\tau}$    | 0         | 0         |
| Proportionnel intégrale (PI)         | $\frac{0,9}{R.\tau}$  | $3.3\tau$ | 0         |
| Proportionnel intégrale derive (PID) | $\frac{1,27}{R.\tau}$ | $2\tau$   | $0,5\tau$ |

### II.6.1- METHODE DE ZIEGLER NICHOLS EN BOUCLE FERMEE

Cette méthode est la plus connue des méthodes pratiques de réglage des boucles de régulation. L'avantage de cette méthode est qu'il n'y a pas besoin de connaître la fonction de transfert du procédé et que le réglage se fait directement sur le site en chaîne fermée. Après avoir porté la mesure près du point de consigne désiré (manuellement ou automatiquement), le régulateur est réglé en action proportionnelle seule. Le gain  $G_r$ , est alors augmenté progressivement jusqu'à obtention du pompage sans que l'organe réglant ne soit jamais en saturation. Le gain critique  $G_{rc}$  Du régulateur est le plus petit gain qui permet l'entretien des oscillations. La période d'oscillation est mesurée sur l'enregistrement (figure II.6) [26]

Ziegler et Nichols proposent alors les valeurs de réglage du tableau suivant :



**Figure II-5 : Mise en oscillation juste entretenues d'un système**

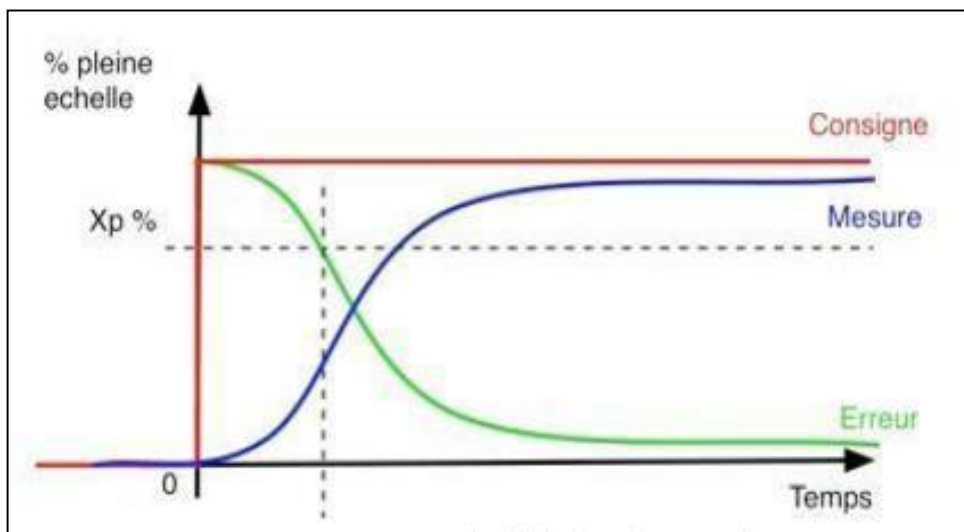
**Tableau II.2 : Réglages préconisés par Ziegler et Nichols en boucle fermée**

| Type de régulateur | P                  | PI                 | PI Parallèle        | PID série         | PID Parallèle           | PID mixte         |
|--------------------|--------------------|--------------------|---------------------|-------------------|-------------------------|-------------------|
| Grc                | $\frac{Grc}{2}$    | $\frac{Grc}{2.2}$  | $\frac{Grc}{2.2}$   | $\frac{Grc}{3.3}$ | $\frac{Grc}{1.7}$       | $\frac{Grc}{1.7}$ |
| Ti                 | Ti maxi ou annulée | $\frac{Tosc}{0.2}$ | $\frac{2Tosc}{Grc}$ | $\frac{Tosc}{4}$  | $\frac{0.85Tosc}{Grc}$  | $\frac{Tosc}{2}$  |
| Td                 | 0                  | 0                  | 0                   | $\frac{Tosc}{4}$  | $\frac{Grc.Tosc}{13.3}$ | $\frac{Tosc}{8}$  |

### II.6.2- REGULATEUR A ACTION PROPORTIONNELLE (P)

Le régulateur à action proportionnelle, ou régulateur P, a une action simple et naturelle, puisqu'il construit une commande  $u(t)$  proportionnelle à l'erreur  $e(t)$ .

- Loi de commande du régulateur P :  $u(t) = K_p \cdot e(t)$ .
- Fonction de transfert du régulateur P :  $C(s) = U(s)/E(s) = K_p$ .
- Schéma fonctionnel du régulateur P.



**Figure II.6 : Réponse indicielle du régulateur P**

### II.6.3- REGULATEUR A ACTION INTEGRALE (I)

Un régulateur de type intégral permet d'annuler l'erreur statique. Tant que l'erreur n'est pas nulle, la valeur de la variable régulée est ajustée. La régulation se termine lorsque la sortie a atteint la valeur de la consigne ou que la variable régulée a atteint un seuil maximal fixé par les propriétés du système (U max, P max...etc.).

La formulation mathématique de ce comportement intégral est que la variable régulée est proportionnelle à l'intégrale par rapport au temps de l'erreur  $\epsilon$ .

$$u(t) = K_i \int \varepsilon(t) dt \quad (\text{V.3})$$

La fonction de transfert d'un correcteur Intégrale s'écrit alors :

$$\left\{ \begin{array}{l} C(s) = \frac{K_i}{s} \\ K_i = \frac{K_p}{T_i} \end{array} \right\} \quad (\text{V.4})$$

La constante de temps  $T_i$  exprimée souvent en unité de temps est appelée la constante de temps d'intégration. La vitesse avec laquelle la variable régulée augmente (ou diminue) dépend de l'erreur statique et de la constante de temps d'intégration choisie. L'influence de  $T_i$  sur le système est montrée dans la figure ci-dessous :

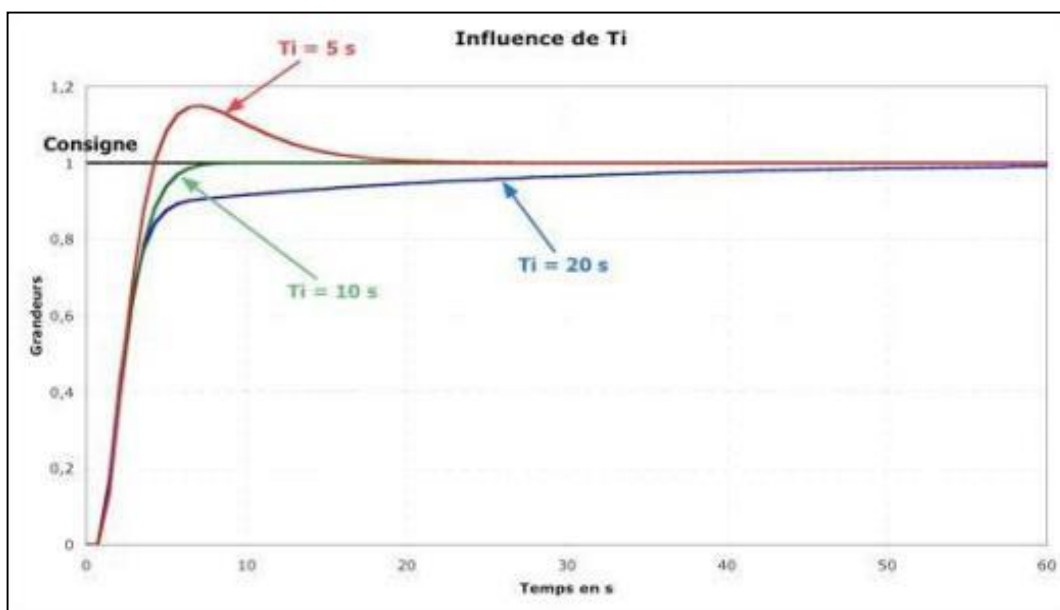


Figure II.7: Influence du temps d'intégration I

#### II.6.4- REGULATEUR DERIVATEUR PUR (D)

Le régulateur dérivé établit une valeur régulée en fonction de la vitesse de variation de l'erreur et pas en fonction de l'amplitude comme pour le régulateur P. C'est pour cette raison qu'il réagit beaucoup plus rapidement qu'un régulateur P. Même face à une petite erreur il va générer une grande valeur régulée dès lors qu'il y a une variation d'amplitude de l'erreur. Le régulateur D sera inefficace face à une erreur résiduelle permanente, quelle que soit sa valeur puisque celle-ci reste constante (pas de variation d'amplitude donc pas de réaction du régulateur).

C'est pourquoi ce type de régulateur sera rarement utilisé ; Seul dans la pratique, il est couramment associé à un régulateur de type P. La loi de commande est de la forme :

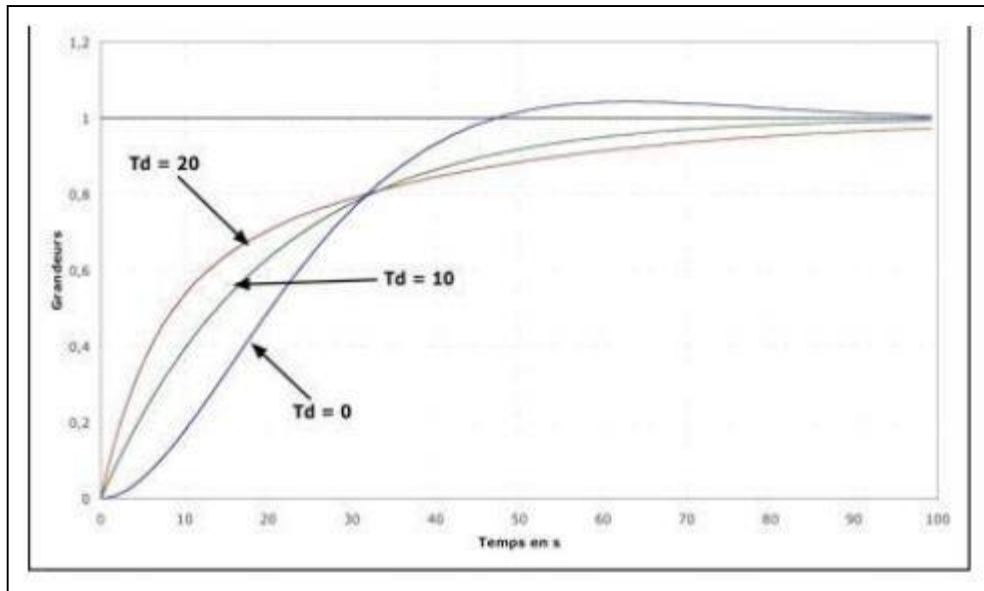
$$u(t) = \frac{d\varepsilon(t)}{dt} \quad (\text{v-5})$$

La fonction de transfert d'un correcteur dérivée s'écrit alors :



$$\begin{cases} C(s) = K_d s \\ K_d = K_p T_d \end{cases} \quad (V.6)$$

La constante de temps  $T_d$  de dimension de l'unité de temps est appelée constante de temps de dérivation. Elle intervient également comme un gain. La fonction de transfert de ce type de correcteur est purement théorique : un système physique ne peut pas avoir un numérateur de degré supérieur à celui du dénominateur. L'influence de  $T_d$  sur le système est montrée dans la figure ci-dessous :



*Figure II.8 : Influence du temps de dérivation D*

Sous forme d'un tableau récapitulatif, on résume les avantages et l'influence des actions de base des régulateurs PID :

*Tableau II.3 : Influence des actions des régulateurs PID*

| Action                | Proportionnel   | Intégrateur  | Dérivée  |
|-----------------------|---|--|--|
| <b>Points forts</b>   | Action instantanée  | Annule l'erreur statique                               | Action très dynamique améliore la rapidité                         |
| <b>Points faibles</b> | Ne permet pas d'annuler une erreur statique mais permet de la réduire | Action lente Ralentit le système (effet déstabilisant) | Sensibilité aux bruits Forte sollicitation de l'organe de commande |
| <b>Stabilité</b>      | Diminue   | Augmente   | Diminue  |
| <b>Précision</b>      | Augmente  | Pas influence  | Pas influence  |
| <b>Rapidité</b>       | Augmente  | diminue  | Augmente   |

## **II.7- DIFFERENTES TECHNIQUES ET METHODOLOGIE DE REGULATION ET ASSERVISSEMENT PEUVENT S'EFFECTUE PAR API**

## II.7.1- REGULATION TOUT OU RIEN [16]

### II.7.1.1- ACTION CONTINUE / ACTION DISCONTINUE

On sépare le fonctionnement d'un régulateur en deux types d'actions distincts :

- Une action continue avec une sortie du régulateur peut prendre toutes les valeurs comprises entre 0 et 100% ;
- Une action discontinue, dans laquelle la sortie Y du régulateur ne prend que deux valeurs. On appelle aussi le fonctionnement discontinu fonctionnement Tout Ou Rien.

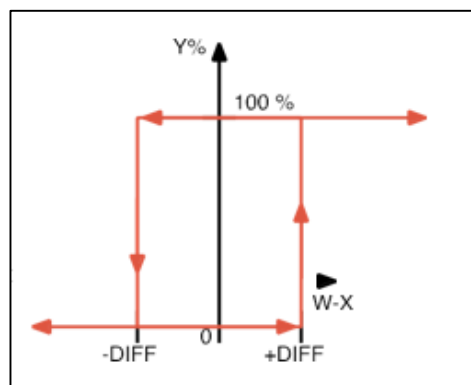


*Figure II.9 : Action continue / Action discontinue*

Le fonctionnement TOR se caractérise par deux états possibles pour la commande. Celui qui correspond à la commande maximale (100 %) et celui qui correspond à la commande minimale (0 %). Un seuil limite la fréquence de commutation du système pour éviter une fatigue prématurée des organes de réglages.

Le réglage du régulateur se fait à l'aide de deux paramètres :

- La consigne W, fournie en unité de mesure ;
- Le seuil DIFF, donné généralement en % de la consigne.

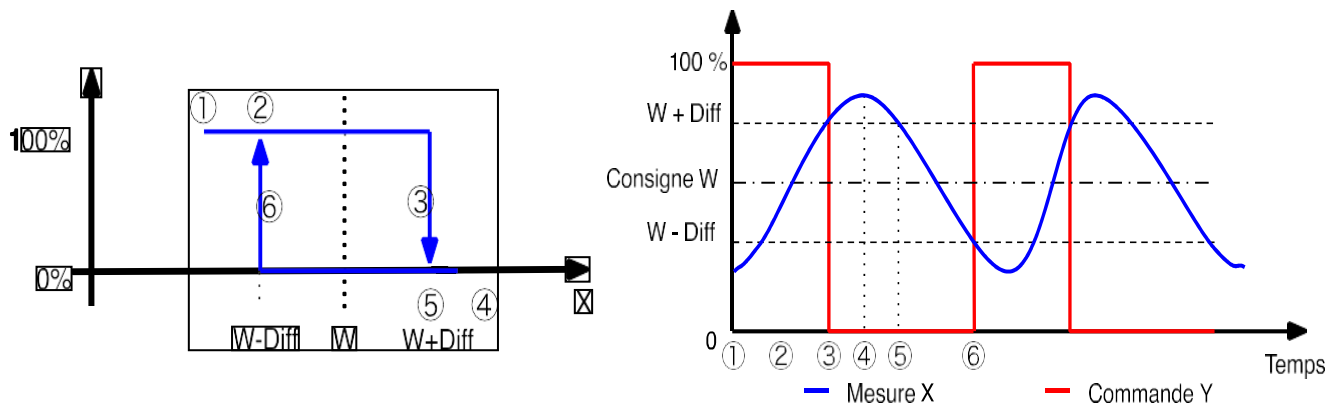


*Figure II.10 : Le réglage du régulateur*

### II.7.1.2- FONCTIONNEMENT

La grandeur réglée oscille autour du point de fonctionnement. À chaque dépassement des seuils de commutation, la sortie du régulateur change d'état. Compte tenu de l'inertie du système, la valeur absolue de l'erreur  $\varepsilon$  peut dépasser le seuil DIFF.

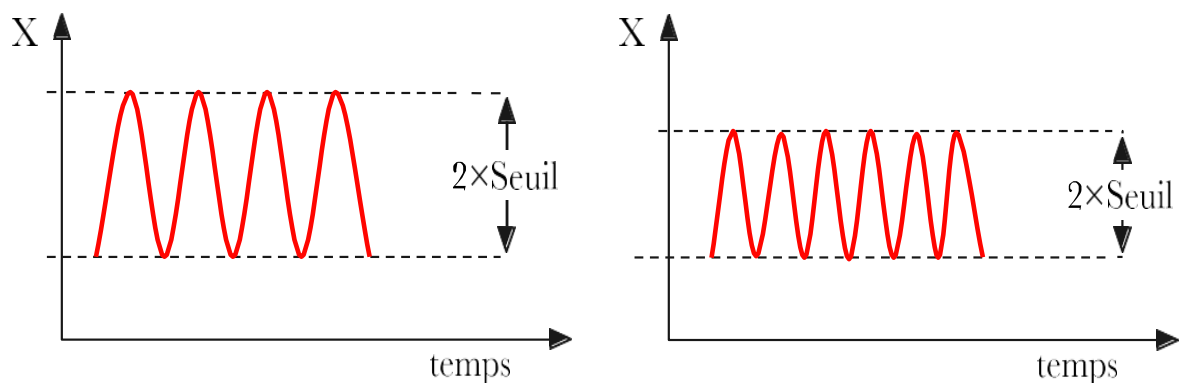
**Remarque :** La mesure ne peut pas être constante dans ce type de régulation, le système est en régime d'instabilité entretenue.



**Figure ii.11 : Fonctionnement**

### II.7.1.3- INFLUENCE DU PARAMETRE SEUIL

La valeur du seuil influe sur la fréquence des permutations et l'amplitude de la variation de la grandeur mesurée. Plus le seuil est faible, plus la fréquence est élevée, moins l'amplitude est grande. Une augmentation de la fréquence réduit d'autant la durée de vie de l'organe de réglage.



