

**République Algérienne Démocratique et Populaire**  
**Ministère de l'enseignement Supérieur Et de la**  
**Recherche Scientifique**



**Université M'hamed Bougara Boumerdes**

**Faculté des Sciences de l'Ingénieur**



**Département : Ingénierie des Systèmes Electriques**

**Option : Ingénierie des Systèmes Electriques**

**Spécialité : Electronique des Systèmes Embarquées**

# **Mémoire de fin d'étude**

**En vue d'obtention  
du diplôme de master**

## **Thème**

**Etude et réalisation d'un dégazeur par  
l'application d'une boucle de régulation Split-  
range à base d'un API SIEMENS S7-300**

Réalisé par :

**Mr. BOUARRA Ahmed**

Membre des jurys :

**Mr. NAFA Farès Président**

**Mme HAROUN Radia Membre**

**Mr. RAHMOUNE Fayçal Encadreur**

**Mr. MAALLEMI Mohamed Co Encadreur**

**2017-2018**

# Remerciements

*Nous tenons à remercier notre promoteur  
Dr RAHMOUNE qui a bien voulu nous encadrer  
pour l'élaboration et le suivi de notre projet de  
fin d'études.*

*Nous tenons à remercier également les membres du  
jury qui nous ont fait l'honneur d'examiner ce  
travail et d'apporter leurs réflexions et suggestions  
scientifiques.*

*Nous tenons à exprimer notre profonde gratitude à  
notre encadreur Mr MAALLEMI Mohamed pour son  
aide et son encouragement qui n'a épargné aucun  
effort pour que notre stage se déroule dans de  
meilleures conditions ainsi que  
Mme GHARSA Nassima chef département TMI,  
OUAZENE Hamza et tous les responsables  
et le personnel de l'IAP.*

*Nous profitons de l'occasion pour remercier tous nos  
enseignants de INGM/UMBB.*

*Enfin, que toutes celles et tous ceux qui, de près ou de  
loin nous ont généreusement offert leurs concours à  
l'élaboration de ce travail, trouvent ici l'expression  
de notre profonde sympathie.*

# Dédicaces

*A la femme la plus courageuse, sensible, généreuse,  
la plus belle à mes  
yeux, à celle qui a su me donner amour et joie de vivre,  
à celle qui a toujours  
montrée affection et compréhension à mon égard,  
le symbole de la tendresse,  
qui m'a soutenu en toutes circonstances,  
ma  
mère que j'aime.*

*A l'homme de courage et de force à celui  
qui a toujours été présent dans mon cœur,  
qui m'a appris les vraies valeurs de la vie à celui  
qui m'a soutenu en toutes circonstances,  
dieu ait son âme  
mon  
père que j'aime.*

*A ceux qui m'ont donné joie et bonheur,  
mes sœurs Sara et Intissar*

*A mes frères Zakaria, Mouhamed, Ilyes, Abd arrahmen  
et Omar.*

*A mon très chère cousin Mouhamed.*

*A mes amis les plus fidèles ZENTOUT Adel et BOULEMZAOUED Med.*

*A tous ceux que j'aime, à tous ceux qui m'aiment, je dédie ce modeste  
travail.*

**Remerciements**

**Dédicaces**

**Introduction générale**

**Chapitre 1 : Réalisation d'un dégazeur a l'IAP de Boumerdes**

Introduction ..... 1

1.1 Missions de l'IAP ..... 1

1.2 Organisation de l'IAP ..... 2

    1.2.1 Structures opérationnelles de l'IAP ..... 2

    1.2.2 Spécialités de l'Ecole de Boumerdès..... 3

1.3 L'école de Boumerdès ..... 3

    1.3.1 Ses domaines d'activité..... 3

    1.3.2 Les laboratoires..... 3

    1.3.3 UFR Transport, Maintenance & Instrumentation..... 4

        1.3.3.1 Personnel..... 4

        1.3.3.2 Laboratoires ..... 4

1.4 Description de CIS ..... 5

    1.4.1 Unité de traitements eaux huileuses CIS ..... 5

        1.4.1.1 Le séparateur triphasique ..... 6

        1.4.1.2 Le dégazeur..... 7

1.5 l'étude de notre boucle de régulation split range ..... 8

    1.5.1 la régulation split range ..... 8

    1.5.2 la surveillance des alarmes ..... 9

1.6 La supervision ..... 9

Conclusion..... 10

**Chapitre 2 : Régulation industrielle et boucle de régulation split range**

Introduction ..... 11

2.1 Régulation Industrielle ..... 11

2.2 Principe de régulation ..... 12

    2.2.1 La chaîne de traitement de l'information ..... 12

    2.2.2 Les éléments de la chaîne ..... 12

        2.2.2.1 Le capteur..... 12

        2.2.2.2 L'acquisition de la mesure ..... 13

        2.2.2.3 Le traitement de la mesure ..... 13

        2.2.2.4 Le traitement de l'algorithme de régulation..... 14

2.2.2.5 La commande .....	14
2.2.2.6 L'actionneur .....	15
2.2.2.7 Le procédé.....	15
2.3 Les types de procédés .....	16
2.3.1 Continu .....	16
2.3.2 Discontinu.....	16
2.3.3 Stable .....	17
2.3.4 Instable .....	17
2.4 Type de système dynamique .....	17
2.4.1 Mono variable.....	17
2.4.2 Multi variable .....	18
2.5 La régulation PID .....	18
2.5.1 Les actions .....	18
2.5.1.1 Action Proportionnelle.....	18
2.5.1.2 Action Intégrale .....	19
2.5.1.3 Action Dérivée .....	20
2.5.2 Les différents types d'implémentations.....	20
2.5.2.1 Structure PID Série .....	21
2.5.2.2 Structure PID Parallèle .....	21
2.5.2.3 Structure PID Mixte.....	22
2.5.3 L'influence des coefficients .....	22
2.6 Les différents types de calculs .....	22
2.6.1 L'algorithme Absolu.....	22
2.6.2 L'algorithme Incrémental .....	23
2.7 Les boucles de régulations .....	23
2.7.1 Régulation en boucle ouverte .....	23
2.7.2 régulation en boucle fermée .....	24
2.7.3 régulation cascade .....	25
2.7.4 régulation de rapport .....	25
2.7.5 régulation split range .....	26
2.7.5.1 Séquençage de vanne complémentaire .....	26
2.7.5.2 Séquençage de vanne exclusive .....	28
2.7.5.3 Séquençage de vanne progressive.....	30
2.7.5.4 Application split-range sur notre dégazeur .....	31
Conclusion.....	36