**Figure ii.12 : Influence du paramètre seuil**

## II.7.2- REGULATION PID

### II.7.2.1- COMPOSANTE PROPORTIONNELLE

➤ **Bande Proportionnelle :** La bande proportionnelle c'est l'écart entre la valeur mesurée et la valeur de la consigne notions que si la bande proportionnelle est égale à zéro la régulation fonctionne en tout ou rien. L'action proportionnelle du régulateur s'exprime de différentes manières, soit par le gain (coefficient multiplicateur de l'écart), soit en % à appliquer pour que la sortie varie 0 à 100 %, soit encore en K ici on parlera d'écart pour passer de 0% à 100 %. [27] :

• **En % :** Température mesurée 26 °C, température consigne 24 °C, BP 20 % on aura en sortie de régulateur :  $(26-24)/(26*20\%) = 38\%$  ;

• **En gain** : Température mesurée 20°C, température consigne 22°C, pour un gain de 20 on aura ici en sortie :  $(22-20)*20=40\%$ .

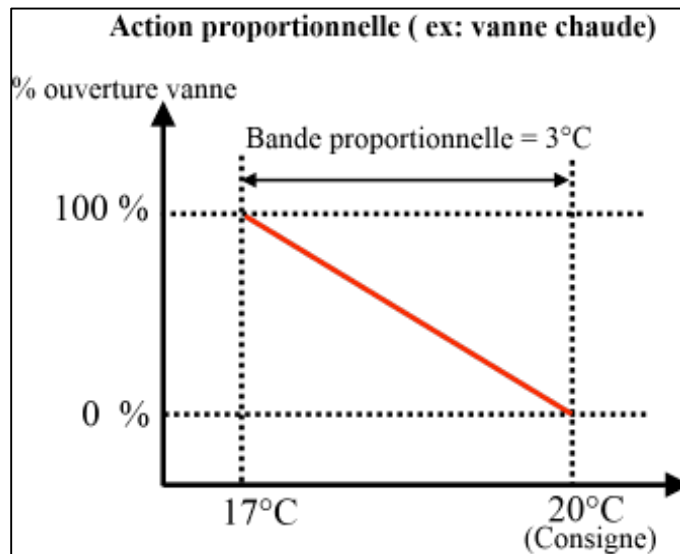


Figure II.13 : Acton proportionnelle

### II.7.2.2- COMPOSANTE INTEGRALE

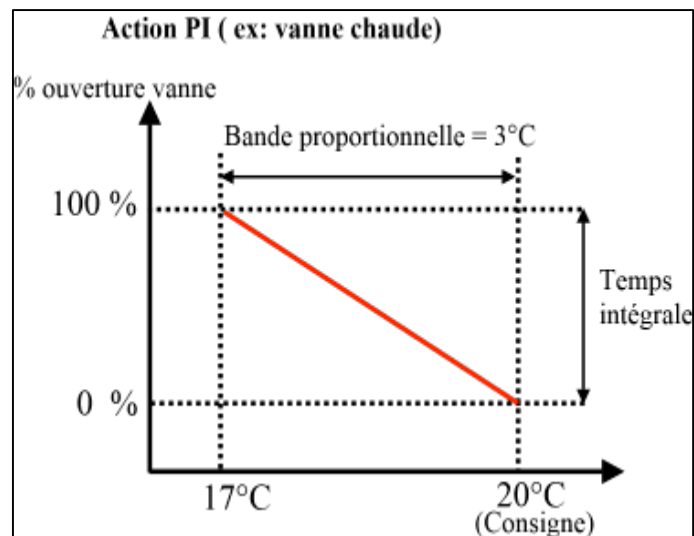


Figure II.14 : Acton intégrale

Cette composante apporte une notion de temps d'intégration à la correction, cette notion de temps s'exprime généralement en seconde. Cette action est complémentaire à l'action proportionnelle, elle permet de stabiliser dans le temps l'action proportionnelle, plus l'erreur mesurée est constante plus la correction est constante. L'algorithme de la régulation utilisera une mémoire des différents écarts déjà mesurer pour affiner le temps d'intégrale à appliquer à la correction. [17]

### II.7.2.3- COMPOSANTE DEVIEE

Cette action permet d'anticiper la réponse de la régulation en cas de perturbation rapide ou de modification de consigne ce qui améliore la stabilité du système. On peut donc dire que cette composante permet de compenser tout dépassement excessif de la consigne.

### II.7.3- REGULATION ANALOGIQUE [26]

Un régulateur analogique est conçu à partir du pont de Wheatstone, le signal délivré est proportionnel à l'écart entre la mesure et la consigne, c'est une régulation non communicante. Un courant est appliqué en entrée du pont de Wheatstone, quatre résistances sont raccordées en série parallèle, celles-ci peuvent être toutes variables ou certaines à résistances variables et certains fixes. Le courant émit en sortie sera modulé en fonction de la valeur ohmique des résistances. Si les quatre résistances ont la même valeur ohmique rien ne se passe s'il y a déséquilibre car une ou plusieurs résistances varies une tension est générée celle-ci est utilisée pour commander divers actionneurs. Ce signal permet d'actionner après amplification des contacts secs (contact on/off) ou de libérer à l'aide d'un convertisseur un courant de 0 à 10 V (volet mélange, vanne 2 et 3 voies etc..).[12]

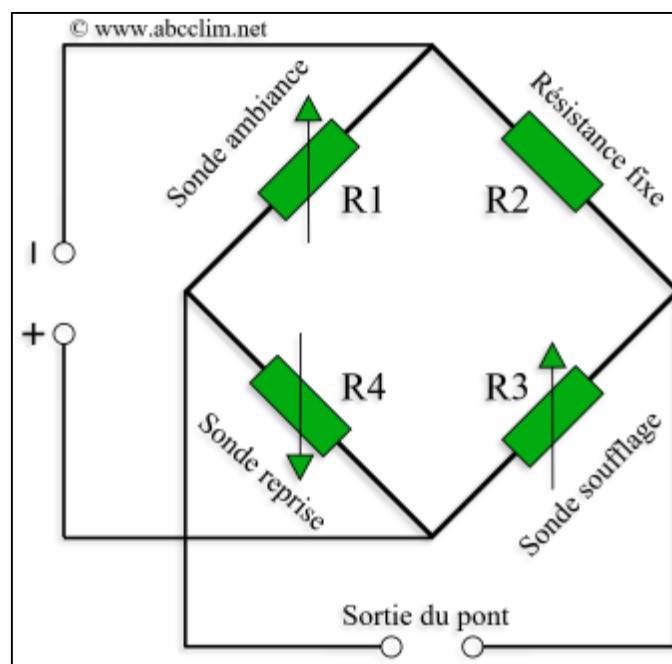


Figure II.15 : Pont de Wheatstone

La régulation analogique permet des actions de types :

- Proportionnelles (P). □ Intégrales (PI).
- Proportionnelles Intégrales Dérivées (PID).

### II.7.4- REGULATION NUMERIQUE [26]

#### II.7.4.1- FONCTIONNEMENT GENERAL

Un microprocesseur interprète des informations fournies par des entrées analogiques ou tout ou rien (Tor) suivant un programme (P, PI, PID) préenregistré dans la mémoire du processeur ou spécifique introduit à la demande, pour agir sur des actionneurs.

Le programme est sauvegardé en cas d'incident soit dans une mémoire flash ou dans une mémoire physique alimentée par pile, on peut aussi récupérer les paramètres et données via un ordinateur ou une interface contenant la programmation (Smart key). Le dialogue entre les différentes interfaces ou modules est assuré par un "bus", liaison informatique à 2 fils.

#### II.7.4.2- TYPE D'ENTREES ET DE SORTIES

➤ **ENTREES : (INPUT)**

1. AI : (Analogique, Input) Sondes passive ou active = sondes ambiances, soufflage, reprises, capteur de pression etc.
2. DI : (Digital Input) entrée on/off = pressostat d'air, HP, BP, hypothermie, sécurité.

➤ **SORTIES : (OUTPUT)**

1. AO : (Analogique Output), sortie proportionnelle = Vannes trois voies, volets de mélange, triac (résistances).
2. DO : (Digital Output) sortie tout ou rien = Compresseurs, pompes, ventilateurs etc

#### II.7.5- LOGIQUE FLOUE

La commande floue a pour but de traiter des problèmes de commande de processus à partir uniquement des connaissances du comportement de système que les spécialistes du procédé doivent formuler sous forme linguistique (la connaissance des diverses fonctions de transferts n'est pas nécessaire). Ces connaissances du comportement permettent d'élaborer les règles de commande implantées dans le contrôleur flou. [29] Contrairement à la logique classique (binaire), les variables de la logique floue peuvent prendre plusieurs valeurs entre 0 et 1.

La logique floue est fondée sur la théorie des sous ensemble qui permet de traiter des problèmes vagues, imprécis et flous. La figure (32) représente respectivement l'ensemble classique (A) et l'ensemble flou (B).

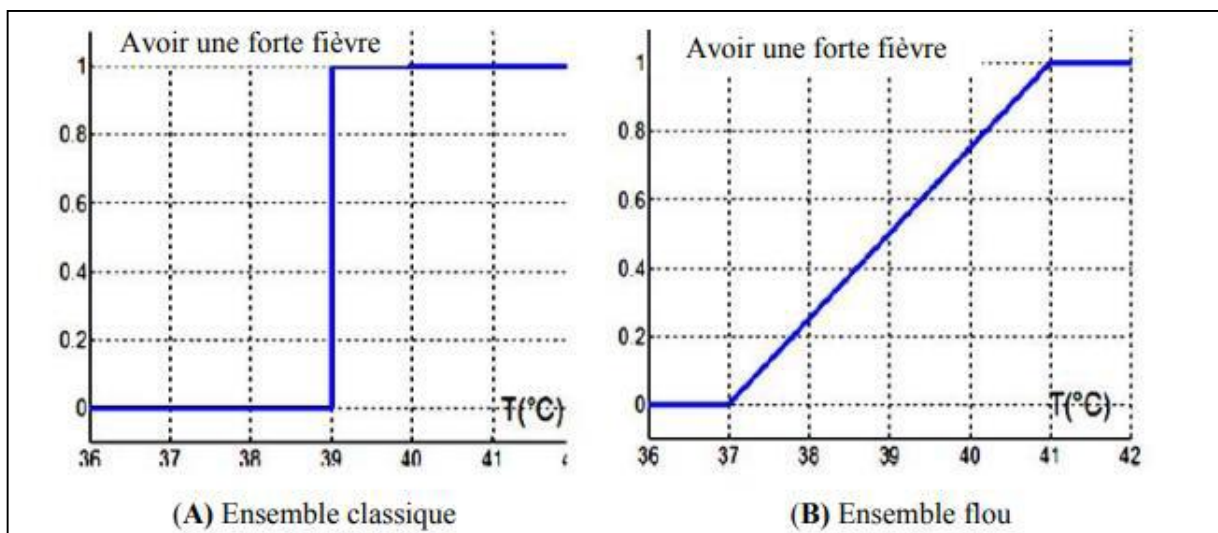


Figure II.16 : Représentation des ensembles

## II.7.6- REGULATION PAR LA LOGIQUE FLOUE

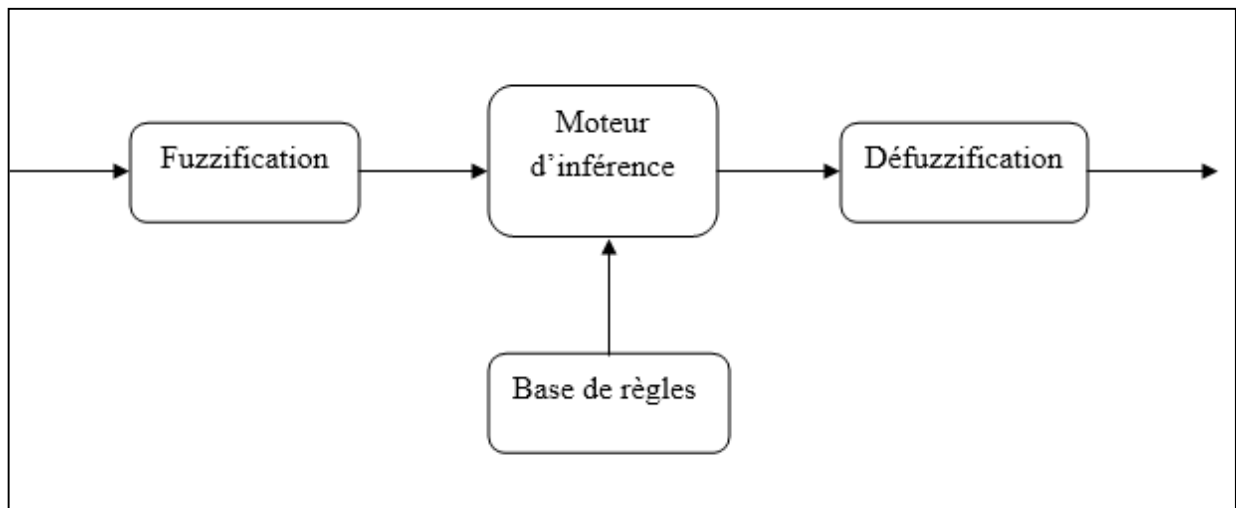
Contrairement à la régulation classique, la régulation par la logique floue ne traite pas une information comme une relation mathématique bien définie, mais elle utilise des inférences avec plusieurs règles.

## II.7.7 PRINCIPE DE LA COMMANDE FLOUE [31]

La structure de la commande floue, peut être décomposée en trois grands modules. Comme présenté en figure :

Le bloc d'entrée de fuzzification permet d'associer à chacune des entrées réelles, en utilisant les fonctions d'appartenance, avec un degré d'appartenance pour chacun des sous ensemble flous définis sur l'univers de discours. Le bloc représentant le moteur d'inférence constitue base de règles, de type « Si.... Alors ..... ».

Permettant de calculer les degrés d'appartenance aux sous ensemble flous correspondant à la commande du système. Le bloc de sortie de défuzzification, permet de transformer les degrés d'appartenance des sous ensemble de commande en grandeur numérique applicable à l'actionneur de contrôle.



*Figure II.17 : Structure générale d'une commande floue*

## II.8- REMARQUES

Suite à notre investigation sur les commandes peuvent être implantées dans un API, régulation PID qui a été conçu par un bloc s'appelle (FB 41). En outre la réalisation des autres méthodes de commande avancées comme la logique floue et la commande prédictive consiste à utiliser des langages de programmation avancées comme C++ et les appeler dans l'API.

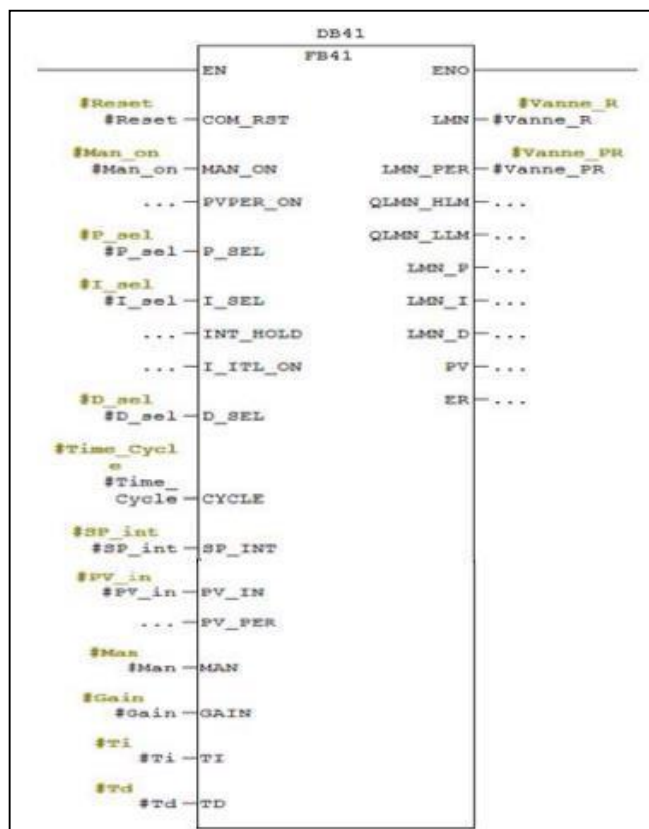


Figure II.18: Configuration de notre bloc FB 41

Comme la régulation par logique flou. Mettre en oeuvre du correcteur flou sous automate S7-300 , La programmation des fonctions d'appartenance de l'erreur, passe par l'utilisation de deux comparateurs, composant entre l'erreur stockée dans le double mot interne (MD15) et les différentes limites des intervalles, si la comparaison est vraie alors le programme de la figure (36) calcule l'appartenance de l'erreur qui sera chargée dans les doubles mots MD90 et MD165 en utilisant les blocs de soustraction et de division.[30]

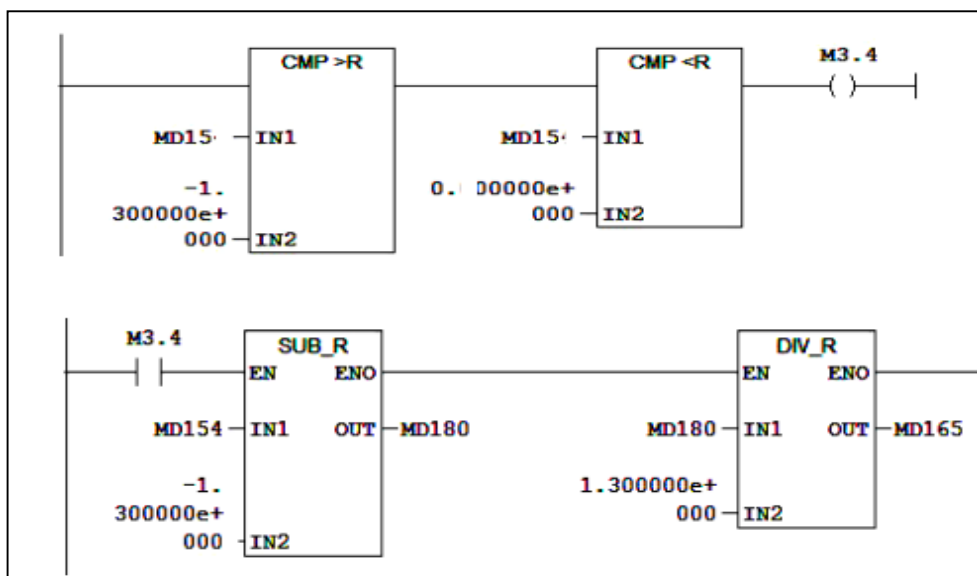
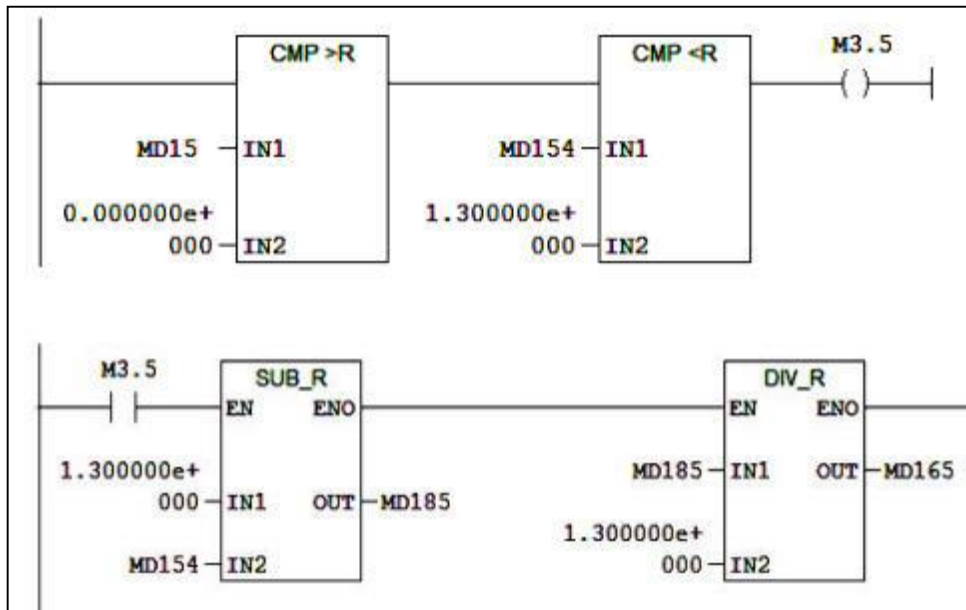


Figure II.19: Bloc de calcul de La partie croissante de la fonction d'appartenance





**Figure II.20: Bloc de calcul de la partie décroissante de la fonction d'appartenance**

Quant aux méthodes de contrôle prédictif et adaptatif, ces méthodes sont utilisées dans le domaine du contrôle de vitesse dans le domaine automobile. Elle est basée sur la technologie GPS. Des informations de carte routière détaillées sont utilisées pour connaître les conditions de conduite que va rencontrer le véhicule.