**Chapitre 3 : Automate Programmable industriel**

Introduction ..... 37

3.1 Structure des systèmes automatisés ..... 37

3.2 Les systèmes de commande ..... 38

    3.2.1 Les solutions câblées ..... 38

    3.2.2 Les solutions programmées (utilisation d'un automate programmable)..... 39

3.3 L'automate programmable industriel ..... 39

    3.3.1 Définition..... 39

    3.3.2 Architecture des automates programmables industriels ..... 40

        3.3.2.1 Le processeur ..... 40

        3.3.2.2 La mémoire ..... 40

        3.3.2.3 Interface Entrées-Sorties ..... 41

        3.3.2.4 Les bus ..... 41

        3.3.2.5 Le module d'alimentation ..... 42

    3.3.3 Principe de fonctionnement d'un API ..... 42

    3.3.4 Les langages de programmation des API ..... 43

        3.3.4.1 Langages graphiques..... 43

        3.3.4.2 Langages littéraux ..... 45

    3.3.5 Comparaison des langages ..... 46

3.4 Critères de choix d'un automate ..... 46

3.5 L'automate Siemens S7-300 ..... 47

    3.5.1 Caractéristiques de l'API S7-300 ..... 48

    3.5.2 Constitution ..... 48

        a) Module d'alimentation (PS)..... 48

        b) Unité centrale (CPU) ..... 49

        c) Coupleurs (IM) : ..... 51

        d) Modules de signaux (SM) : ..... 51

3.6 Logiciel de programmation SIMATIC Manager STEP7 V5.5 ..... 51

    3.6.1 Présentation du logiciel ..... 51

    3.6.2 Démarrer le SIMATIC STEP7 V5.5 ..... 51

    3.6.3 La conception d'un programme avec SIMATIC Manager STEP7 V5.5 ..... 52

        3.6.3.1 Création d'un nouveau projet..... 53

        3.6.3.2 Configuration matériels..... 54

        3.6.3.3 Compilation et chargement de la configuration ..... 57

        3.6.3.4 La création de la table des mnémoniques ..... 58

## Sommaire

---

3.6.3.5 Ecriture du programme .....	59
3.6.4 Structure d'un programme .....	64
3.6.5 Simulation avec S7-PLCSIM.....	65
3.6.6 Test du programme .....	66
3.6.6.1 Calcul des paramètres du bloc FB41 « le régulateur PID » .....	66
3.6.6.2 Le splitter .....	69
3.7 Le logiciel de supervision WinCC flexible 2008 .....	71
3.7.1 Avantage de la supervision.....	71
3.7.2 Composants de WinCC .....	71
3.7.3 Configuration.....	72
3.7.4 Elément de WinCC flexible 2008.....	72
3.7.5 Résultat obtenu .....	73
Conclusion.....	75

**Conclusion générale**

**Bibliographie**

**Annexe**

**Glossaire**

## Liste des figures

<b>Figure 1.1</b> : Structures opérationnelles de l'IAP .....	2
<b>Figure 1.2</b> : Mise en œuvre de l'IAP .....	2
<b>Figure 1.3</b> : Mise en œuvre de l'école de Boumerdes .....	4
<b>Figure 1.4</b> : Les laboratoires de l'UFR TMI .....	4
<b>Figure 1.5</b> : Unité de traitement des eaux huileuses SH-DP-CIS .....	6
<b>Figure 1.6</b> : le séparateur triphasique .....	7
<b>Figure 1.7</b> : Le dégazeur .....	7
<b>Figure 1.8</b> : Notre dégazeur.....	8
<b>Figure 1.9</b> : Régulation de pression en split-range.....	9
<b>Figure 1.10</b> : La surveillance des alarmes .....	9
<b>Figure 2.1</b> : les éléments de la chaîne de régulation.....	12
<b>Figure 2.2</b> : Capteur et transmetteur.....	13
<b>Figure 2.3</b> : acquisition de la mesure.....	13
<b>Figure 2.4</b> : Traitement de la mesure avant l'utilisation .....	14
<b>Figure 2.5</b> : Algorithme qui va définir les actions à effectuer .....	14
<b>Figure 2.6</b> : L'écriture vers l'entrée et l'écriture vers l'extérieur.....	15
<b>Figure 2.7</b> : Les actionneurs industriels .....	15
<b>Figure 2.8</b> : Un exemple de procédé continu (Dépropaniseur).....	16
<b>Figure 2.9</b> : Un exemple de procédé discontinu (Autoclave) .....	16
<b>Figure 2.10</b> : Régulateur en mode manuel d'un process stable.....	17
<b>Figure 2.11</b> : Régulateur en mode manuel d'un process instable .....	17
<b>Figure 2.12</b> : Système mono variable.....	18
<b>Figure 2.13</b> : Système multi variable .....	18
<b>Figure 2.14</b> : Impulsion sur la consigne .....	19
<b>Figure 2.15</b> : Action Proportionnelle .....	19
<b>Figure 2.16</b> : Action Intégrale .....	20
<b>Figure 2.17</b> : Structure PID Série .....	21
<b>Figure 2.18</b> : Structure PID Parallèle .....	21
<b>Figure 2.19</b> : Structure PID Mixte.....	22
<b>Figure 2.20</b> : Régulation de la température d'un four en boucle ouverte .....	24
<b>Figure 2.21</b> : Régulation de la température d'un four en boucle fermée .....	24
<b>Figure 2.22</b> : processus complexe en deux sous-systèmes.....	25
<b>Figure 2.23</b> : Régulation cascade .....	25
<b>Figure 2.24</b> : Régulation de rapport .....	26
<b>Figure 2.25</b> : Mélange de deux liquides pour former une peinture colorée .....	27
<b>Figure 2.26</b> : Représentation de chaque ouverture de vanne.....	28
<b>Figure 2.27</b> : Traitement des eaux par l'addition d'acides ou de caustiques.....	28
<b>Figure 2.28</b> : Représentation de chaque ouverture de vanne.....	29
<b>Figure 2.29</b> : Traitement des eaux pH élevé est ramenée au neutre par l'addition d'acides.....	30
<b>Figure 2.30</b> : Représentation de chaque ouverture de vanne.....	30
<b>Figure 2.31</b> : Séparateur d'eau huileuse et dégazeur par deux vannes régulatrices .....	31
<b>Figure 2.32</b> : Le signal pneumatique commun.....	32
<b>Figure 2.33</b> : Un courant de signal commun .....	33
<b>Figure 2.34</b> : plusieurs sorties sont configurées sur un contrôleur.....	34
<b>Figure 2.35</b> : le séquençage à split range par deux contrôleurs .....	34

## Liste des figures

<b>Figure 2.36</b> : Représentation l'ouverture des vannes et la bande morte .....	35
<b>Figure 2.37</b> : Le splitter reçoit un signal de commande unique et émet deux signaux .....	36
<b>Figure 3.1</b> : Structure d'un système automatisé .....	37
<b>Figure 3.2</b> : Structure interne d'un API .....	40
<b>Figure 3.3</b> : Fonctionnement d'un API .....	43
<b>Figure 3.4</b> : Exemple d'un langage à contacts .....	43
<b>Figure 3.5</b> : Exemple d'un langage FBD .....	44
<b>Figure 3.6</b> : Exemple d'un langage SFC .....	44
<b>Figure 3.7</b> : Exemple d'un langage Instruction List.....	45
<b>Figure 3.8</b> : Exemple d'un langage ST.....	45
<b>Figure 3.9</b> : Présentation de la gamme SIMATIC .....	47
<b>Figure 3.10</b> : Modules de l'API S7-300 .....	48
<b>Figure 3.11</b> : Présentation de la CPU 313C .....	50
<b>Figure 3.12</b> : Lancement de SIMATIC Manager en utilisant le menu Démarrer .....	52
<b>Figure 3.13</b> : Lancement de SIMATIC Manager en utilisant le raccourci du bureau.....	52
<b>Figure 3.14</b> : Organisation pour la création de projets sous SIMATIC Manager STEP7.....	53
<b>Figure 3.15</b> : Création d'un nouveau projet.....	53
<b>Figure 3.16</b> : Insertion Station SIMATIC 300 .....	54
<b>Figure 3.17</b> : Icône « Station SIMATIC 300 ».....	54
<b>Figure 3.18</b> : Icône « matériel » .....	54
<b>Figure 3.19</b> : La liste des « stations ».....	55
<b>Figure 3.20</b> : Profilé support du RACK-300.....	55
<b>Figure 3.21</b> : Placement du module alimentation de 5A .....	56
<b>Figure 3.22</b> : Choix de la CPU .....	56
<b>Figure 3.23</b> : Les différents câbles utilisés.....	57
<b>Figure 3.24</b> : Le fichier « Mnémoniques ».....	58
<b>Figure 3.25</b> : La création de la table de mnémonique .....	58
<b>Figure 3.26</b> : Les blocs utiliser dans notre travail .....	59
<b>Figure 3.27</b> : Les fonctions FC105 et FC106 (SCALE et UNSCALE) .....	61
<b>Figure 3.28</b> : Création des blocs et des fonctions.....	62
<b>Figure 3.29</b> : L'organigramme présentant les alarmes de notre boucle split range .....	63
<b>Figure 3.30</b> : Le régulateur PID et le Splitter.....	63
<b>Figure 3.31</b> : Choisir le langage de programmation Ladder (CONT).....	64
<b>Figure 3.32</b> : Le degré d'imbrication maximum des blocs dépend la CPU .....	64
<b>Figure 3.33</b> : L'organigramme présentant la programmation structurée de notre projet .....	65
<b>Figure 3.34</b> : Simulation de la partie « Alarmes » de notre projet avec S7-PLCSIM.....	66
<b>Figure 3.35</b> : L'application « Paramétrage du régulateur PID ».....	67
<b>Figure 3.36</b> : Traceur de courbes.....	68
<b>Figure 3.37</b> : Présentation t et $\tau$ avec le traceur de courbes .....	68
<b>Figure 3.38</b> : Splitter « Partie d'actionnement la vanne de dégazage » .....	70
<b>Figure 3.39</b> : Splitter « Partie d'actionnement la vanne d'engazage ».....	71
<b>Figure 3.40</b> : notre boucle split range sur le WinCC.....	72
<b>Figure 3.41</b> : perturbation par la vanne de dégazage manuelle.....	73
<b>Figure 3.42</b> : perturbation par la vanne de dégazage manuelle.....	73
<b>Figure 3.43</b> : cas d'engazage en boucle split-range .....	74
<b>Figure 3.44</b> : cas de dégazage en boucle split-range.....	74

## Liste des tableaux

---

<b>Tableau 2.1</b> : Récapitulatif de l'action des coefficients .....	22
<b>Tableau 2.2</b> : la relation entre l'ouverture de chaque vanne et la sortie du contrôleur.....	27
<b>Tableau 2.3</b> : la relation entre l'ouverture de chaque vanne et la sortie du contrôleur.....	29
<b>Tableau 2.4</b> : la relation entre l'ouverture de chaque vanne et la sortie du contrôleur.....	31
<b>Tableau 2.5</b> : la relation entre l'ouverture des vannes, la sortie du contrôleur et la sortie des deux transducteurs I/P .....	35
<b>Tableau 3.1</b> : Comparaison des langages .....	46
<b>Tableau 3.2</b> : Caractéristique technique du module alimentation.....	56
<b>Tableau 3.3</b> : Caractéristique technique de la CPU 313C.....	57
<b>Tableau 3.4</b> : caractéristiques du bloc d'alarme cyclique OB35 .....	59
<b>Tableau 3.5</b> : Les paramètres du régulateur P, PI et PID .....	69
<b>Tableau 3.6</b> : Limite de conversion des FC106 et FC105.....	70

## Introduction générale

---

Le terme de régulation renvoie dans son sens concret à une discipline technique, qui se rattache sur le plan scientifique à l'automatique. La régulation est un cas spécifique de rétroaction où le système tend à réduire ses écarts à la commande.

La régulation dans le domaine des procédés industriels concerne la mise en œuvre de l'ensemble des moyens théoriques, matériels et techniques pour maintenir chaque grandeur physique essentielle égale à une valeur désirée, appelée consigne, par action sur une grandeur réglante, et ce, malgré l'influence des grandeurs perturbatrices du système.

En automatique, pour la régulation d'un système dynamique, une boucle de régulation est un dispositif constitué d'un ou de plusieurs capteurs mesurant une grandeur physique à contrôler, fournissant ainsi les données nécessaires au système de commande d'une machine ou d'un processus dont l'état est susceptible de modifier cette grandeur.

La boucle split-range est un type qui permet de décomposer la plage de réglage d'une variable en plusieurs segments, en général deux.

Ce genre de régulation se rencontre en climatisation chaud et froid : par exemple la commande de 0 % à 50 % agit sur une vanne NO affectée à de l'eau glycolée (Froid) qu'elle ferme progressivement et de 50 % à 100 % sur une vanne NF affectée à de la vapeur surchauffée (Chaud) qu'elle ouvre progressivement. Le régulateur balaye ainsi la plage de 0 % (refroidissement maximal) à 100 % (chauffage maximal) en passant par 50 % (action neutre).

Ce type de boucle se rencontre dans certains réglages de débits, par exemple l'alimentation en eau des (générateurs de vapeur) ou l'alimentation en vapeur d'une turbopompe d'une centrale nucléaire, peut-on le trouver dans les domaines pétroliers industriels, gazières industriels, pharmaceutiques industriels, les traitements des eaux, ...

On a choisi pour faire notre étude un dégazeur comme dans l'unité de traitements eaux huileuses CIS à Hassi Messaoud.

### **Objectif**

Dans les labos « Régulation et API » de l'Institut Algérien du pétrole de Boumerdès, on a fait la réalisation d'un dégazeur et appliqué la boucle split-range pour réguler la pression dans l'intérieur de la cuve à une certaine consigne (2 Bar), en utilisant deux vannes régulatrices, l'une d'engagement et l'autre pour le dégazage, un transmetteur de pression qui transmette la valeur de la grandeur physique (pression) vers un API S7-300 et ce dernier gère un signal de manipulation pour commander l'ouverture de vannes ainsi de conserver la pression égale à la consigne.

---

# Chapitre 1

Réalisation d'un dégazeur a l'IAP de  
Boumerdes

**Introduction [1]**

La première entreprise en Algérie et en Afrique, Sonatrach, met tout en œuvre pour perfectionner et spécialiser les différents métiers qu'inclut ses filiales. Elle dispose de plusieurs instituts de formations pour optimiser ses opérations d'apprentissages.

Les différents objectifs sont de former des nouveaux spécialistes dans les différents métiers de base comme le forage, l'engineering, ...etc. Professionnaliser et perfectionner les cadres administrateurs dans les métiers de la comptabilité et l'audit, la finance, le juridique, la fiscalité pétrolière, les ressources humaines et développer les compétences des gestionnaires dans le management des projets, éveiller la culture HSE et perfectionner les compétences des spécialistes en sécurité industrielle. Ainsi que la préparation de la relève dans tous les postes clés aux seins de l'entreprise.

L'IAP dans sa nouvelle structure est composé essentiellement de :

➤ CPE : Centre de perfectionnement de l'Entreprise

Il opère dans les domaines des technologies et techniques de gestion ainsi que les langues vivantes, les finances et le juridique, informatique et systèmes d'information et le HSE. Il s'occupe aussi de la formation de formateurs.

➤ IAP : Algerian Petroleum Institute

Assurant des formations opérationnelles dans le domaine de l'énergie et des mines : amont hydrocarbures, transport hydrocarbures, Aval hydrocarbures, économie pétrolière, management, HSE, génie de l'environnement, finances, ressources humaines, communication d'entreprise, ...

Il dispose de trois établissements à Boumerdès, Oran et Skikda.

**1.1 Missions de l'IAP [1]**

Les principales missions de l'IAP sont :

1. Organisation et mise en œuvre des formations spécialisantes et diplômantes (Opérateurs, Agents de Maîtrise, Techniciens, Techniciens Supérieurs Spécialisés, Ingénieurs d'Application, Ingénieurs Spécialisés et Post graduation).
2. La formation continue par le recyclage et le perfectionnement.
3. La réalisation de prestations de services (Etudes, Analyses et Expertise).
4. La participation à la maîtrise technologique par le développement de la recherche appliquée.
5. L'organisation et l'animation des manifestations à caractère scientifique et technique.
6. Le développement de la documentation scientifique et technique.
7. La mise en place, le développement et la promotion de la formation à distance.