## II.9- CONCLUSION

Dans ce chapitre, nous avons présenté Les commandes avancées les plus utilisées dans l'acadimia et les différente techniques et méthodologie de régulation et asservissement peuvent s'effectué par API. Leurs caractéristiques et leur applications...et.

Dans la suite de ce mémoire, nous allons présenter à l'implémentation d'une régulation discontinue et régulation PID de débit et niveau d'un system hydraulique et comparaison entre les performances de chaque type de régulation. Simulation par S7-300 Simatic manager et développement d'une interface par Wincc flexible.

**CHAPITRE III : PROPOSITION D'UNE REGULATION DISCONTINUE ET REGULATION PID DE DEBIT ET NIVEAU D'UN SYSTEME HYDRAULIQUE**

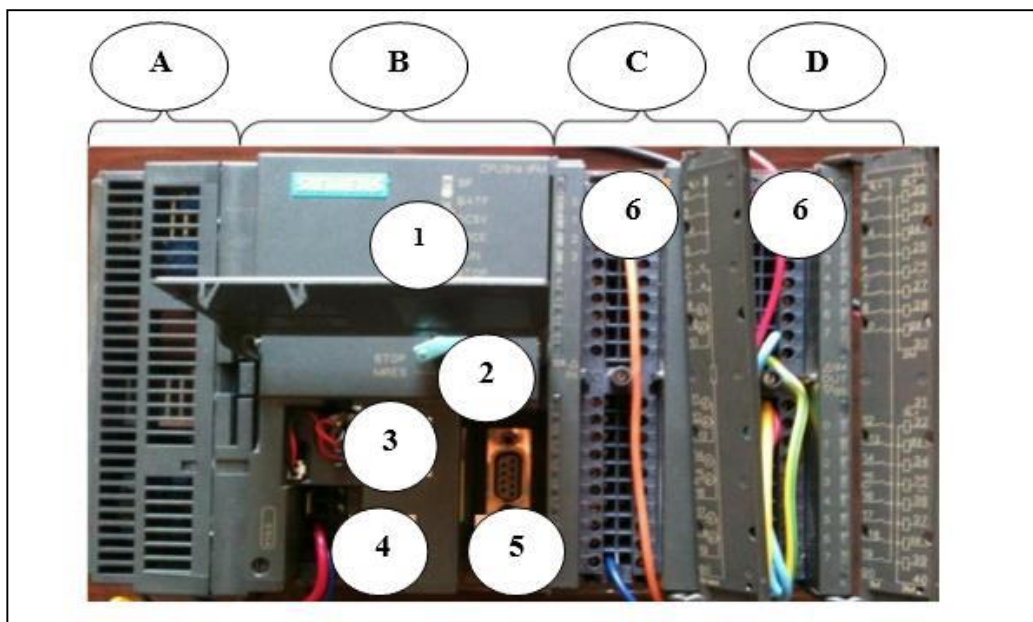
### III.1- INTRODUCTION

Les API ont pour fonctions de remplir des tâches de commande en élaborant des actions en suivant un algorithmiques appropriée à partir des informations données par des capteurs. Dans ce qui va suivre nous allons présenter l'étude du logiciel de programmation et de communication step7, ce dernier permet la programmation et la simulation de procédures automatisées avec différents langages normalisés.

L'objectif de ce chapitre est de présenté une description du matériel (API) et logiciels (step7/WinCC) utilisés pour l'implémentation des lois de commande pour réaliser une régulation discontinue et régulation PID de débit et niveau d'un system hydraulique, à base d'un API SIMATIC S7- 300 qui est programmé sous le logiciel Step7 ainsi que la conception d'une interface IHM de supervision en utilisant l'outil WinCC flexible.

### III.2- AUTOMATE S7-300 (CPU314)

L'automate s7-300 est l'un des automates les plus utilisés dans l'industrie grâce à son coût, ses avantages et ses caractéristiques. C'est un mini-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface Multipoint (MPI), PRO FIBUS et Industriel Ethernet (PROFINET).



*Figure III.1: SIMATIC S7-300 en détail*

*Tableau III.1 : Description des différents éléments de l'API S7 300*

|          |                                     |          |                                       |
|----------|-------------------------------------|----------|---------------------------------------|
| <b>1</b> | LED de visualisation d'état défaut  | <b>4</b> | Bornes pour la tension d'alimentation |
| <b>2</b> | Commutateur de fonctionnement       | <b>5</b> | interface multi point (MPI)           |
| <b>3</b> | Logement pour la pile de sauvegarde | <b>6</b> | Entrées /sorties intégrées            |
| <b>A</b> | Module d'alimentation 24v           | <b>C</b> | Module d'entrée intégré B             |
| <b>B</b> | La CPU 314                          | <b>D</b> | Module de sortie intégré              |

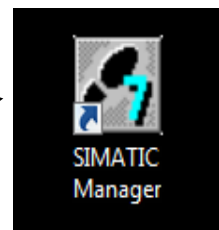
### III.3- CREATION D'UN PROJET ET CONFIGURATION MATERIELLE DANS STEP 7

Afin de créer un nouveau projet STEP7, il nous est possible d'utiliser « l'assistance de création de projet, ou bien de créer le projet soi-même et le configurer directement, cette dernière est un peu plus complexe, mais nous permet aisément de gérer notre projet. En sélectionnant l'icône SIMATIC Manager, on aura la fenêtre principale qui s'affiche, pour sélectionner un nouveau projet validé. Comme le projet est vide il nous faut insérer une station Chapitre III Les progiciels de programmation et de supervision, Step7 et WinCC flexible 30 SIMATIC 300/400.

#### III.3.1- CREATION D'UN PROJET DANS SIMATIC MANAGER

##### Création d'un projet dans SIMATIC Manager

- Ouvrir SIMATIC Manager.
- Cliquer sur l'icône
- 'SIMATIC Manager'
- Créer un nouveau Projet (→ Fichier → Nouveau).



Cliquer sur  
"Nouveau"

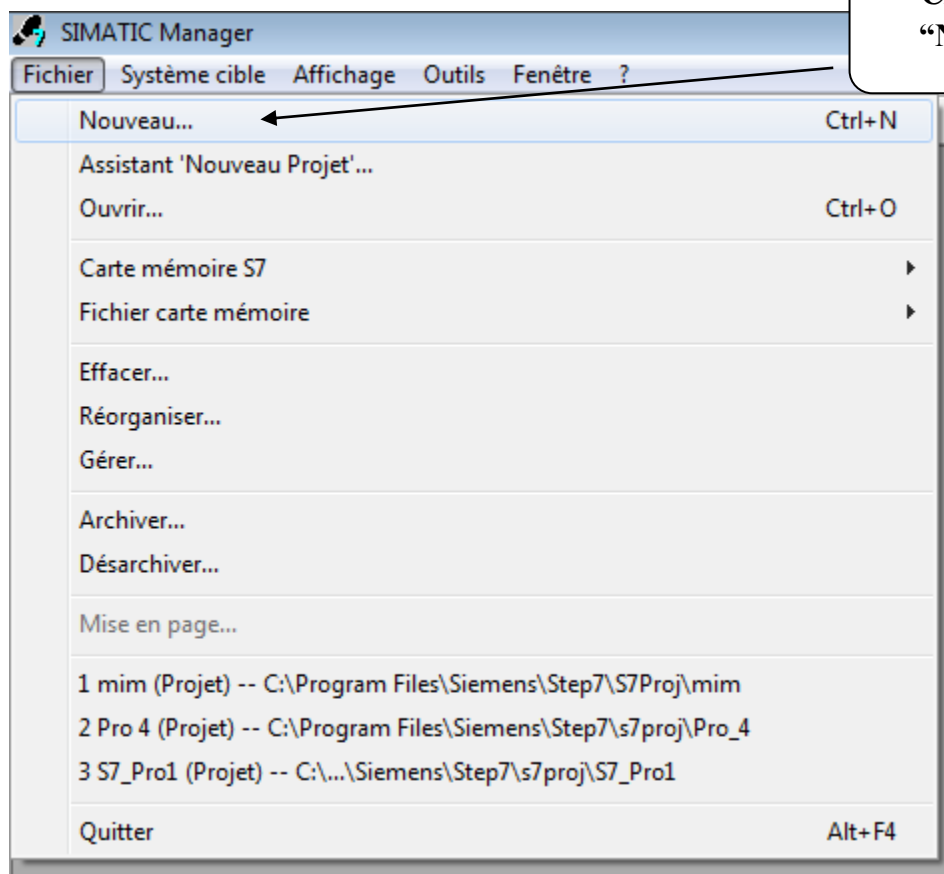
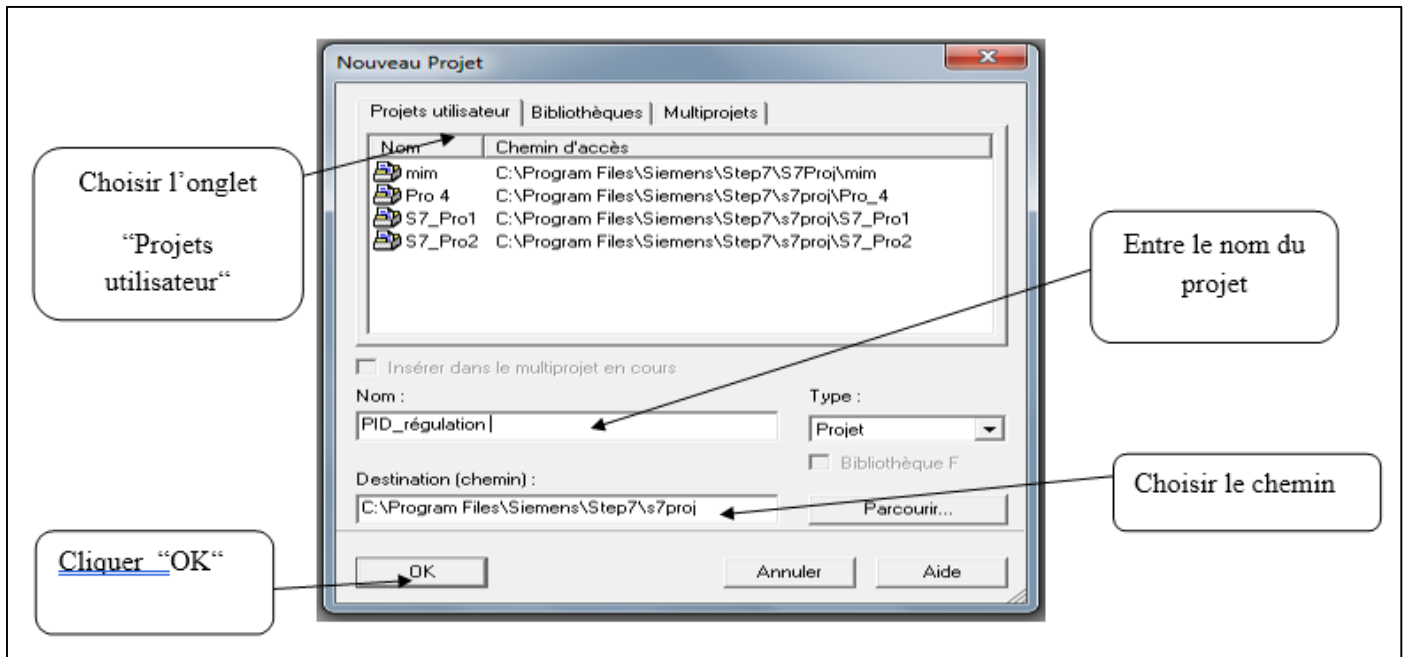


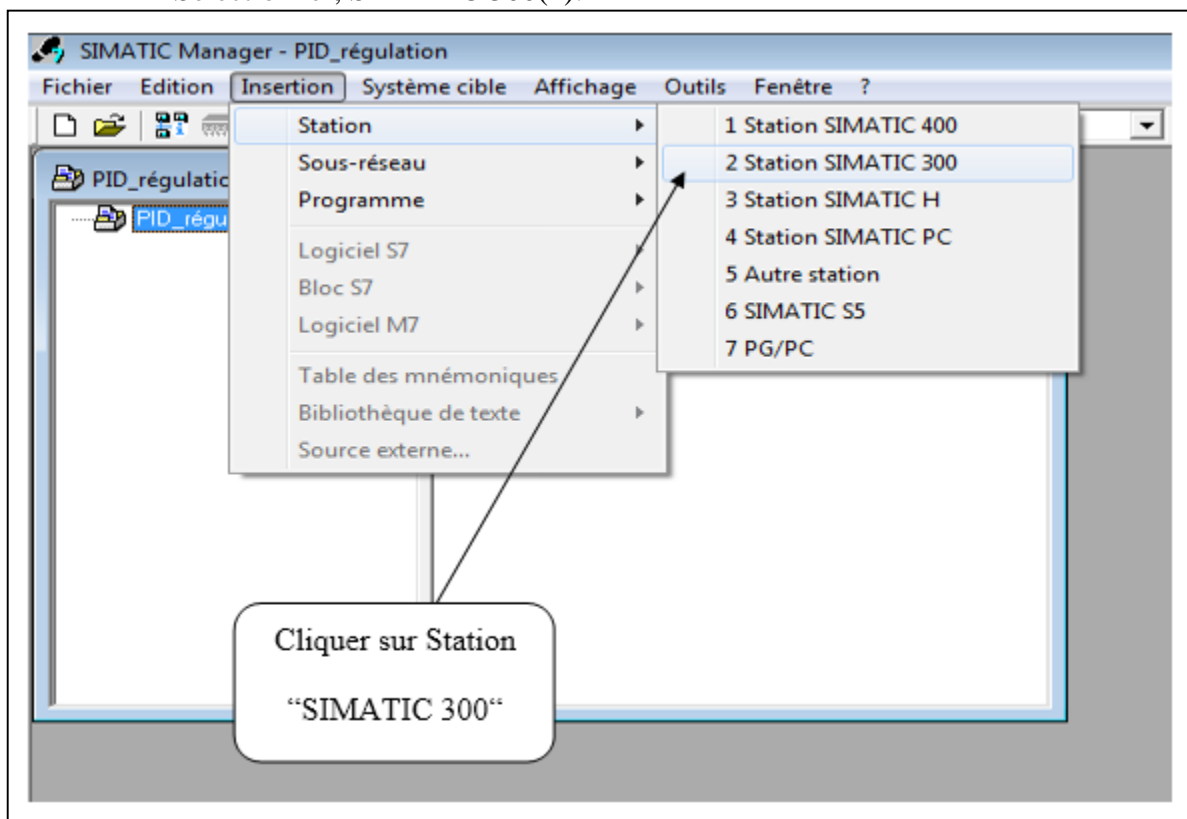
Figure III.2 : Création de projet 1

- Créer un nouveau projet, donner le chemin de sauvegarde ainsi que le nom du projet ( → Projets utilisateur → PID\_ régulation → OK).

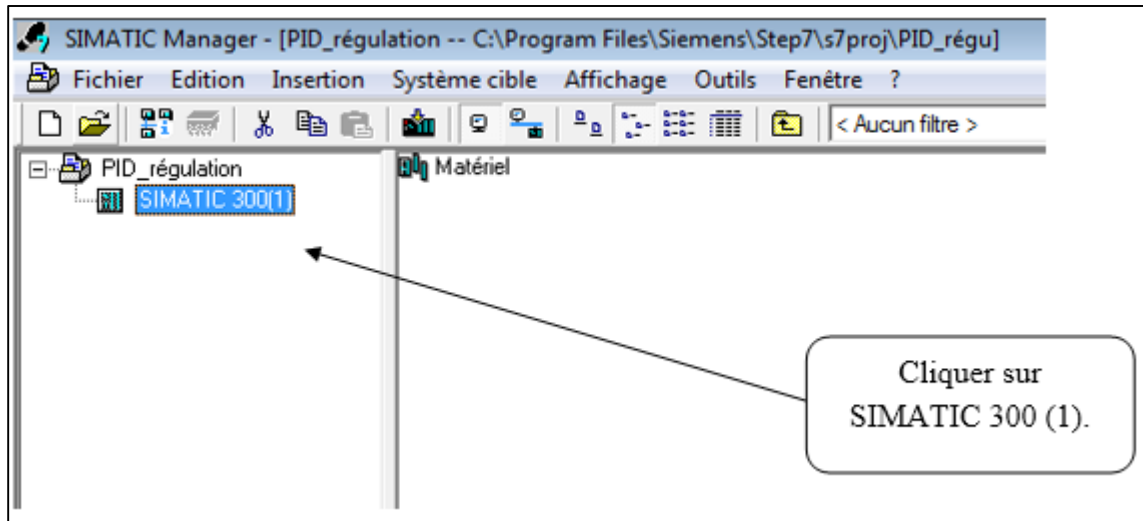


**Figure III.3 : Création de projet 2**

- Insérer un objet de type 'Station SIMATIC 300' (→ Insertion → Station → Station SIMATIC 300).
- Sélectionner, SIMATIC 300(1).