8. La prise en charge de la mémoire du secteur de l'énergie et des mines mise en place d'une structure « projets complétés ».
9. L'établissement d'échanges et de partenariats avec les universités et instituts nationaux et internationaux.
10. Le perfectionnement et le recyclage des Cadres et Techniciens en activité dans le secteur des hydrocarbures.
11. La recherche appliquée orientée en priorité vers les besoins du secteur des hydrocarbures.

1.2 Organisation de l'IAP [1]

1.2.1 Structures opérationnelles de l'IAP

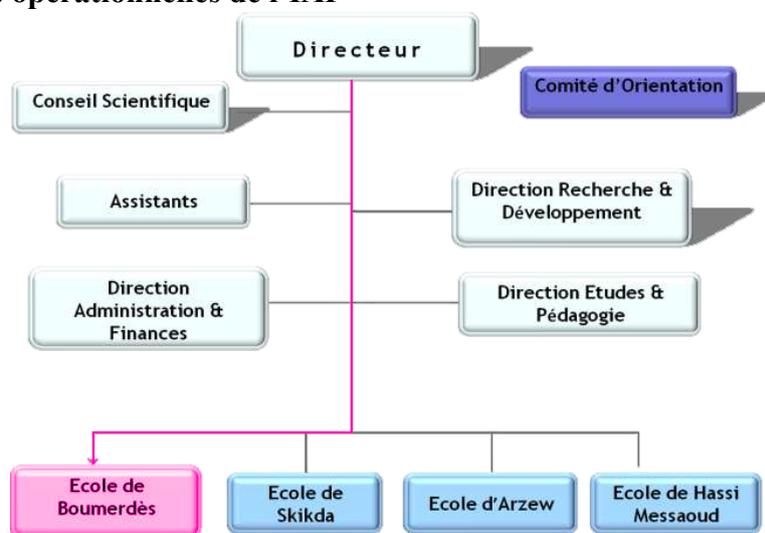


Figure 1.1 : Structures opérationnelles de l'IAP [1]

**Activités de Formation & Recherche**

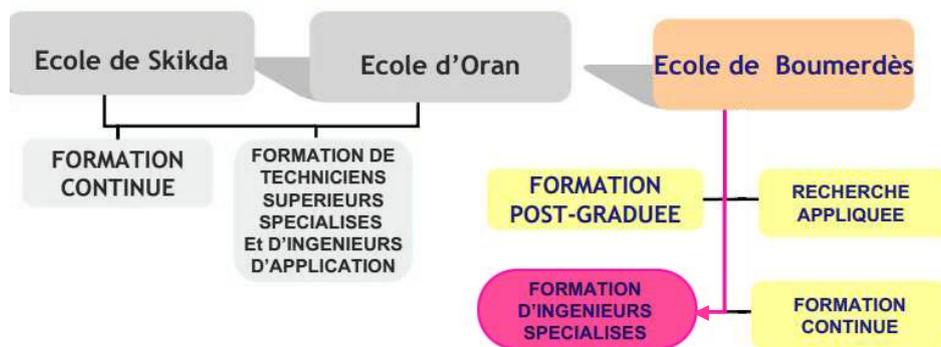


Figure 1.2 : Mise en œuvre de l'IAP [1]

### 1.2.2 Spécialités de l'Ecole de Boumerdès

- ✓ Géosciences (Géologie & Géophysique).
- ✓ Réservoir Engineering.
- ✓ Forage & Production.
- ✓ Exploitation des Hydrocarbures.
- ✓ Raffinage, Pétrochimie.
- ✓ Gaz Naturel Liquéfié.
- ✓ Chimie & Analyse des Hydrocarbures.
- ✓ Instrumentation Pétrolière.
- ✓ Mécanique pétrolière.
- ✓ Procurement/Economie Pétrolière.

### 1.3 L'école de Boumerdès [1]

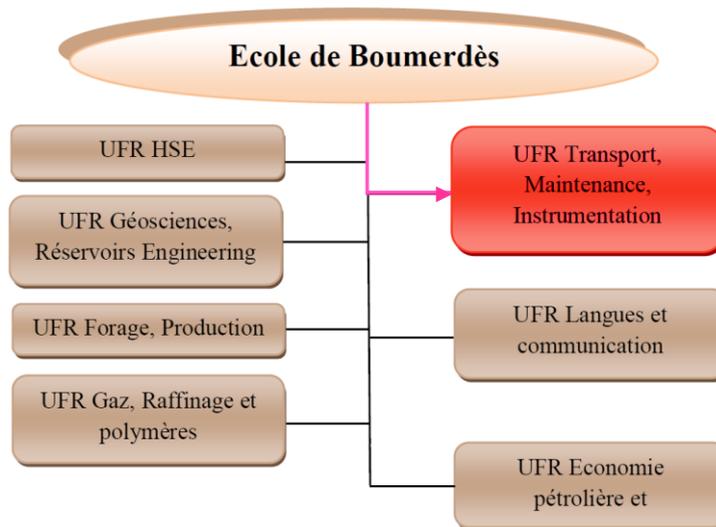
Créée en 1971, l'Ecole est implantée à Boumerdes. Ses missions sont les Formations Graduées et Post-graduées, la Formation Continue (perfectionnement et séminaires) et la Recherche Appliquée.

#### 1.3.1 Ses domaines d'activité

Les domaines d'activité de l'EBM embrassent l'ensemble de la chaîne pétrolière et parapétrolière Géosciences, Géologie pétrolière, Géophysique pétrolière, Forage-Production, Réservoir engineering, Oil and gas processing, Raffinage & Pétrochimie, Génie des polymères, Gaz Naturel Liquéfié, Transport et distribution du gaz, Transport et distribution des hydrocarbures, Maintenance industrielle, Turbomachines, Instrumentation pétrolière, Economie pétrolière et gazière, Marketing des hydrocarbures et dérivés, Management, Génie de l'environnement, HSE, Sécurité industrielle et prévention, Power engineering, Mines et Communication des Entreprises.

#### 1.3.2 Les laboratoires

Essais normalisés, TBP, Banc d'essais moteurs, Moteur CFR, Process control, Instrumentation moderne, Environnement, Analyse instrumentale, Caractérisation des matériaux, Maintenance, Stations de travail Amont, Simulateur de forage, Laboratoire, Multimédia (Langue Anglaise).



**Figure 1.3 :** Mise en œuvre de l'école de Boumerdes [1]

### 1.3.3 UFR Transport, Maintenance & Instrumentation

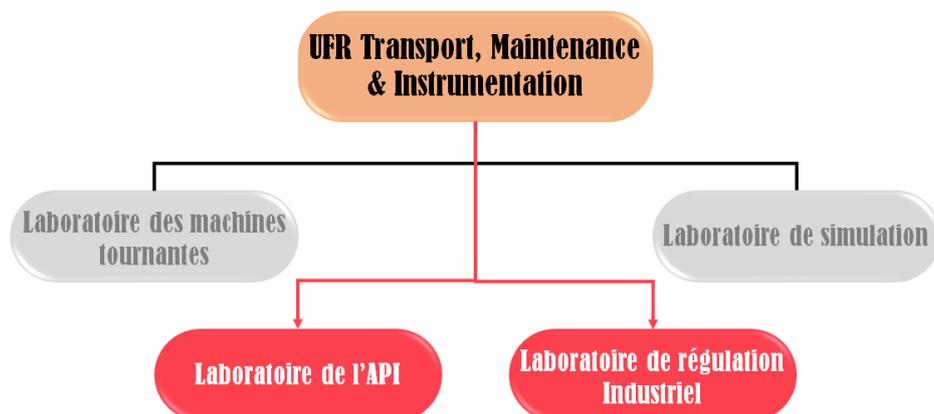
#### 1.3.3.1 Personnel

L'UFR IMT s'appuie sur l'expérience d'un groupe d'enseignants-chercheurs de l'IAP-Spa ainsi que sur celle des intervenants extérieurs qui viennent compléter cette équipe en fonction des besoins.