**Figure III.4 : SIMATIC 300 (1)**



**Figure III.5 : Station SIMATIC 300 (2).**

### **III.3.2- CONFIGURATION MATERIELS (HARDWARE)**

Dans une table de configuration, on définit les modules mis en œuvre dans la solution d'automatisation.

- Ouvrir le programme de configuration matérielle (→ Edition → Ouvrir un objet).
- Ouvrir le catalogue Hardware. Il se compose des répertoires suivants : PROFIBUS-DP, SIMATIC 300, SIMATIC 400 et SIMATIC PC Based Control, ainsi que tous les supports et composants modules d'interface nécessaires à la configuration de vos installations.
- Insérer un support profilé ( → SIMATIC 300 → RACK-300 → Profilé Support).

Il apparaît alors automatiquement un tableau pour la configuration du Rack300

- Choisir les composants présents sur votre rack dans le catalogue Hardware et les insérer dans le tableau de configuration. Pour se faire, il suffit de cliquer sur l'élément voulu et de le faire glisser à la place souhaitée dans le tableau.

Après que la configuration de matérielle ait été effectuée, il faut la compiler et la charge dans mémoire de l'automate en appuyant respectivement sur :

**Le bouton :**

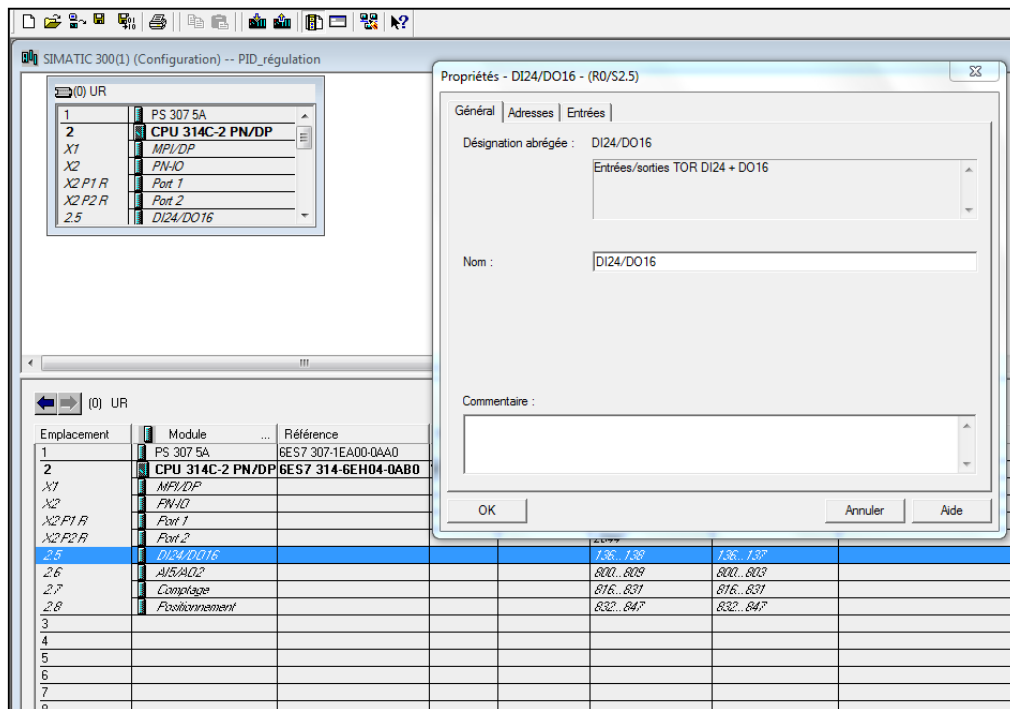


**Puis :**



### **III.3.3- MODULES D'E/S TOUT OU RIEN (TOR)**

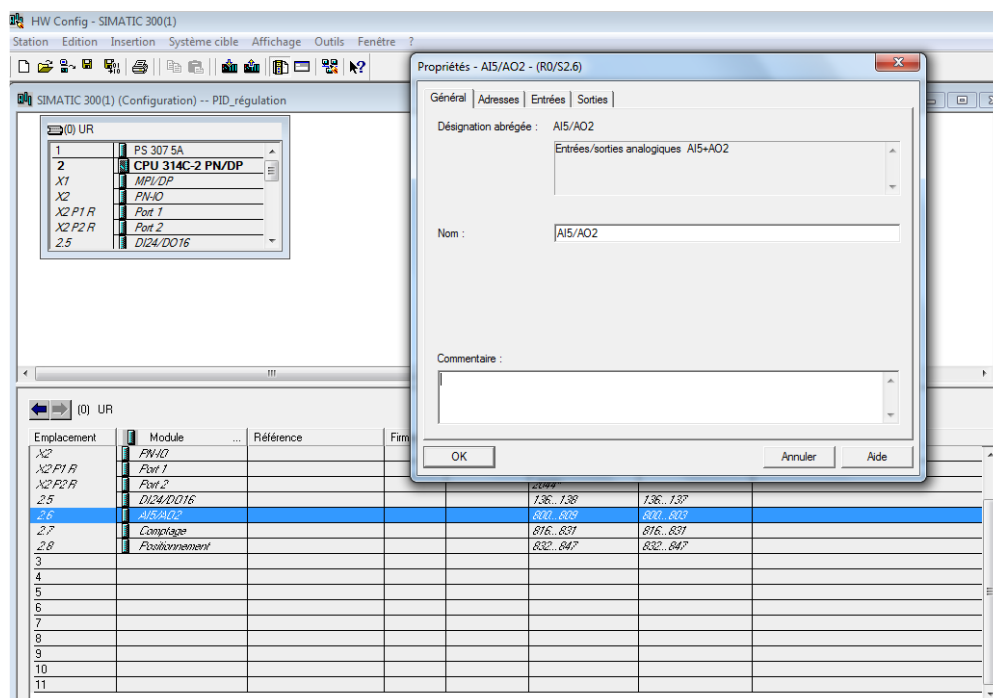
Les E/S (entrées / sorties) tout ou rien ne sont des composants qui génèrent ou réagissent à des signaux électriques tout ou rien normalisés. Les modules de sortie génèrent des signaux en réaction à des séquences du programme pour activer ou désactiver des actionneurs ou des pré-actionneurs. Les modules d'entrée, quant à eux, reçoivent les données et font évoluer le programme qui génère les sorties.



**Figure III.6 : Modules entrées et sorties TOR**

### III.3.4- MODULES D'E/S ANALOGIQUES

Les modules d'entrée et de sortie analogiques tout comme les modules tout ou rien ne génèrent et reçoivent des signaux. Les modules d'entrée analogiques reçoivent le signal analogique normalisé d'un capteur couplé à un transmetteur. Le signal est échantillonné et la valeur de l'échantillonnage est sauvegardée dans la mémoire de l'automate. Les modules de sortie recomposent un signal analogique à partir d'une valeur numérique sauvegardée dans la mémoire de l'automate. Ce dernier est transmis à l'actionneur.

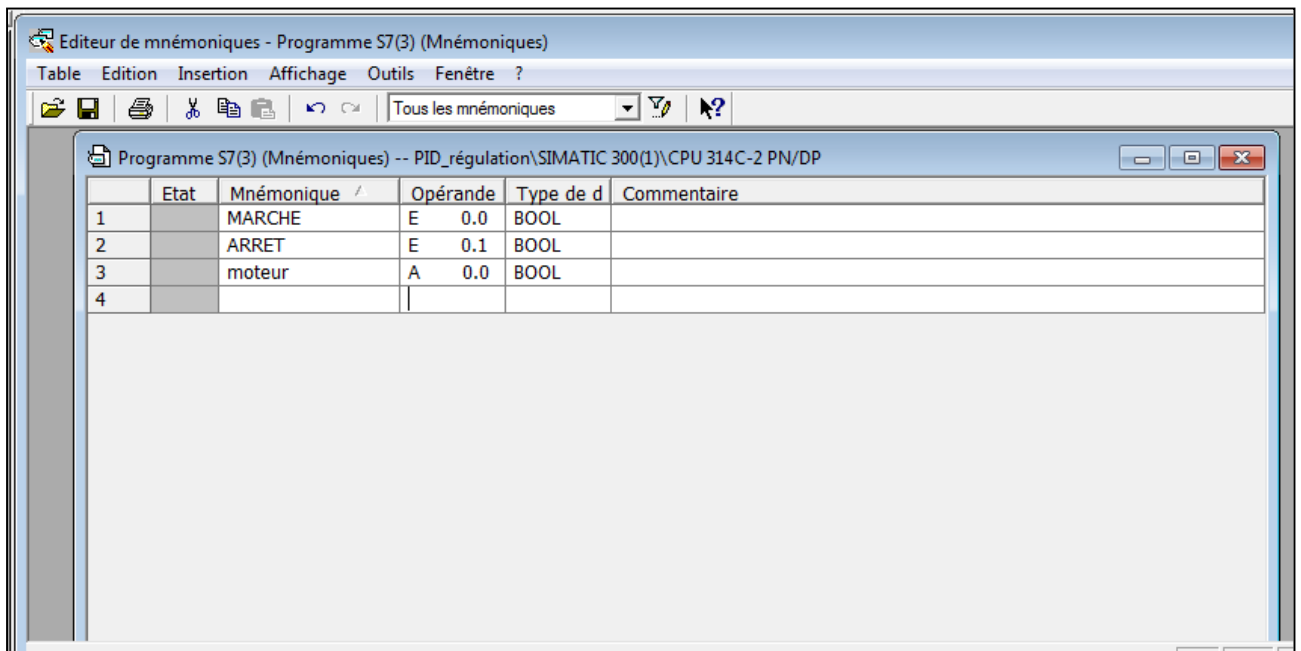


**Figure III.7: Modules entrées et sorties analogiques**

### Création de la table des mnémoniques

Dans tout programme il faut définir la liste des variables qui vont être utilisées lors de nos programmations. Pour cela la table des mnémoniques est créée. L'utilisation des noms appropriés rend le programme plus compréhensible est plus facile à manipuler. Ce type d'adressage est appelé (relatif). Pour créer cette table, on suit le cheminement suivant :

- Insérer nouvel objet table des mnémoniques. ;
- Editer la table des mnémoniques en respectant notre cahier de charge, pour les entrées et les sorties ;
- Pour trouver table mnémonique aller vers (Outils □ table mnémoniques).



*Figure III.8: Modules entrées et sorties analogiques*

### III.3.6- BLOCS DE PROGRAMMATIONS

En utilisant l'un des langages de programmation mis à disposition, on crée un programme affecté ou non à un module, qu'on enregistre sous forme de blocs, de sources ou de diagrammes. Le dossier bloc, cité auparavant, contient les blocs que l'on doit charger dans la CPU pour réaliser la tâche d'automatisation, il englobe :

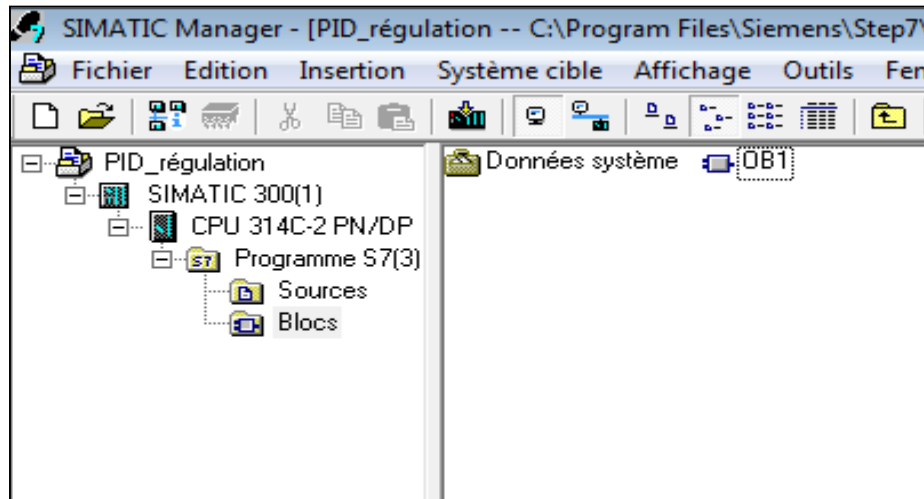
- Blocs de code (OB, FB, FC) qui contiennent les programmes. [34] ;
- Blocs de données DB d'instance et DB globaux qui contiennent les paramètres du programme.

#### III.3.6.1- BLOCS D'ORGANISATION (OB)

Un OB est appelé cycliquement par le système d'exploitation et réalise ainsi l'interface entre le programme utilisateur et le système d'exploitation. Le dispositif de commande est informé dans cet OB par des commandes d'appel de blocs, de quels blocs de programme il doit traiter pour que le traitement du programme démarre, le projet doit posséder au moins un



OB. Les blocs d'organisation définissent l'ordre (événements de déclenchement) dans lequel les différentes parties du programme sont traitées. L'exécution d'un OB peut être interrompue par Appel d'un autre OB. Cette interruption se fait selon la priorité des OB. Le bloc OB1 est généré automatiquement lors de la création d'un projet. C'est le programme cyclique appelé par le système d'exploitation.



**Figure III.9: Bloc d'organisation (OB).**

### **III.3.6.2- BLOCS FONCTIONNELS (FB),(SFB)**

Le FB est un sous-programme écrit par l'utilisateur et exécuté par des blocs de code. On lui associe un bloc de données d'instance relatif à sa mémoire et contenant ses paramètres. Un bloc fonctionnel système (SFB) est un bloc fonctionnel intégré à la CPU S7. Comme les SFB font partie du système d'exploitation, ils ne sont pas chargés en tant que partie du programme. Pareil aux FB, les SFB sont aussi des blocs avec mémoire. Il faut donc également créer pour les SFB des blocs de données en tant que partie du programme. Les CPU S7 proposent des SFB : [34]. :

- Pour la communication via des liaisons configurées.
- Pour des fonctions spéciales intégrées (par exemple, SFB29 HS\_COUNT dans la CPU 312 IFM et la CPU 314 IFM).

### **III.3.6.3- FONCTIONS (FC)**

Les (FC) sont des fonctions permettant la programmation des sous-programmes. Le plus souvent on les appelle par l'intermédiaire de l'OB1. À l'intérieur du bloc des variables temporaires pour ces calculs et aussi pour l'affectation de ces E/S. En appelant ce FC dans un autre endroit du programme (FC, OB, ...), alors celui-ci exécutera son contenu à l'aide de ses paramètres d'entrées et modifiera des valeurs de sorties. On raccordera aux E/S du FC les variables faisant partie de notre table de mnémonique.

### **III.3.6.4- BLOCS DES DONNEES (DB)**

Les blocs de données (DB) permettent de stocker des valeurs dans une base de données. Un DB peut être de deux types différents :

- DB d'instance (propre à un FB).

• DB globale (interrogeable dans le programme dans n'importe quel bloc). Le nombre et la taille des DB ne sont pas fonction de la CPU mais de la taille mémoire utilisateur.

En effet les DB prennent de la place mémoire, de la même manière que les FC, FB, OB. On peut y stocker tous types de variables: BOOL, BYTE, WORD, INT, DWORD, DINT, REAL....

**III.3.6.5 - BLOC FC 105**

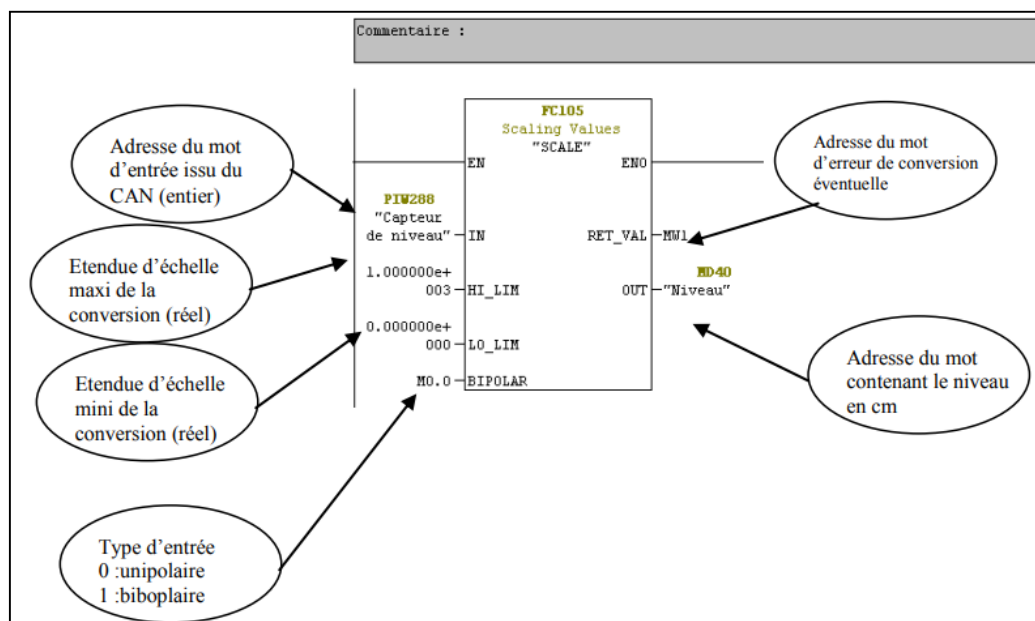
La fonction mise à l'échelle (SCALE) prend une valeur entière (IN) et la convertit en une valeur réelle exprimée en unités physiques, comprises entre une limite inférieure (LO\_LIM) et une limite supérieure (HI\_LIM). Les valeurs d'entrée sont comprises entre deux valeurs K1 et K2:

- Si l'entrée est Bipolaire : La valeur entière d'entrée est supposée être comprise entre -27648.0 et 27648.0, donc : K1 = -27648.0 et K2 = +27648.0
- Si l'entrée est unipolaire : La valeur entière d'entrée est supposée être comprise entre 0.0 et 27648.0, donc : K1 = 0.0 et K2 = +27648.0.

Si la valeur entière d'entrée est supérieure à K2, la sortie (OUT) est saturée à la valeur la plus proche de la limite supérieure (HI\_LIM) et une erreur est signalée. Si la valeur entière d'entrée est inférieure à K1, la sortie est saturée à la valeur la plus proche de la limite inférieure (LO\_LIM) et une erreur est signalée [32].

**Les paramètres de la FC105 :**

- **EN** : Alimentation du module.
- **IN** : Valeur d'entrée à convertir.
- **HI\_LIM** : limite supérieure en unité physique.
- **LO\_LIM** : limite inférieure en unité physique.
- **BIPOLAIRE** : L'état de signal "1" signifie que la valeur d'entrée est bipolaire et l'état de signal "0" qu'elle est unipolaire.
- **OUT** : Résultat de la conversion d'échelle.



**Figure III.10: Fonction mise à l'échelle (SCALE)**

### III.3.6.5 - BLOC FC 106

C'est l'inverse de Bloc SCALE, cela transfère une valeur numérique entière (entre 0 et 27648) pour la sortie analogique.

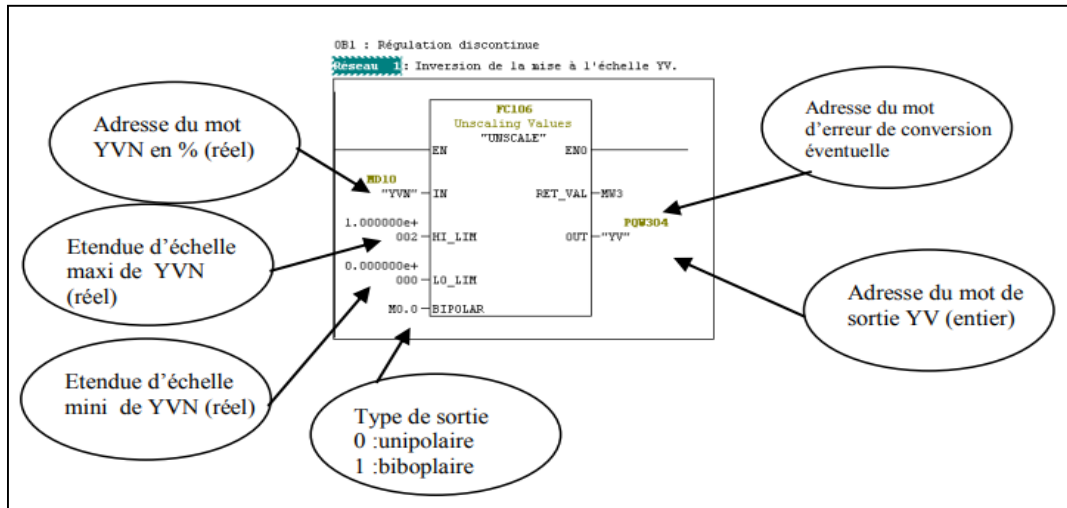


Figure III.11: Fonction mise à l'échelle (UNSCALE).

## III.4- PROGICIEL DE CONCEPTION ET CONFIGURATION D'IHM WINCC FLEXIBLE

### III.4.1- PRESENTATION GENERALE

Win CC flexible est un progiciel qui partage l'environnement STEP7 et offre une gamme de systèmes d'ingénierie évolutifs pour les tâches de configuration pour la configuration de différents pupitres opérateur. Créer des interfaces graphiques et des variables signifie pouvoir lire des valeurs de processus via l'automate, les afficher pour que l'opérateur puisse les interpréter, éventuellement ajuster le processus, toujours via l'automate. Alors que la complexité des processus augmente et que les machines et installations doivent répondre à des spécifications fonctionnelles de plus en plus strictes, les opérateurs exigent une transparence maximale. Cette transparence est obtenue grâce à l'interface homme-machine (IHM). Siemens lance SIMATIC Win CC flexible en tant que progiciel pour la conception et la configuration d'IHM's.[33]

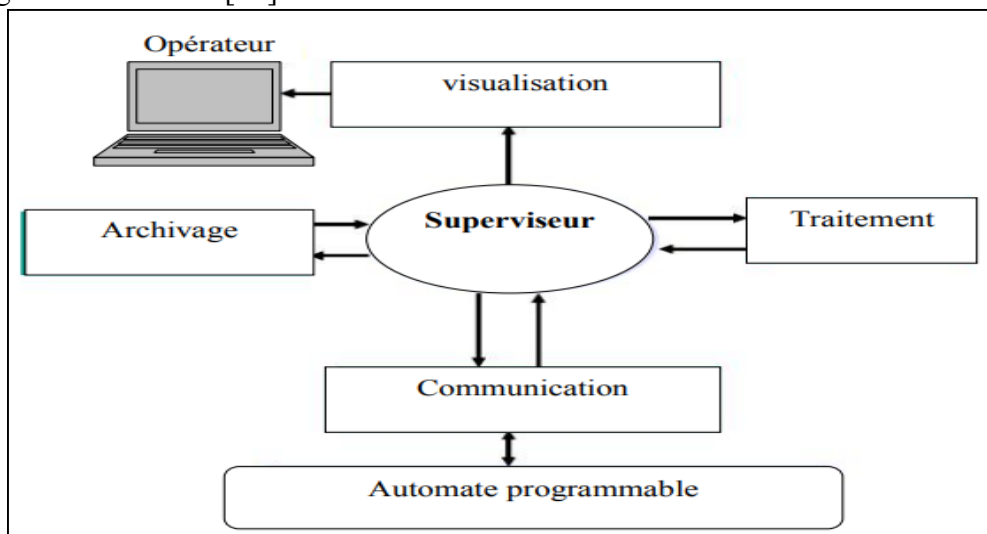


Figure III.12: Modules fonctionnels d'un système de supervision

Un système HMI se charge des tâches suivantes :

- Représentation du processus ;
- Commande du processus ;
- Vue des alarmes ;
- Archivage de valeurs processus et d'alarmes ;
- Documentation de valeurs processus et d'alarmes.
- Gestion des paramètres du processus et de machine.

### III.4.2- PRINCIPALES ETAPES SUIVIES POUR CREER NOTRE APPLICATION SOUS WINCC

- Créer un projet ;
- Sélectionner et installer l'API ;
- Définir les variables dans l'éditeur de variables ;
- Créer et éditer les vues dans l'éditeur graphics designer ;
- Paramétrer les propriétés de WINCC runtime ;
- Activer les vues dans le WINCC runtime ;
- Utiliser le simulateur pour tester les vues du processus.

### III.4.3-COMMUNICATION ENTRE WINCC ET AUTOMATE PROGRAMMABLE INDUSTRIEL (API)

Dans le cadre de l'échange de données, les variables de processus sont le lien entre les données WINCC et automates programmables. Chaque variable de processus WINCC correspond à Valeur processus déterminée dans la mémoire d'un des automates programmables lié. Au runtime, WINCC lit la zone de données de l'API, où La valeur de processus est stockée, déterminant ainsi la valeur de la variable de processus.

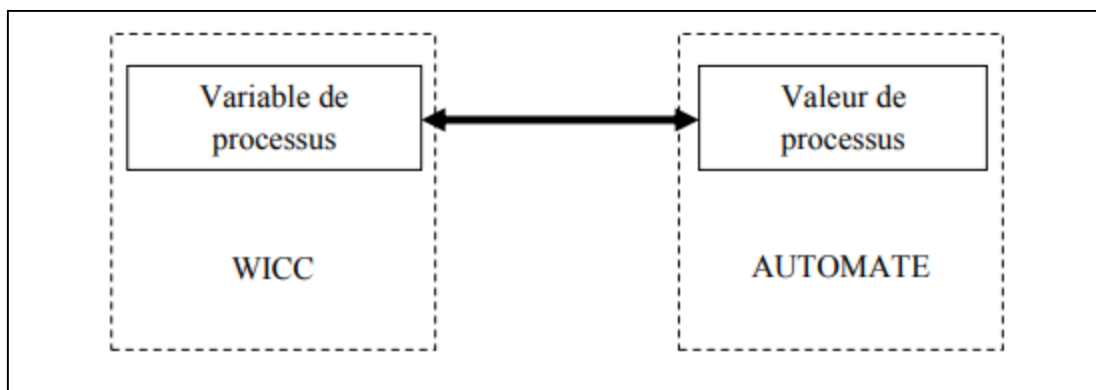


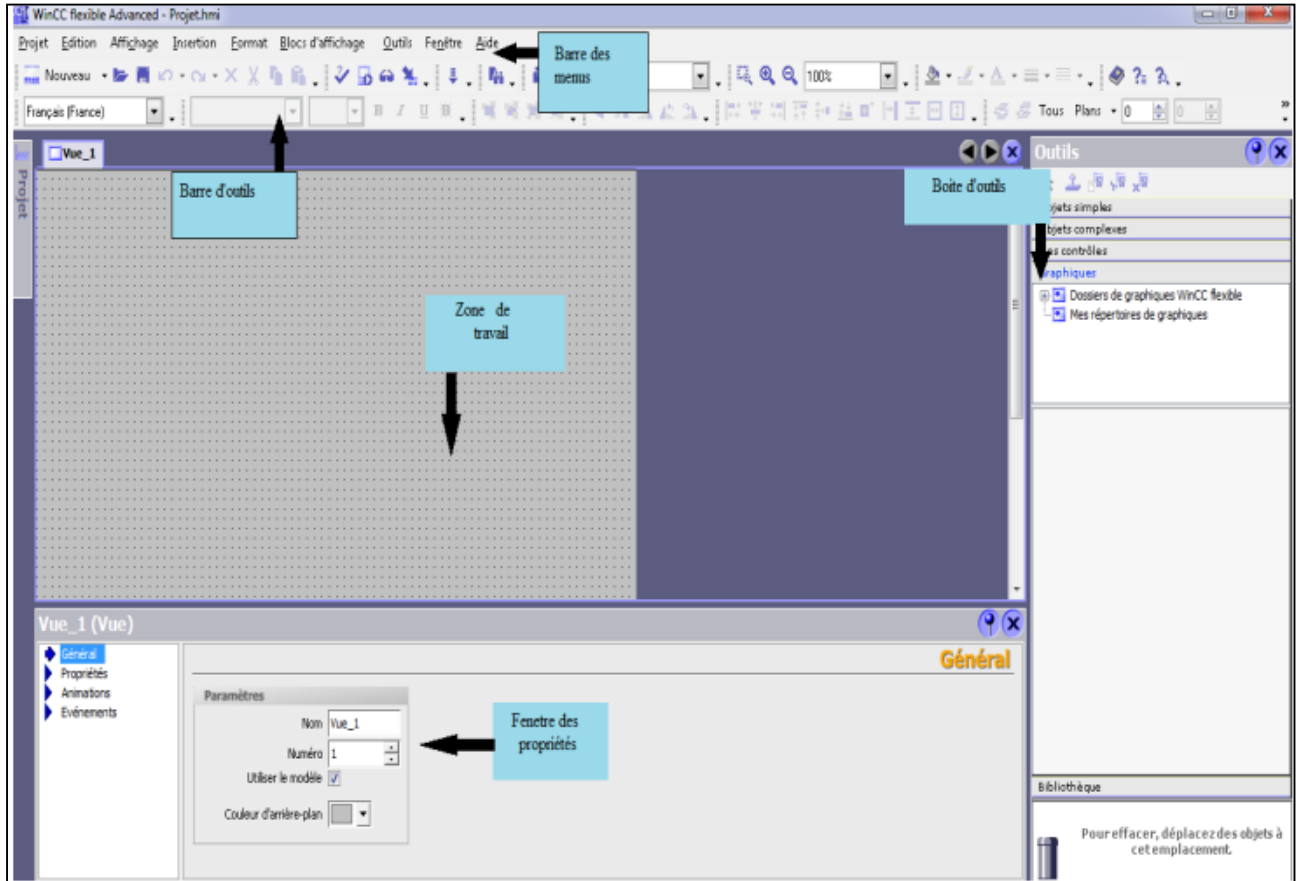
Figure III.13: Communication entre WINCC et automate programmable industriel (API)

### III.4.4- ELEMENTS DU PROGICIELS WINCC FLEXIBLE

L'environnement de travail de Win CC flexible se compose de plusieurs éléments. Certains de ces éléments sont liés à des éditeurs particuliers, visibles uniquement lorsque cet éditeur est actif. Il met à disposition un éditeur spécifique pour chaque tâche de configuration. On peut configurer l'interface utilisateur graphique d'un pupitre operateur avec l'éditeur Vues. Pour la configuration d'alarmes, on peut utiliser l'éditeur « Alarme TOR » ou autres.

Dans Win CC flexible, chaque projet créé contient principalement des vues que l'on crée pour le contrôle commande des machines et installations. Lors de la création des vues, nous disposons d'objets prédéfinis permettant d'afficher des procédures et de définir des valeurs du processus [33].

Les différents outils et barres de l'éditeur Vues sont représentés dans la figure qui suit Barre de menus.



**Figure III.14: Vue d'ensemble du logiciel Win cc flexible**

Il contient toutes les commandes nécessaires à l'utilisation de Win CC flexible. Les raccourcis disponibles sont indiqués en regard de la commande du menu :

- **Barre d'outils** : La barre d'outils permet de tout afficher dans le programmeur.
- **Zone de travail** : La zone de travail sert à configurer des vues, de façon qu'elles soient les plus compréhensibles par l'utilisateur, et très faciles à manipuler pour consulter les résultats.
- **Boite à outils** : La fenêtre des outils propose un choix d'objets simples ou complexes qu'on insère dans les vues, par exemple, des objets graphiques et éléments de commande.
- **Fenêtre des propriétés** : Le contenu de la fenêtre des propriétés dépend de la sélection actuelle dans la zone de travail, lorsqu'un objet est sélectionné, on peut éditer les propriétés de l'objet en question dans la fenêtre des propriétés.

### III.4.5- CREATION DE L'INTERFACE WINCC

Dans WinCC flexible, nous allons créer des vues pour la commande de machines et d'installations. Pour créer la vue, nous avons des objets prédéfinis pour représenter notre installation, afficher le processus et définir les valeurs de processus.[33]

➤ **Les Alarmes :** Les alarmes affichent des événements ou des états de fonctionnement qui se produisent dans une installation industrielle. Les alertes peuvent être utilisées pour diagnostiquer les erreurs. Il existe deux types d'alarmes :

**a) Alarmes TOR :** elles indiquent les modifications d'état (vrai ou faux) et elles sont déclenchées par les E/S. une alarme indique si une vanne est ouverte ou fermée par exemple.

**b) Alarmes analogiques :** elles indiquent des dépassements de limites. Une alarme analogique est déclenchée si le régime d'un moteur est inférieur à une valeur donnée par exemple.

### III.5- REALISATION DE PROJET

#### III.5.1- DESCRIPTION DE L'UNITE DE NIVEAU PUL-2 [31]

La station de régulation de niveau / débit PUL-2 (figure 51) est constituée d'une cuve graduée pour les lectures de niveau (La plage du niveau va de 0 à 500mm) et d'un réservoir de stockage servant à extraire le liquide et le recueillir dans la cuve de régulation, une pompe à palette actionnée par une pompe qui fournit de liquide (eau) nécessaire pour atteindre le niveau souhaitée et le maintenir.

Le système possède aussi une électrovanne proportionnelle à commande électrique qui fonctionne entre [0-10v] et une vanne **TOR** (tout ou rien) qui est alimentée avec 24(Vcc).

Un transducteur de niveau contient un capteur de pression situé au fond de la cuve de processus. L'unité PUL-2 est équipée aussi d'une vanne manuelle, agit sur le débit de sortie, permettent d'imposer la "charge" et ses variations.

L'unité de processus PUL-2 est équipée de plus, d'un débitmètre à aubes, permettant de réaliser des études sur les paramètres débit de liquide de l'unité. La cuve est transparente pour des raisons didactiques.