#### 1.3.3.2 Laboratoires

L'UFR dispose de quatre Laboratoires :

- Laboratoire des machines tournantes.
- Laboratoire de simulation.
- Laboratoire de l'Automate Programmable Industriel (API).
- Laboratoire de régulation Industriel.



**Figure 1.4 :** Les laboratoires de l'UFR TMI [1]

Les rectangles arrondis colorés en rose dans la (figure 1.5) présente les laboratoires de l'UFR TMI où nous avons fait l'étude de la boucle split range et la réalisation d'un dégazeur.

L'utilisation de la boucle split range est dans plusieurs domaines industriels, peut-on le trouver dans les domaines pétroliers industriels, gazières industriels, pharmaceutiques industriels, les traitements des eaux, ...

Le dégazeur de l'unité de traitements eaux huileuses CIS à Hassi Messaoud, est approprié pour faire l'étude de la boucle split range.

#### 1.4 Description de CIS [2]

Le complexe industriel CIS situé au sud du champ de Hassi Messaoud, reçoit la production totale en huile de la zone sud, cette production provient essentiellement des unités satellites d'une part et directement des puits en LDHP (ligne directe haute pression), en LDBP (ligne directe basse pression) et en LDMP (ligne directe moyenne pression) d'autre part.

Le complexe CIS est composé d'une chaîne de procédés plus ou moins complexe ont pour fonction le traitement des effluents en provenance des puits producteurs. Ces procédés sont faits pour prendre en charge :

- Le traitement de l'huile : séparation huile- eau- gaz, dessalage, et stabilisation.
- Le traitement des gaz associés pour la production des GPL et condensât.
- Le traitement des eaux huileuses pour la protection de l'environnement.
- Le raffinage d'une partie du brut pour la production de carburants.
- La réinjection des gaz résiduels pour le maintien de la pression du gisement.
- L'injection d'eau pour le maintien de la pression du gisement.

##### 1.4.1 Unité de traitements eaux huileuses CIS

L'unité de traitement est composée d'équipements suivants :

- ✓ Le Séparateur triphasique.
- ✓ Le Dégazeur.
- ✓ Les bassins API.
- ✓ Les Puisards et pompes de transfert des eaux huileuses.
- ✓ Les Puisards et pompes de transfert des huiles récupérées.
- ✓ Les skimmers.
- ✓ Les skids d'injection des produits chimiques.
- ✓ Le Séparateur de flottation à gaz induit « IGF ».
- ✓ Le Séparateur à gaz dessous « DGF ».
- ✓ Les unités de filtration.
- ✓ Pompes d'expédition.

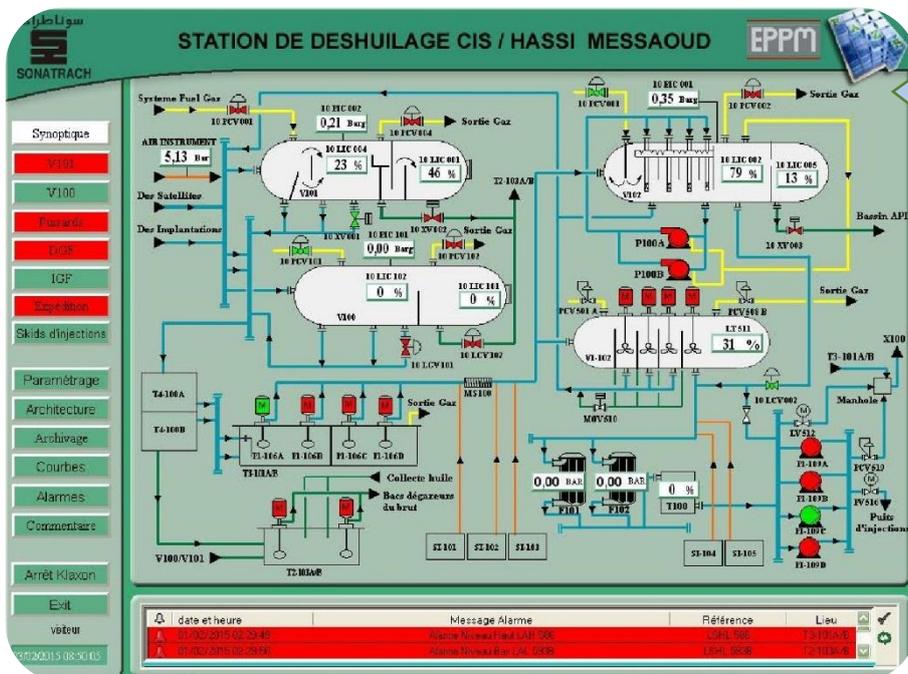


Figure 1.5 : Unité de traitement des eaux huileuses SH-DP-CIS [2]

### 1.4.1.1 Le séparateur triphasique

Le séparateur triphasique se compose de cinq compartiments principaux séparés par des chicanes métalliques :

- Compartiment d'admission : récupération du grand pourcentage des particules solides en provoquant leur décantation.
- Compartiment central de tranquillisation.
- Compartiment de séparation gaz-liquide et liquide-liquide.
- Compartiment de séparation huile/eau traitée.
- Compartiment de récupération d'eau traitée.

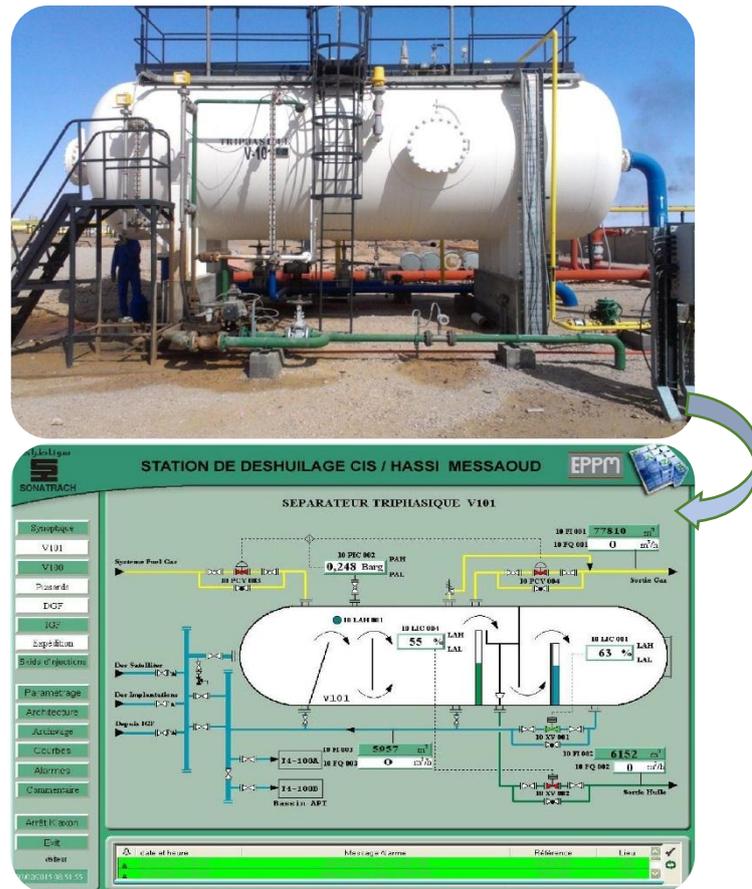


Figure 1.6 : le séparateur triphasique [2]

### 1.4.1.2 Le dégazeur

Cet équipement a été placé en amont de la station de déshuilage, il permet la séparation des gaz et quelques quantités d'huile provenant des diverses unités de traitement du centre industriel Sud. Il a été construit et installé par la Direction Technique Sonatrach. Cet équipement a été installé avant l'installation du séparateur triphasique.

Vu son faible rendement, le dégazeur fait actuellement la redondance du séparateur triphasique dans les cas de panne ou d'intervention pour nettoyage de longue durée.



Figure 1.7 : Le dégazeur [2]

### 1.5 l'étude de notre boucle de régulation split range

On a divisé le travail a deux parties : la régulation split range et la surveillance des alarmes.

#### 1.5.1 la régulation split range

Pour faire le montage du dégazeur et appliquer la boucle split range, on a utilisé les équipements suivants :

- Automate Programmable (API) S7-300 avec une CPU 313C.
- Capteur de pression caractérisé par 4-20mA de sortie et l'étendu d'échelle de 0 à 4 Bar.
- Deux vannes régulatrices (la vanne d'engazage pneumatique et la vanne de dégazage a servomoteur).
- Deux vannes manuels (simples).
- Deux réservoirs de pression (afin de garder le volume approprié pour l'opération).
- Un ordinateur (pour charger le programme sur l'automate et faire les tests).

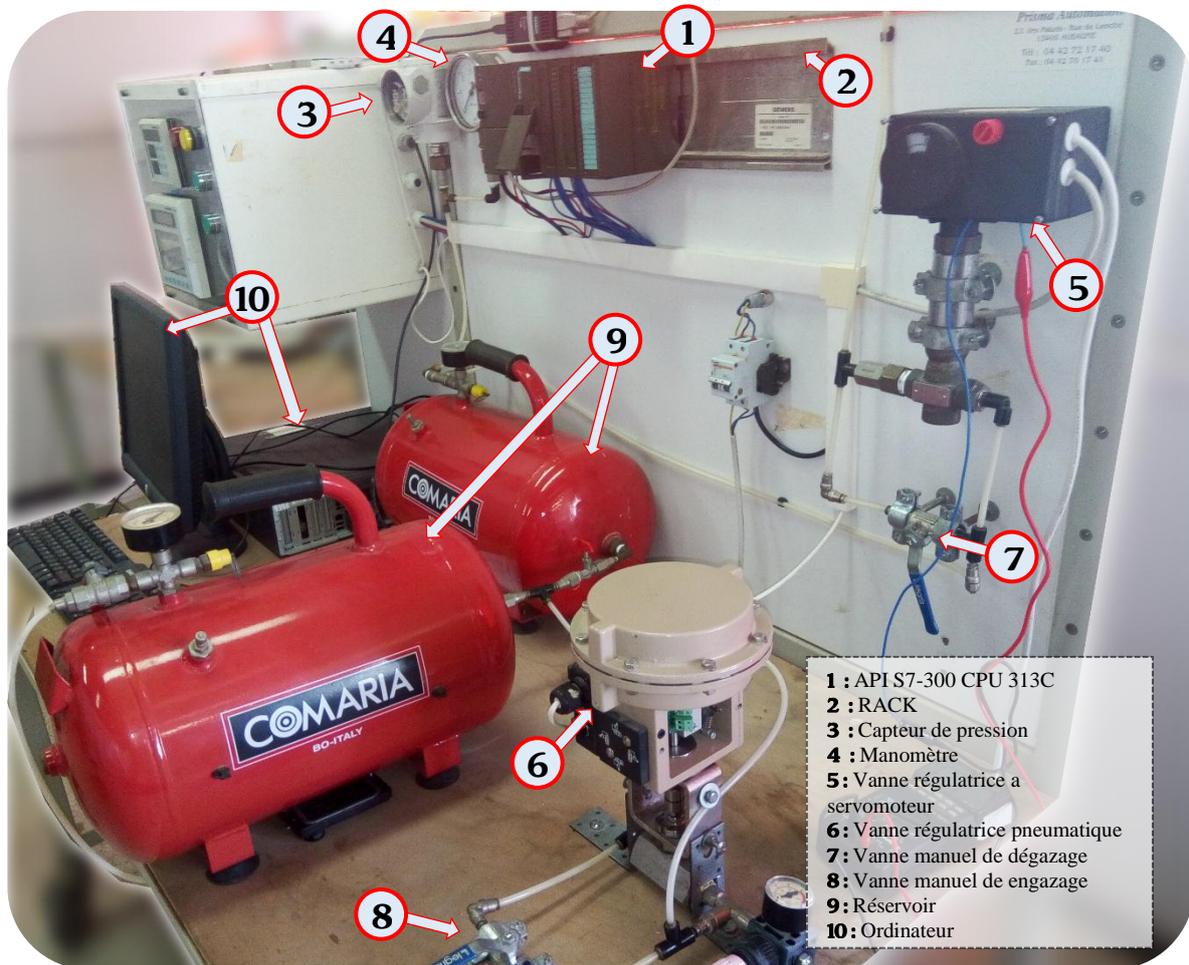
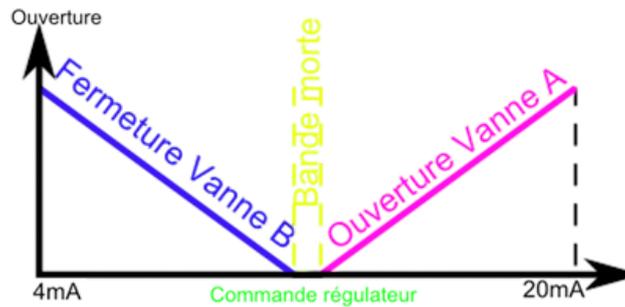


Figure 1.8 : Notre dégazeur

La pression du réservoir est mesurée, et renvoyée vers un régulateur PIC (l'automate). La sortie de celui-ci est envoyée vers deux actionneurs à actions opposées, ici une vanne d'engazage du réservoir (actionneur A) et une vanne de dégazage (actionneur B).

Afin que les deux vannes ne soient pas ouvertes en même temps, la commande des vannes est adaptée à la sortie du régulateur.

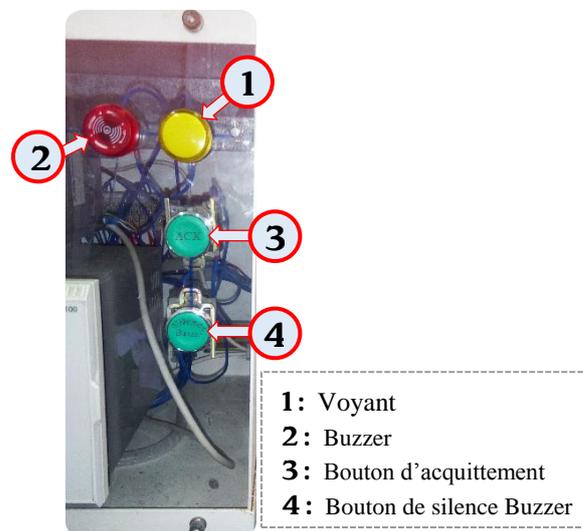


**Figure 1.9 :** Régulation de pression en split-range

### 1.5.2 la surveillance des alarmes

La surveillance des alarmes est aussi importante que la régulation, pour cela on a étudié et préparé le cahier des charges suivant :

- Un voyant clignotant d'une alarme « H\_ALM » surveillant le seuil haut de pression de la cuve, celle-ci doit s'enclencher si la pression de 3.2 bar apparaît et se déclencher sauf si on appuie le bouton ACK.
- Prévoir une information pour la maintenance en cas de dépassement de l'étendue d'échelle du capteur (valeur haute ou valeur base).
- Prévoir la fonction alarme sonore (buzzer) en discontinue pour le « HH\_ALM » avec leur acquittement (silence buzzer).