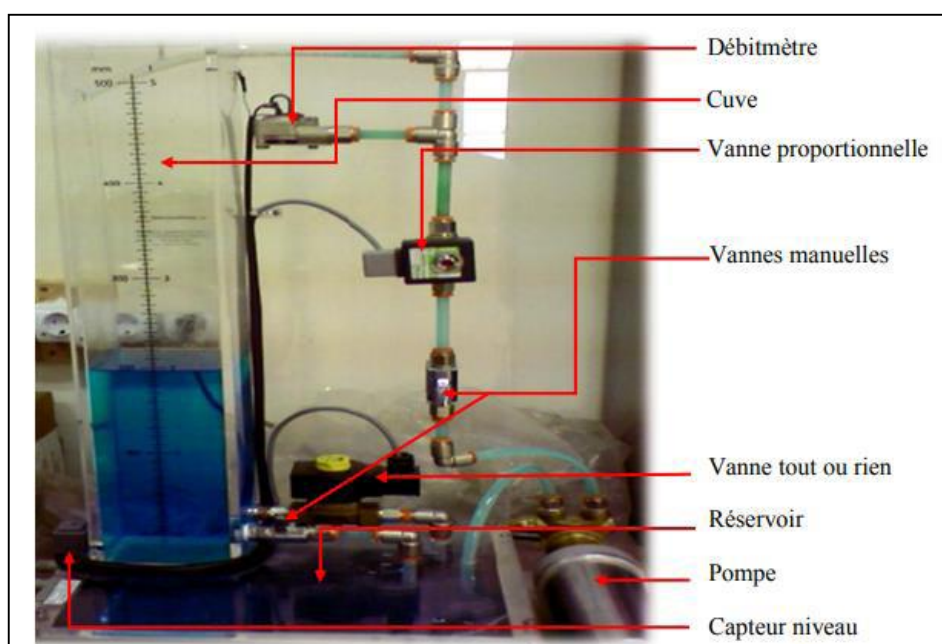
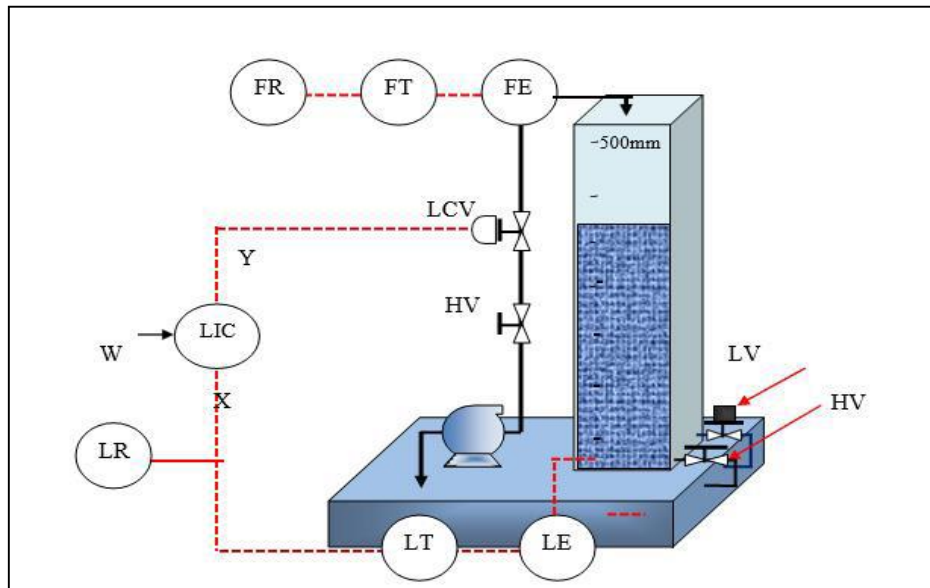


Figure III.15: Station de niveau PUL-2

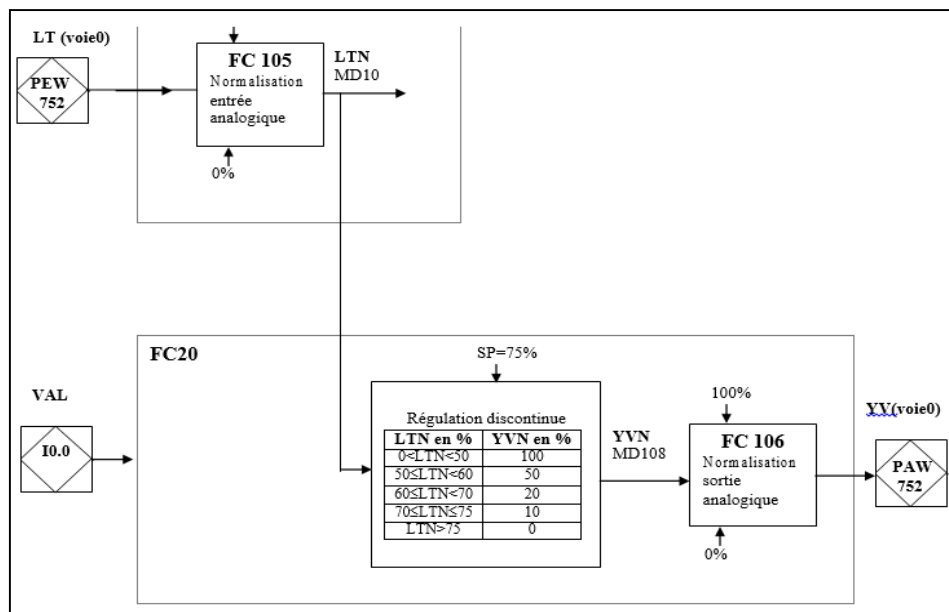


**Figure III.16: Schéma P&ID de la station de niveau**

- FE : Capteur de débit (Flow Element) ;
- FT : Transmetteur de débit (Flow transmitter). FR : Enregistreur de débit (Flow Recorder) ;
- LIC : Régulateur indicateur de niveau (Level Indicator Controller). LE : Capteur de niveau (Level Element) ;
- LT : Transmetteur de niveau (Level Transmitter) LR : Enregistreur de débit (Level Recorder) ;
- LCV : Vanne proportionnelle (Level Controller Valve) HV : Vanne manuelle (Hand Valve).

### III.5.2-CREATION DE PROGRAMME DE REGULATION DISCONTINUE

#### 3.5.2.1- STRUCTURE DU PROGRAMME

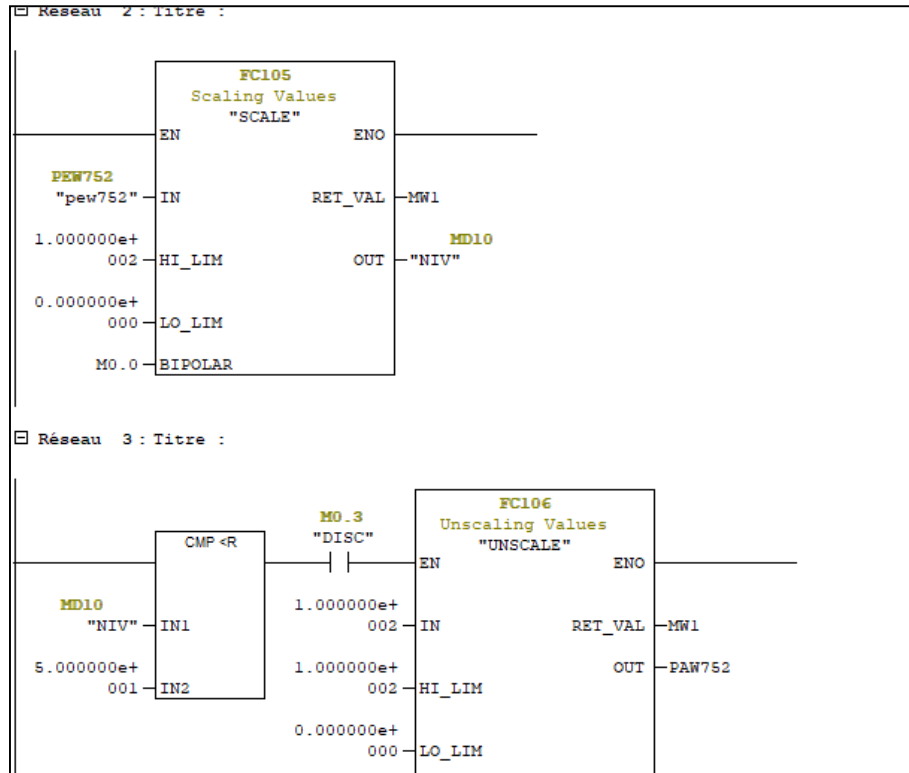


**Figure III.17: Programme à créer**

En dessous nous allons présenter la régulation discontinue selon le concept de la figure (III.17)

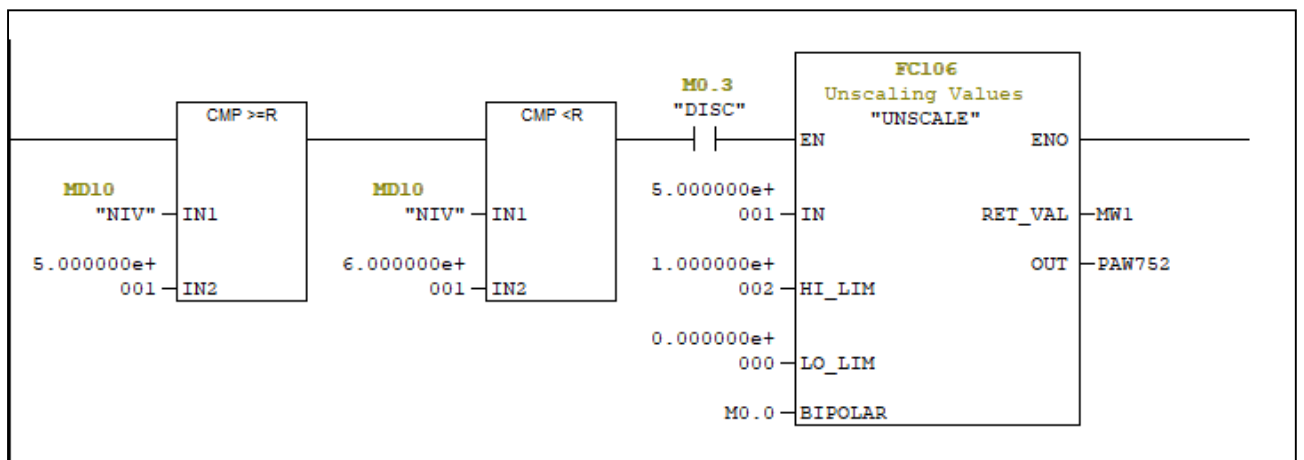
**III.5.2.2- PROGRAMME SUR STEP7**

1-Pour un niveau 50% : Ce bloc va prend de une valeur entier et la convertit en valeur réelle et le comparer avec une consigne pour donner commander a la vanne :



**Figure III.18: Programmation sous step7**

2- Pour un niveau de  $50 < LTN < 60$  :



**Figure III.19: Programmation sous step7-1**



3- Pour un niveau de  $60 \leq LTN < 70$  :

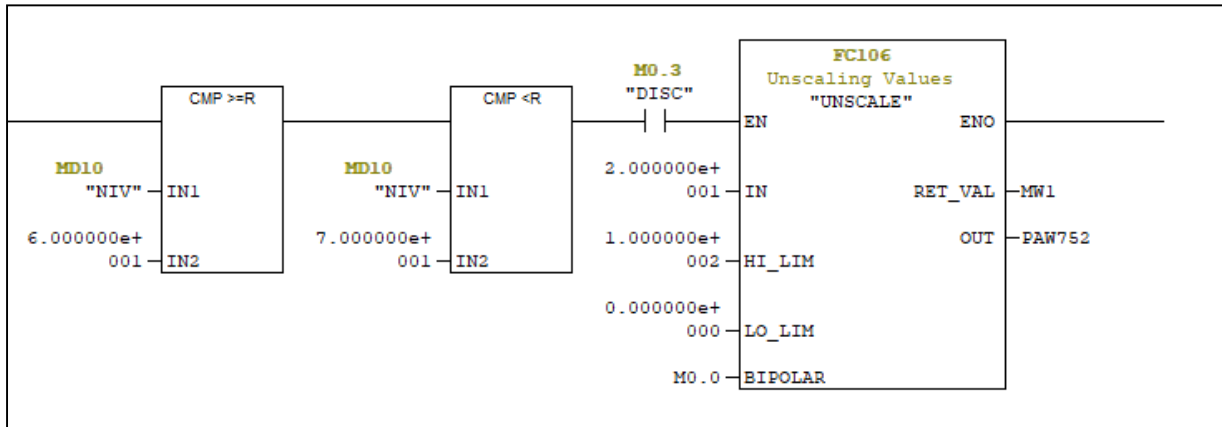


Figure III.20: Programmation sous step7-2

4- pour un niveau de  $70 \leq LTN \leq 75$  :

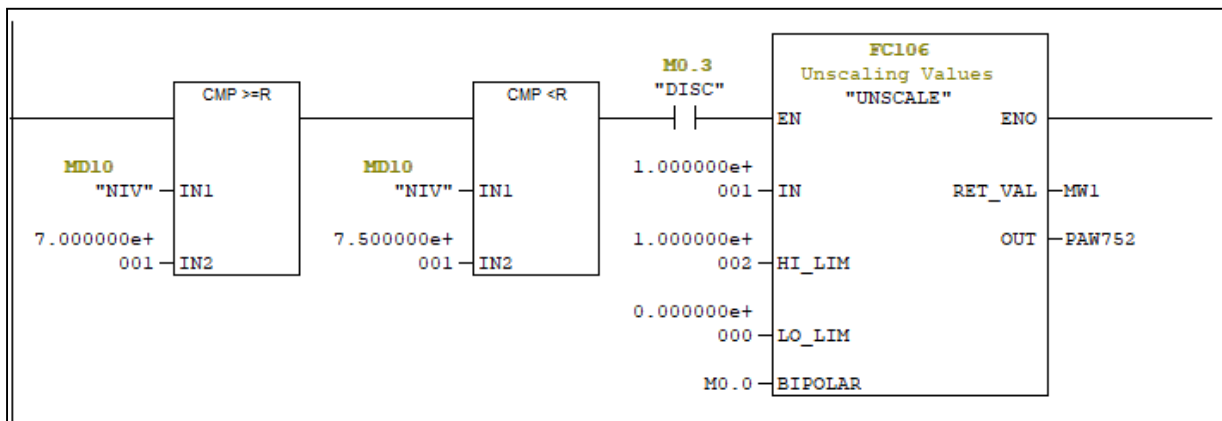


Figure III.21: Programmation sous step7-3

5- pour un niveau de  $LTN > 75$  :

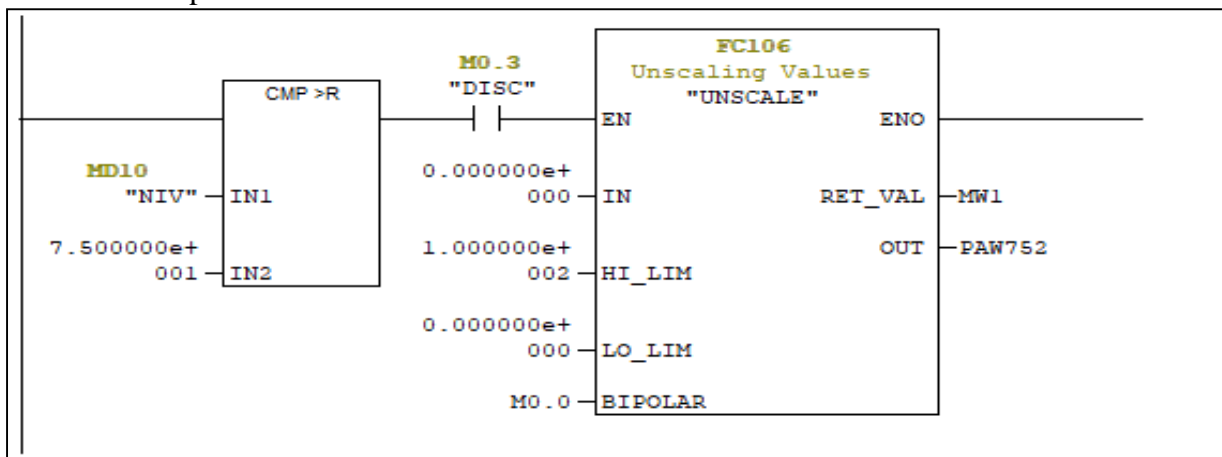


Figure III.22: Programmation sous step7-4

## III.6-CREATION DE PROGRAMME DE REGULATION PID

### III.6.1 PROGRAMMATION DE LA REGULATION PID EN UTILISANT LE BLOC FB 41 INTEGRE [35]

Le logiciel de programmation Step7 offre des blocs fonctionnels (FB) de régulation PID comprennent les blocs pour :

- La régulation continue ;
- La régulation pas à pas ;
- La modulation de largeur d'impulsion.

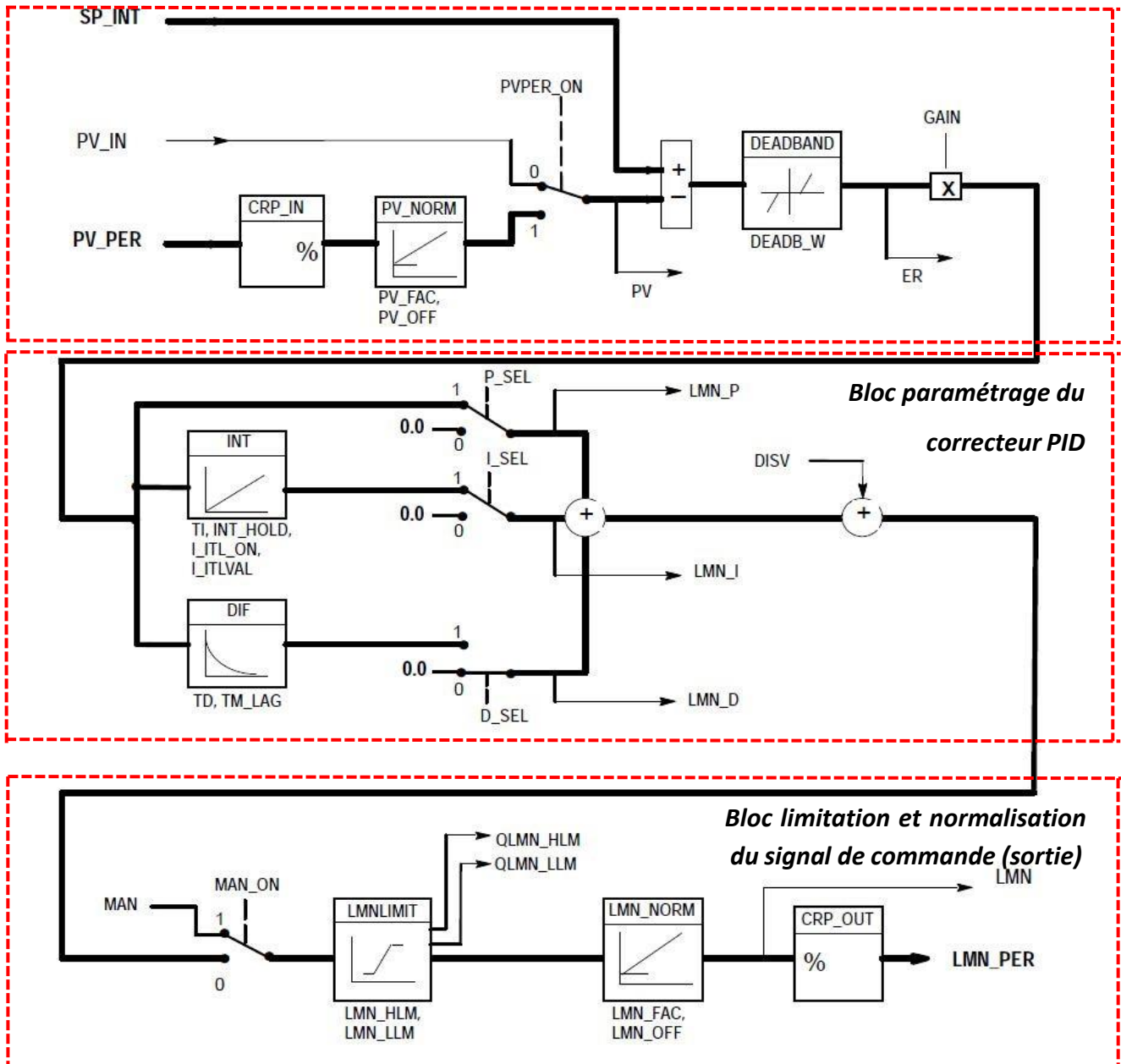
*Tableau III.2 : Différents types de régulation PID sous step7*

| Type de régulation   | Bloc de régulation | Type de la sortie de régulation |
|----------------------|--------------------|---------------------------------|
| Continue             | FB41 (DB41)        | Analogique                      |
| pas à pas            | FB42(DB42)         | Impulsion                       |
| largeur d'impulsions | FB43(DB43)         | Analogique                      |

Les FB de régulation proposent une régulation purement logicielle, c'est-à-dire qu'un bloc contient toutes les fonctions du régulateur. Les données nécessaires au calcul cyclique sont stockées dans des blocs de données associés OB, les blocs de données d'instance, ce qui permet aux FB de les appeler plusieurs fois. Le bloc FB41 sert à réguler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage du bloc FB 41 nous permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système régulé. Les fonctions les plus importantes sont : la consigne et la mesure, le bloc FB41 réalise un PID prêt à l'emploi avec une sortie continue et possibilité d'ajuster manuellement la valeur de sortie.