**Figure 1.10 :** La surveillance des alarmes

### 1.6 La supervision

Généralement, la salle de control regroupant les dispositifs permettant de surveiller les activités d'une installation industrielle et de réguler son fonctionnement.

On trouve aussi des interfaces informatiques qui permet à un usager de manipuler la machine. Elle coordonne les interactions homme-machine et fait partie du design industriel.

Pour cela on a prévu une visualisation de notre procédé avec le progiciel WinCC flexible 2008.

**Conclusion**

Dans ce chapitre on a présenté l'Institut Algérien de Pétrole, différents domaines d'activité et les différents laboratoires. Ainsi que les équipements de notre dégazeur et le comportement dans une boucle de régulation split range avec la surveillance des alarmes.

Dans le chapitre suivant, on s'intéresse de présente la régulation industrielle et les différents types de la boucle split range.

# Chapitre 2

Régulation industrielle et boucle de  
régulation split range

## Introduction

La régulation est au cœur de toutes nos actions : conduire sa voiture, régler la température de sa douche le matin, réaliser une recette de cuisine ... etc. à l'industrie aussi la régulation est une partie essentielle qu'on peut trouver dans toutes les articulations des opérations industriels [3].

Dans ce chapitre, on va présenter la régulation industrielle, le principe de régulation, les types de procédés, les types de régulateur, la régulation PID, les différents types de calculs et les boucles de régulation. C'est dans cette dernière partie qu'on va présenter les différents types de la boucle split range et comment l'appliquer sur le dégazeur qui nous avons réalisé dans l'institut algérien de pétrole (IAP), école de Boumerdès.

### 2.1 Régulation Industrielle [3]

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

Un système asservi est un système dit suiveur, c'est la consigne qui varie (exemple : une machine-outil qui doit usiner une pièce selon un profil donné, un missile qui poursuit une cible). Dans le cas de la régulation, la consigne est fixée et le système doit compenser l'effet des perturbations, à titre d'exemple, le réglage de la température dans un four, de la pression dans un réacteur, le niveau d'eau dans un réservoir.

L'industrie utilise à foison des systèmes d'asservissement ou de régulation :

Que ce soit pour gérer le débit d'un fluide dans une conduite, la température d'un produit, la hauteur d'un niveau de cuve...

Historiquement, les régulateurs n'étaient pas intégrés dans l'unité centrale des Automates Programmables Industriels, mais se présentaient sous forme de modules autonomes gérant leur environnement propre (acquisition, calcul, commande...).

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

Les régulateurs permettent ainsi de lier plus simplement les parties séquentielles et continue du procédé. La régulation fait partie intégrante de la qualité de production : c'est donc un point non négligeable de la chaîne de valeurs d'une installation.

## 2.2 Principe de régulation [3]

### 2.2.1 La chaîne de traitement de l'information

Le contrôle de procédé met en œuvre une chaîne d'outils à plusieurs niveaux. Schématiquement, le lien entre le capteur et l'actionneur peut se représenter comme ci-dessous. Le but étant de maîtriser le procédé et, soit de le piloter, soit de l'empêcher d'évoluer en raison de perturbations.

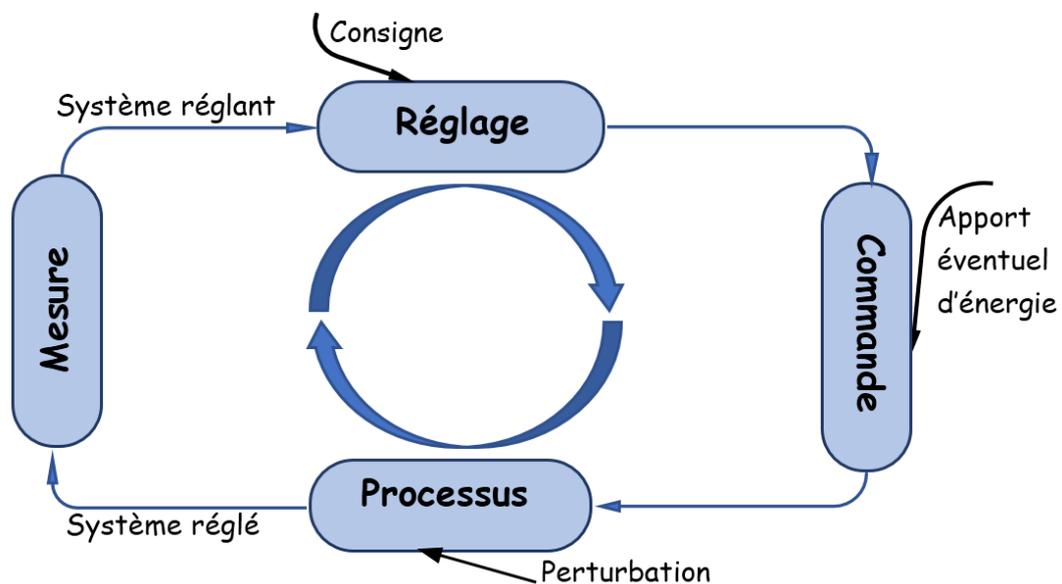


Figure 2.1: les éléments de la chaîne de régulation

### 2.2.2 Les éléments de la chaîne

Le système se décompose en trois zones : mesure, algorithme et commande.

- Le mesure englobe le capteur, l'acquisition par l'automate et le traitement de la mesure.
- L'algorithme correspond à la partie calcul de la commande, en respectant les consignes.
- La commande représente l'écriture sur la carte de sortie ainsi que l'actionneur.

#### 2.2.2.1 Le capteur

*C'est l'élément d'interface avec le réel.*

L'automate programmable sait uniquement acquérir des informations électriques. Il est donc nécessaire de transformer l'information à traiter en valeurs de tension ou de courant représentatives.

C'est le rôle du capteur qui est en réalité composé de deux parties : capteur et transmetteur.



**a** : Transmetteur de pression avec Manomètre



**b** : Transmetteur de pression avec Afficheur LCD

**Figure 2.2:** Capteur et transmetteur

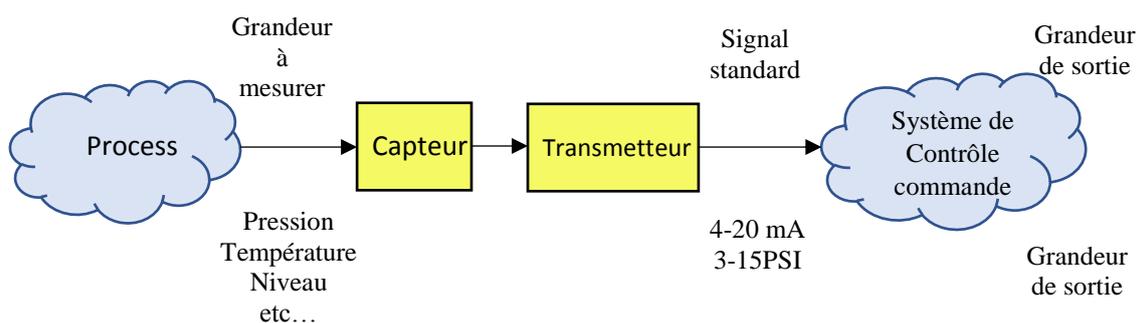
### 2.2.2.2 L'acquisition de la mesure

*L'entrée dans le monde numérique.*

Le processeur de traitement gère exclusivement des éléments appartenant au monde numérique.

Il est donc nécessaire de convertir la mesure en une valeur mémorisable dans un champ de bits. Les capteurs numériques réalisent directement cette conversion et envoient la valeur binaire correspondante.