### III.6.2- FONCTIONNEMENT DE BLOC PID (FB41) [35]

Dans la figure suivante représente le schéma du principe représentant le fonctionnement général d'un bloc PID (FB41) :



**Figure III.23: Schéma fonctionnel de bloc PID (FB41)**

Le bloc de la figure III.23) est constitué de trois sous blocs, un bloc de calcul et de conditionnement du signal d'erreur conditionné passe par le bloc du paramétrage du régulateur PID (Ki, Td, Ti), le signal de sortie du bloc sera normalisé et limité dans le bloc de sortie afin d'être appliqué à l'actionneur proportionnel.

### III.6.3- PARAMETRAGE DU REGULATEUR PID

Les principaux paramètres du régulateur PID dont on aura besoin sont cités dans les deux tableaux(8) et (9) suivants :

Le tableau III.3 : suivant représente les paramètres d'entrée et types de données de réglage du régulateur PID :

| Paramètres | Type de données | Description   |
|------------|-----------------|---|
| COM_RST    | BOOL            | <b>COMPLETE RESTART</b> / Démarrage<br>Le bloc renferme un sous-programme de démarrage qui est exécuté quand cette entrée est à 1.  |
| P_SEL      | BOOL            | <b>PROPORTIONAL ACTION ON</b> / Activation de l'action proportionnelle<br>Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action P est active quand cette entrée est à 1. |
| I_SEL      | BOOL            | <b>INTEGRAL ACTION ON</b> / Activation de l'action par intégration<br>Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action I est active quand cette entrée est à 1.     |
| D_SEL      | BOOL            | <b>DERIVATIVE ACTION ON</b> / Activation de l'action par dérivation<br>Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action D est active quand cette entrée est à 1.    |
| CYCLE      | TIME            | <b>SAMPLE TIME</b> / Période d'échantillonnage<br>Le temps s'écoulant entre les appels de bloc doit être constant. Il est indiqué au niveau de cette entrée.  |
| SP_INT     | REAL            | <b>INTERNAL SETPOINT</b> / Consigne interne<br>Cette entrée sert à introduire une valeur de consigne.   |
| PV_PER     | WORD            | <b>PROCESS VARIABLE PERIPHERIE</b> / Mesure de périphérie<br>La mesure en format de périphérie est appliquée au régulateur par cette entrée.  |
| GAIN       | REAL            | <b>PROPORTIONAL GAIN</b> / Coefficient d'action proportionnelle<br>Cette entrée indique le gain du régulateur.  |
| TI         | TIME            | <b>RESET TIME</b> / Temps d'intégration<br>Cette entrée détermine la réponse temporelle de l'intégrateur.   |
| TD         | TIME            | <b>DERIVATIVE TIME</b> / Temps de dérivation<br>Cette entrée détermine la réponse temporelle de l'unité de dérivation.  |
| PV_FAC     | REAL            | <b>PROCESS VARIABLE FACTOR</b> / Facteur de mesure<br>Cette entrée est multipliée par la mesure. Elle sert à adapter l'étendue de valeur de mesure  |
| PV_OFF     | REAL            | <b>PROCESS VARIABLE OFFSET</b> / Décalage de mesure<br>Cette entrée est ajoutée à la mesure. Elle sert à adapter l'étendue de valeur de mesure  |

*Tableau III.3 : Paramètres d'entrée*

Le tableau **III.4** montre les paramètres de sortie ainsi que leurs types de données :

| <b>Paramètre</b> | <b>Type de données</b> | <b>Description</b>  |
|------------------|------------------------|---|
| LMN_PER          | WORD                   | <b>MANIPULATED VALUE PERIPHERY</b> / Valeur de réglage de périphérie<br>Cette sortie fournit la valeur de réglage en format de périphérie.  |
| PV               | REAL                   | <b>PROCESS VARIABLE</b> / Mesure<br>Cette sortie donne la mesure effective.   |
| ER               | REAL                   | <b>ERROR SIGNAL</b> / Signal d'erreur<br>Cette sortie donne le signal d'erreur effectif.  |
| LMN_P            | REAL                   | <b>PROPORTIONALITY COMPONENT</b> / Composante P<br>Cette sortie contient la composante proportionnelle de la grandeur de réglage.   |
| LMN_I            | REAL                   | <b>INTEGRAL COMPONENT</b> / Composante I<br>Cette sortie contient la composante intégrale de la grandeur de réglage.  |
| LMN_D            | REAL                   | <b>DERIVATIVE COMPONENT</b> / Composante D<br>Cette sortie contient la composante différentielle de la grandeur de réglage.   |
| QLMN_HLM         | BOOL                   | <b>HIGH LIMIT OF MANIPULATED VALUE REACHED</b> /<br>Grandeur de réglage à la limite supérieure<br>La valeur de réglage est toujours bornée à une limite supérieure et à une limite inférieure. Cette sortie signale le dépassement de la limite supérieure. |
| QLMN_LLM         | BOOL                   | <b>LOW LIMIT OF MANIPULATED VALUE REACHED</b> /<br>Grandeur de réglage à la limite inférieure<br>La valeur de réglage est toujours bornée à une limite supérieure et à une limite inférieure. Cette sortie signale le dépassement de la limite inférieure.  |

**Tableau III.4 Paramètres de sortie**

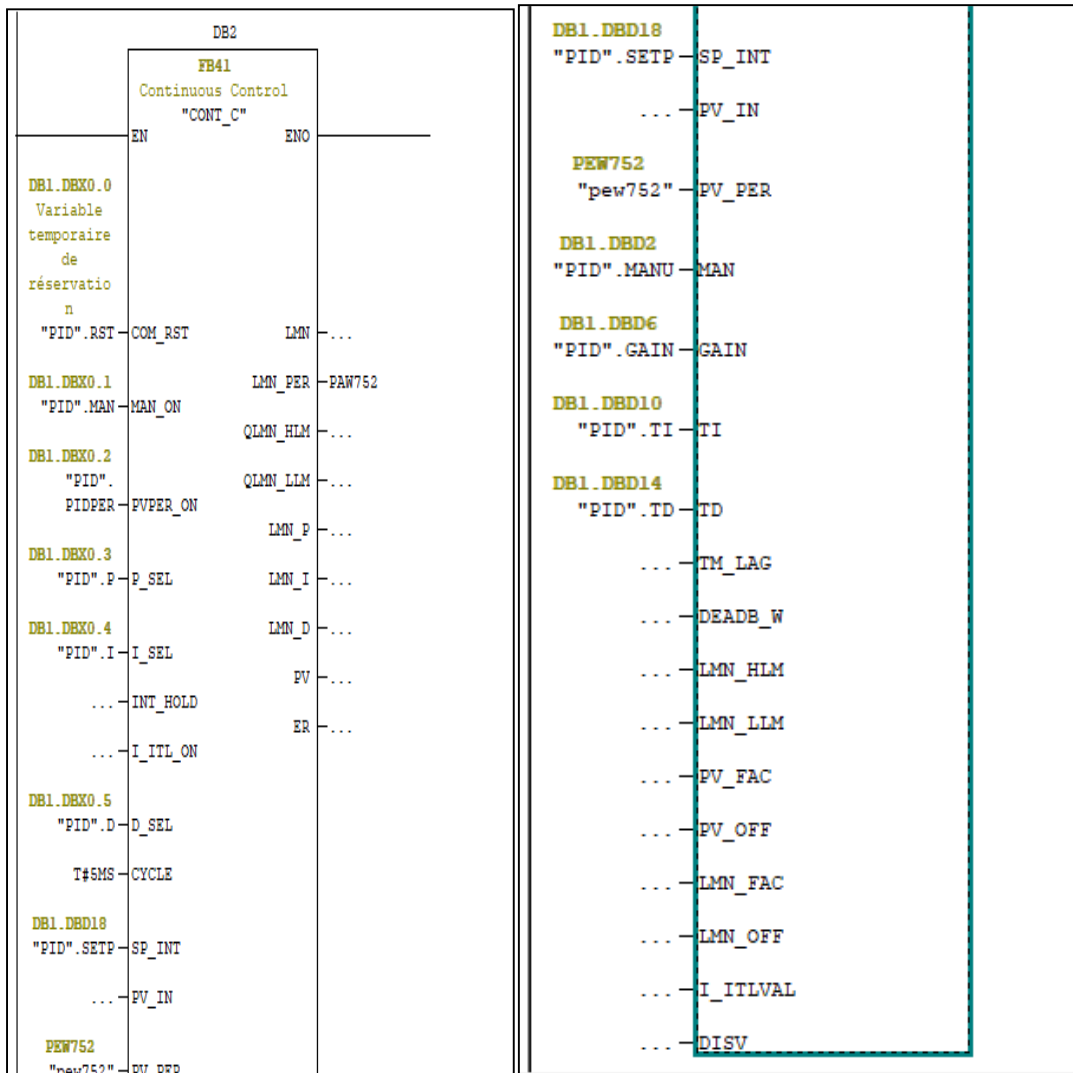
### **III.6.4- MISE EN OEUVRE D'UN REGULATEUR PID SOUS STEP7**

La programmation d'un automate industriel de type SIMATIC S7-300 en tant que régulateur PID se fait avec le logiciel STEP 7, les entrées sorties choisies de régulateur PID (bloc FB41) seront sauvegardées dans le bloc de données (DB1) d'instance associé à l'appel du bloc régulateur FB 41.

Pour concevoir un régulateur PID nous devons suivre les deux étapes essentielles :

- Configuration matériels ;
- Elaboration d'un programme en langage Contact sous STEP7 avec affectation des blocs fonctionnelle et données.

**III.6.5 - PROGRAMMATION DE REGULATION PID SOUS STEP7**



**Figure III.24: Paramétrage du régulateur sous step7 (FB41)**

On va utiliser un bloc de données pour les paramètres de cette régulation : Et un table mnémonique pour les autres blocs :

| Adresse | Nom    | Type       | Valeur initiale | Commentaire                        |
|---------|--------|------------|-----------------|------------------------------------|
| 0.0     |        | STRUCT     |                 |                                    |
| +0.0    | RST    | BOOL       | FALSE           | Variable temporaire de réservation |
| +0.1    | MAN    | BOOL       | FALSE           |                                    |
| +0.2    | PIDPER | BOOL       | FALSE           |                                    |
| +0.3    | P      | BOOL       | FALSE           |                                    |
| +0.4    | I      | BOOL       | FALSE           |                                    |
| +0.5    | D      | BOOL       | FALSE           |                                    |
| +2.0    | MANU   | REAL       | 0.000000e+000   |                                    |
| +6.0    | GAIN   | REAL       | 0.000000e+000   |                                    |
| +10.0   | TI     | TIME       | T#0MS           |                                    |
| +14.0   | TD     | TIME       | T#0MS           |                                    |
| +18.0   | SETP   | REAL       | 0.000000e+000   |                                    |
| =22.0   |        | END_STRUCT |                 |                                    |

**Figure III.25: Bloc de données de (FB41)**

|    | Etat | Mnémonique | Opérande | Type de do | Commentaire        |
|----|------|------------|----------|------------|--------------------|
| 1  |      | ALARME     | A 124.2  | BOOL       |                    |
| 2  |      | ARRET      | M 0.1    | BOOL       |                    |
| 3  |      | CONT_C     | FB 41    | FB 41      | Continuous Control |
| 4  |      | CYC_INT5   | OB 35    | OB 35      | Cyclic Interrupt 5 |
| 5  |      | DEMARER    | M 0.2    | BOOL       |                    |
| 6  |      | DISC       | M 0.3    | BOOL       |                    |
| 7  |      | EVACUATION | M 0.4    | BOOL       |                    |
| 8  |      | HV         | A 124.1  | BOOL       |                    |
| 9  |      | NIV        | MD 10    | REAL       |                    |
| 10 |      | PAW752     | PAW 754  | WORD       |                    |
| 11 |      | pew752     | PEW 752  | WORD       |                    |
| 12 |      | PID        | DB 1     | DB 1       |                    |
| 13 |      | POMPE      | A 124.0  | BOOL       |                    |
| 14 |      | SCALE      | FC 105   | FC 105     | Scaling Values     |
| 15 |      | UNSCALE    | FC 106   | FC 106     | Unscaling Values   |
| 16 |      |            |          |            |                    |

**Figure III.26: Table mnémonique de programme**

On a choisi ce CPU 313C par ce qu'il intègre des modules analogique compact pour travailler sur des périphérique analogique ( Vanne et Capteur ) et aussi a grâce à leur performance en industrie.

### III.7- CONFIGURATION DES MODULES ANALOGIQUE

#### III.7.1 MODULE D'ENTREE/SORTIES ANALOGIQUE

Propriétés - AI5/AO2 - (R0/S2.3)

Général | Adresses | Entrées | Sorties

Unité de température :

| Entrée         | 0                                    | 1                                    | 2                                     | 3                                     | 4                                     |
|----------------|--------------------------------------|--------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| Mesure         |                                      |                                      |                                       |                                       |                                       |
| Type :         | <input type="text" value="U"/>       | <input type="text" value="U"/>       | <input type="text" value="U"/>        | <input type="text" value="U"/>        | <input type="text" value="R 2F"/>     |
| Plage :        | <input type="text" value="0..10 V"/> | <input type="text" value="0..10 V"/> | <input type="text" value="+/- 10 V"/> | <input type="text" value="+/- 10 V"/> | <input type="text" value="600 ohms"/> |
| Fréq. perturb. | <input type="text" value="50 Hz"/>   | <input type="text" value="50 Hz"/>   | <input type="text" value="50 Hz"/>    | <input type="text" value="50 Hz"/>    |                                       |

OK Annuler Aide

|  |           |           |  |
|--|-----------|-----------|--|
|  | 124...126 | 124...125 |  |
|  | 752...761 | 752...755 |  |
|  | 768...783 | 768...783 |  |

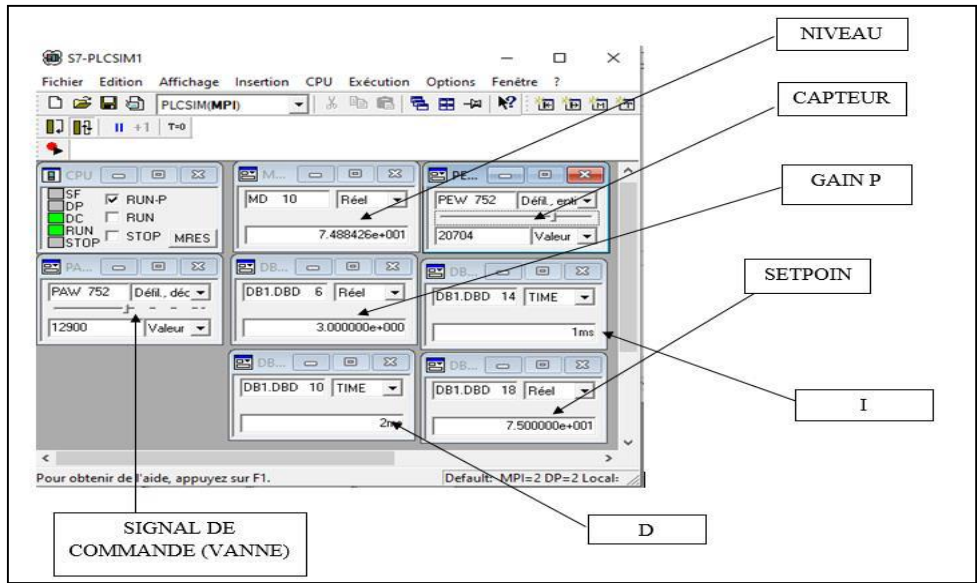
**Figure III.27: Configuration des modules analogique**

**❖ Exemple de simulation :**

On donne une consigne de 75% (SETPOINT) directement la cuve va le indiquer le liquide et le capteur va faire le calcul pour donner le niveau de ce liquide (Mesure) et le comparée avec la consigne et donne un signal à actionneur (Vanne).

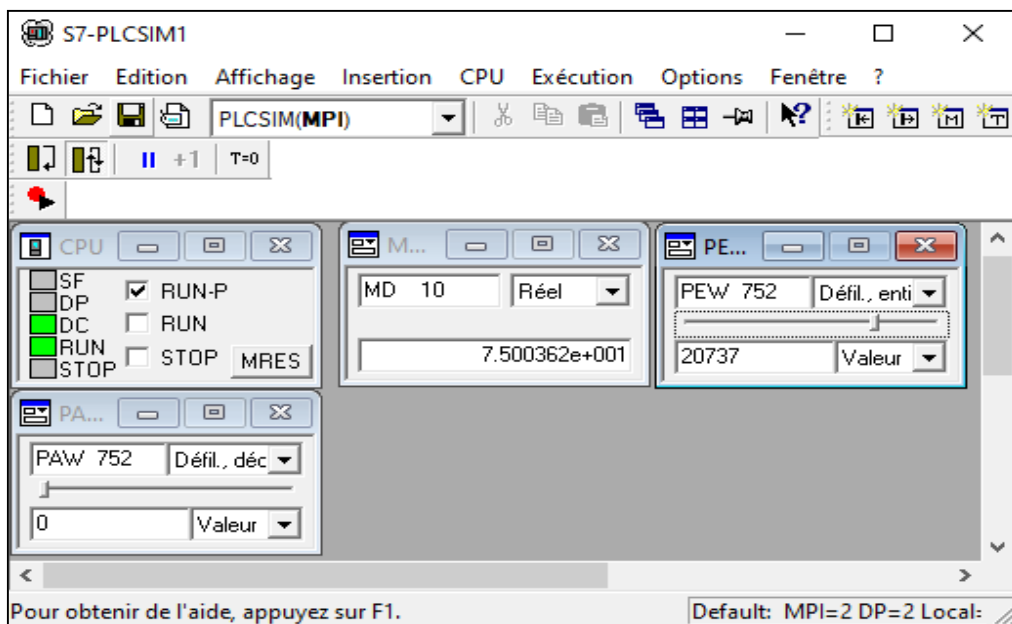
On utilise PLCSIM dans l'expérience pour voir le signal de commande :

**➤ Régulation PID :**



**Figure III.28: Régulation PID**

**➤ Régulation Discontinue :**



**Figure III.29: Régulation Discontinue**



Nous avons constaté que durant l'utilisation de la régulation DISCONTINUE la vanne ne peut pas prendre des valeur pour que le système rend stable par ce que le niveau ne va jamais stabiliser et la vanne va prend une valeur de tout ou rien et ça rend notre procès non confortable sur tout lorsque le système atteint la consigne après chute avec une petite valeur la vanne va ouvre tout et le niveau dépassé la consigne.

Au contraire avec la régulation PID nous avons observé que lorsque le système approche a la consigne (SETPOINT) la vanne va recevoir un signal de commande pour la fermeture et ne dépassée pas la seuil ou bien la limite et il va commander le système bien et rend le système plus stable que la régulation discontinue.

Autrement dans la régulation DISCONTINUE on donne la valeur exact pour l'actionneur pour le faire fonctionner mais. En PID on va juste donner SETPOINT et les autres paramètres (P I D) et le système va fonctionner automatiquement sans configuration de la vanne.

### III.8- CREATION D'UN VUE DANS WINCC FLEXIBLE

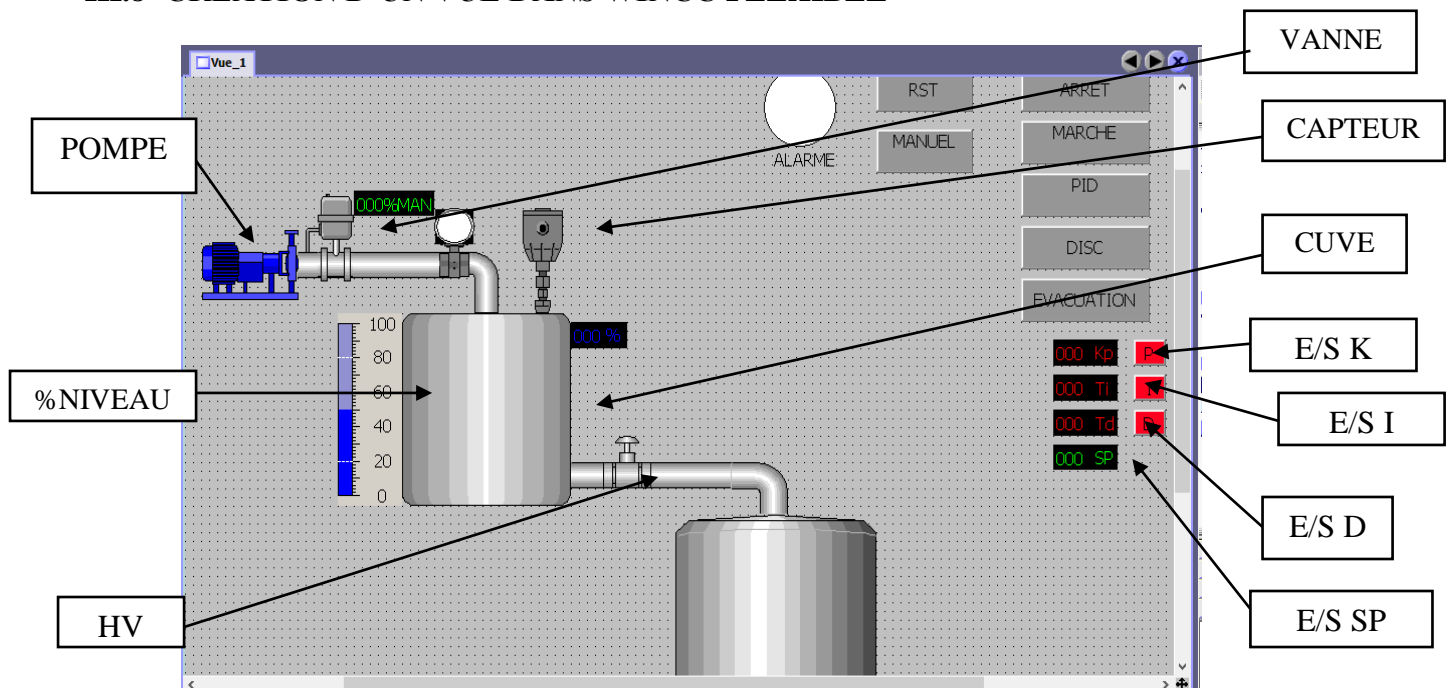


Figure III.30 : Vue d'accueil du système de contrôle de processus PUL-2

#### III.8.1- WINCC FLEXIBLE RUNTIME

L'application runtime de WinCC flexible, permet à l'opérateur de réaliser le contrôle-commande du processus. Les tâches suivantes sont alors exécutées:

- Communication avec les automates du processus.
- Affichage des vues à l'écran de supervision ;
- Commande du processus, par exemple. Spécification de consignes ou ouverture et fermeture de vannes, démarrage et arrêt des moteurs et pompes, etc ;
- Archivage des données de runtime actuelles, des valeurs de processus et événements d'alarme pour un diagnostic de défaillances.

### III.8.2- INTERFACE DE COMMANDE PID ET DISCONTINUE DE LA STATION PUL-2/EV

La figure (III.31) représente la vue de commande et de supervision que nous avons développée sous environnement WINCC flexible pour notre système de régulation de niveau PUL-2/EV.

La plateforme de commande et de supervision développée permet, la sélection des deux stratégies de commande (PID et DISCONTINUE), ainsi que la récupération des différentes informations nécessaires à l'exploitation de cette application.

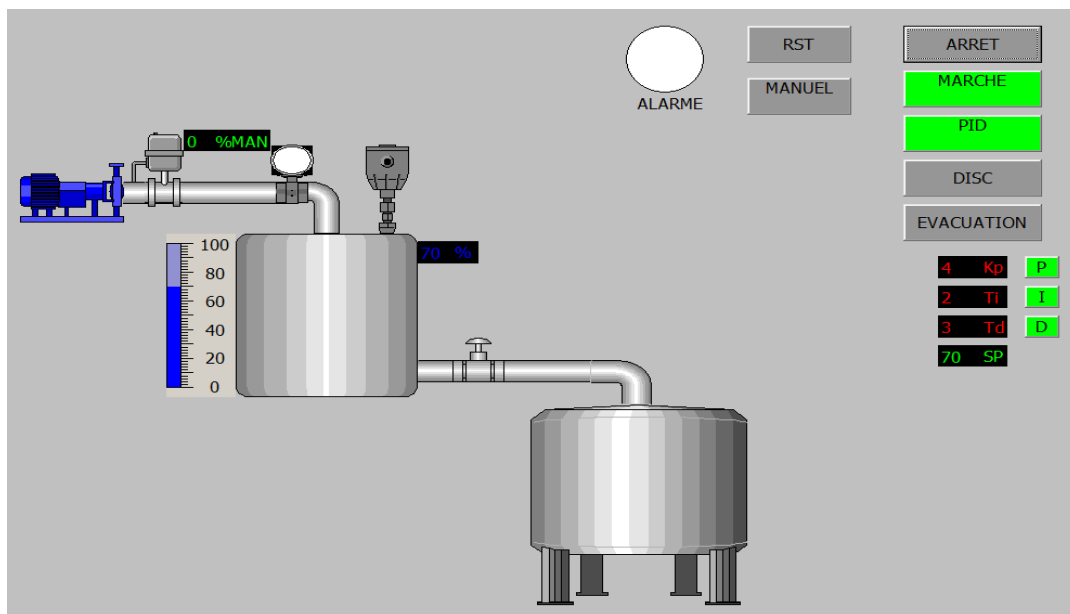


Figure III.31 : Runtime du système de contrôle de processus PUL-2

### III.9- CONCLUSION

Ce chapitre était consacré à la présentation des différents techniques d'implémentation de régulateurs (PID et DISCONTINUE), et aux résultats expérimentaux obtenus de l'application de ces techniques sur l'automate S7 300, nous avons présenté une interprétation des concepts de PID en langage programmable (LADDER) en utilisant les outils de programmation fourni dans la bibliothèque du logiciel de programmation des APIs (S7-300).

D'après les résultats expérimentaux, on peut déduire que la Commande PID donne des résultats meilleurs, surtout lorsqu'on effectue des SETPOINT sur le système. Ce qui n'est pas le cas pour la commande DISCONTINUE qui nous donne des résultats ne change pas et n'achève pas notre besoin (niveau, débit...). Le régulateur PID que nous avons implémenté sous S7-300 étant plus recommandé pour notre application de régulation de niveau vu sa stabilité et ses performances dynamiques.

**CONCLUSION GENERALE**

## CONCLUSION GENERALE

La commande PID est une méthode classique de commande des procédés industriels. Elle utilise des techniques s'appuyant sur le savoir-faire humain, exprimées en mots et en phrases ordinaires, plutôt que sur des équations.

Le but de notre projet était d'élaborer une technique d'implémentation et d'application de la régulation PID et DISCONTINUE dans un contexte d'automate programmable S7-300. Afin d'atteindre cet objectif, nous avons choisi le système de régulation de niveau PUL-2 EV.

Au début nous avons déterminé expérimentalement les deux domaines (Régulation et Automatisation). Ensuite, nous avons présentées des commandes avancées qui n'ont pas été implémentées sur les API jusqu'à maintenant et faire des investigations concernant ce sujet. Après l'investigation, nous avons constaté que des langages de programmation avancés comme C++ et les appeler dans l'API pour réaliser ces commandes. Ensuite, la régulation DISCONTINUE que nous avons choisie ne nous offre pas des performances dynamiques suffisantes.

Pour pallier à ce problème, nous nous sommes intéressés à la commande PID, afin de profiter de ses avantages et d'étudier ses possibilités d'implémentation pratique.

L'implémentation du régulateur PID sur un API S7 300, avec CPU 313C, nous a permis de développer une procédure de programmation structurée. L'application théorique de ce programme développé sur la station de niveau PUL2/EV, nous a permis d'améliorer les performances dynamiques de la régulation, notamment les performances par rapport au mode. Il est à noter que la procédure de programmation que nous avons choisie lors de l'implémentation du correcteur PID, peut être adaptée à tous types d'automates se programmant en langage Ladder.

Ce travail nous a permis d'acquérir une expérience enrichissante dans le domaine de la régulation et de la programmation des APIs, ainsi que la manipulation des logiciels (Step7, WinCC) de SIEMENS. De plus, ce travail nous a permis également de mettre en pratique les connaissances acquises durant notre cursus.

Enfin, nous espérons que nos efforts puissent servir comme support technique et que ce mémoire soit un bon guide pour les promotions futures, et que notre travail contribuera pour enrichir la qualité des travaux pratiques au sein du laboratoire d'automates programmables et de la régulation industrielle.

**REFERENCES BIBLIOGRAPHIQUES**

## REFERENCES BIBLIOGRAPHIQUES

- [1] **AIT AMRANE Adlane. ZENATI Noureddinen**, Titre : Réalisation d'un banc d'essai de la chaîne d'action « variateur de vitesse – moteur asynchrone » commandé par un API et supervisé par un écran tactile, Université Djilali Bounâama - Khemis Miliana, 2015/2016.
- [2] <https://www.technologuepro.com/>, Cours de : Capteurs et actionneurs en instrumentation.
- [3] <http://www.electrosup.com/servomoteur.php> , rapport servomoteur.
- [4] **H. AYAD**, Cours PLC, Master 2, Automatique et Systèmes, Département d'électronique, USDB1, Algérie, 2015.
- [5] **G. Michel**, Les API : Architecture et Application des Automates Programmables Industriels, édition DUNOD, Paris, 1987.
- [6] <https://w3.siemens.com/mcms/programmable-logic-controller/en/basic-Controller/s7-1200/CPU/pages/default.aspx>. BROCHURE: Siemens.com/Simatic-s7-1200.
- [7] [http://stimonge.fr/maintenancesystemes/wpcontent/uploads/2013/02/s71200\\_System\\_Manual\\_fr-FR\\_fr-FR.pdf](http://stimonge.fr/maintenancesystemes/wpcontent/uploads/2013/02/s71200_System_Manual_fr-FR_fr-FR.pdf) , Automate programmable S7-1200, Manuel système, 04/2012.
- [8] **IBAZIZENE Amar ELHADJEN Hamza**, Thème Présenté par IBAZIZENE Amar ELHADJEN Hamza Amélioration de la supervision de la station de traitement de sable de l'entreprise MFG, Université Mouloud MAMMERI, Tizi-Ouzou 2012.
- [9] PL7 Micro/Junior/Pro Présentation rapide de PL7, <https://readthedocs.web.cern.ch/download/attachments/21178144/Tquick.pdf?version=1&modificationDate=1246433731000&api=v2> .
- [10] cours Système d'automatisation S7-300 Installation et configuration : CPU 312 IFM -318-2 DP. <https://cours.etsmtl.ca/gpa141/AutresDocuments/siemens/Fibel.pdf> .
- [11] Logiciel TIA PORTAL, Logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système Volume 1/2 Manuel de référence.
- [12] TIAPORTAL, <https://www.industry.siemens.com/topics/global/fr/tia-portal/hmi-sw-tiaportal/wincc-tia-portal-es/pages/default.aspx>
- [13] <https://www.univ-reims.fr/> site web - Descriptif des Logiciels (univ-reims.fr).
- [14] **N. Tayssir SGHAIER**, Automatiser un filtre de poussière, Mémoire de licence appliquée en Informatique Industrielle Systèmes Embarqués, Institut supérieur informatique de Médenine (Tunisie), THELB, 2011.
- 81
- [15] **BEDOUD KHOULOUD**, 2010, « Modélisation et diagnostic de défaillance d'une électrovanne pneumatique », Magister, Université Badji Mokhtar-Annaba.
- [16] <http://cira83.com> , Régulation Industrielle BTS CIRA 1 - Lycée Rouvière.
- [17] **Mrs, BENARBIA Taha** cours Régulation de "Niveau-Débit Pression". Chapitre V.
- [18] **Ghania Boukerche**, « Etude et Synthèse d'un Contrôleur PI et Application », Université Badji Mokhtar-Annaba. 2017.
- [19] **SAHRAOUI Mebarek** THEME Commandes non linéaires de la machine asynchrone UNIVERSITE DE M'SILA 2012/2013.
- [20] **M.DJAGHDALI** Cour de La commande prédictive directe du couple de la machine à induction Chapitre IV, UNIVERSITE de Biskra.
- [21] **Ahmed CHEMORI** Master 2 : ScTIC Cours: Systèmes robotiques pour l'assistance à la mobilité et à la rééducation. UNIVERSITE Montpellier 2.
- [22] **LEGRIOUI SAID**. Université de Constantine 1, Thème Etude et réalisation de la commande DTC des machines asynchrones par la logique floue,
- [23] **Nadia HAMLAOUI**, Thème Commande Optimale d'un Bras Manipulateur, Option : Robotique, UNIVERSITE DE BATNA FACULTE DE TECHNOLOGIE.
- [24] <https://elearning-facsci.univ-annaba.dz> , Cours la commande optimale.
- [25] Principes de régulation, <https://www.support-omron.fr/>.

- [26] **DR.M. Rabi**, « régulation analogique industriel », ESTT. G. thermique.  
<http://www.est-usmba.ac.ma/Rabi/C-Analogique/diapos.pdf/Chapitre4.pdf> .
- [27] <https://www.abcclim.net/regulation-p-pi-pid.html>, Tout savoir sur la régulation PID (abcclim.net). Guide technique
- [28] <https://www.abcclim.net> Régulation numérique et analogique en climatisation (abcclim.net). Guide technique.
- [29] **C. ROSENTAL** ; «Histoire de la logique floue. Une approche sociologique des pratiques de démonstration», octobre-décembre 1998.
- [30] **Hansruedi Buhler**, Cours « Réglage par logique floue », CH-1015 Lausanne, 1994. Université de lorraine.

82

[31] **SEDDAR Yahia GHEZAL Katia**, Thème Contribution à l'Implémentation d'un Régulateur Flou sous un API S7-300, Mémoire de Fin d'Etude, Automatique et Informatique Industrielles, Université Mouloud Mammeri De Tizi-Ouzou.

[32] Manuel de référence SIMATIC, SIEMENS AG 2010.

[33] **WinCC flexible** 2008/Compact / Standard / Advanced ; Manuel d'utilisation référencé .....6AV6691-1AB01-3AC0; 2008.

[34] **madame Nassima Aissani**, Cours l'automatisme M1 – Maintenance en Automatisme et instrumentation Industriel. IMSI 2021.

[35]SIMATIC Logiciel de base pour S7-300/400 Régulation PID, manuel Siemens AG, 1996.

**Logiciel utilisée :**

-SIMATIC MANAGER STEP7

-WINCC FLEXIBLE 2008

-PLC SIM