On verra que l'acquisition réalisée par une carte d'entrée présente des caractéristiques dues à l'échantillonnage et à la quantification.



**Figure 2.3:** acquisition de la mesure

### 2.2.2.3 Le traitement de la mesure

*La valeur brute est rarement utilisable.*

En général, la valeur acquise n'est pas directement exploitable. En effet, des couplages CEM peuvent avoir généré des bruits parasites, la mesure peut avoir été réalisée avec une stratégie qui impose une opération mathématique pour la rendre utilisable...

Cette partie, si elle n'est pas systématique, doit cependant être maîtrisée car elle fournit les bases des informations pour la suite.

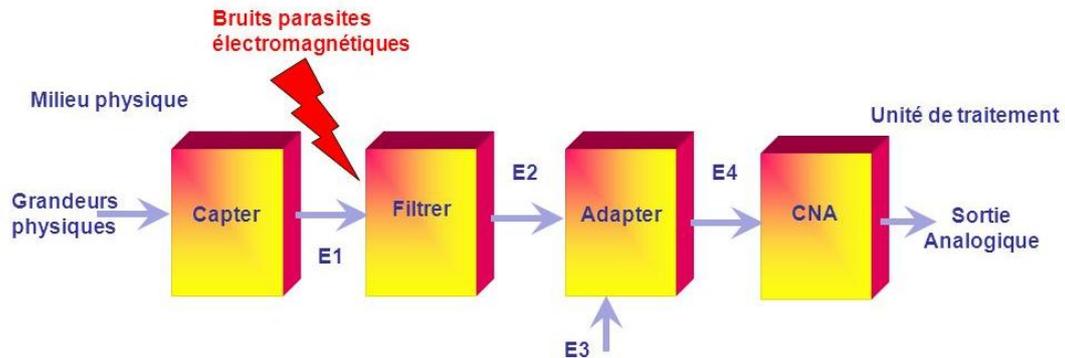


Figure 2.4: Traitement de la mesure avant l'utilisation

### 2.2.2.4 Le traitement de l'algorithme de régulation

*Le cerveau de la boucle.*

A partir des informations actuelles et de la stratégie de commande choisie, l'algorithme va définir les actions à effectuer. C'est en général la partie la mieux traitée par les développeurs bien qu'elle ne puisse pas toujours résoudre des problèmes antérieurs qui auraient été négligés.

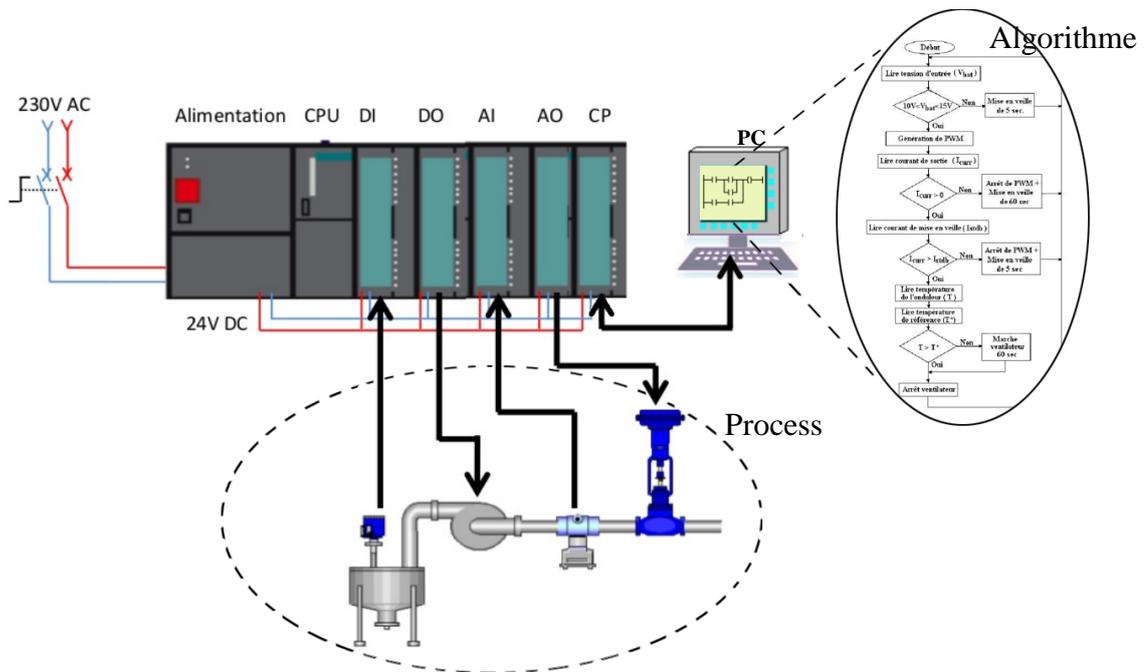


Figure 2.5 : Algorithme qui va définir les actions à effectuer

### 2.2.2.5 La commande

*L'écriture vers l'extérieur.*

La valeur calculée par l'algorithme de régulation n'est pas obligatoirement envoyée directement au procédé. Elle peut être modifiée (cas des servo-moteurs) ou contrainte. La valeur

finale est ensuite envoyée sur une carte de sortie pour être convertie en valeur électrique (TOR ou analogique).

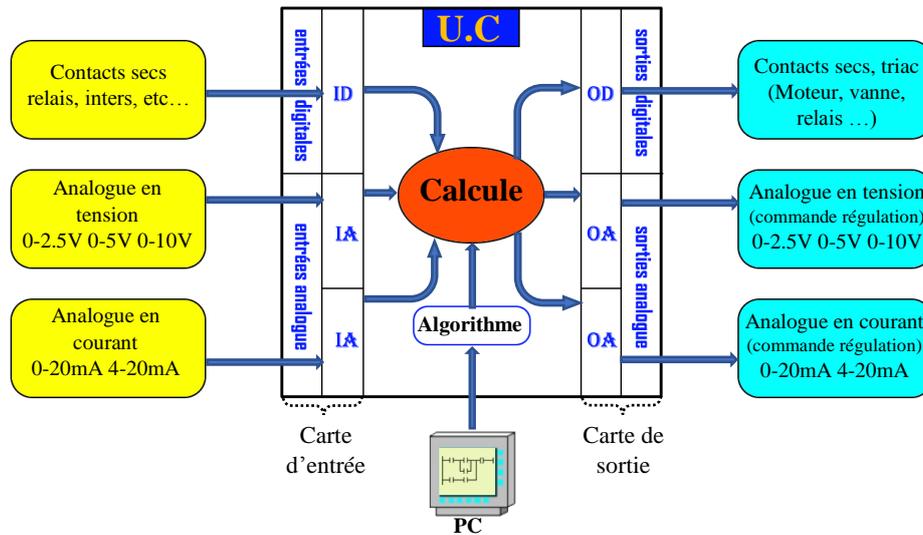


Figure 2.6 : L'écriture vers l'entrée et l'écriture vers l'extérieur

### 2.2.2.6 L'actionneur

*La connexion avec le procédé.*

C'est l'interface d'action sur le procédé. Cela peut être une vanne, une pompe, ...etc.



Figure 2.7 : Les actionneurs industriel [4]

### 2.2.2.7 Le procédé

Dans la dernière étape, le procédé fournit une nouvelle valeur au capteur à partir de la commande passée mais aussi à partir des perturbations qui vont le solliciter.

La connaissance de son comportement est donc nécessaire.

L'évolution interne (exemple : montée adiabatique dans un réacteur) sera plus souvent vue comme une perturbation, plutôt que comme un élément de pilotage. Le procédé pourra être stable ou instable.

## 2.3 Les types de procédés [5]

### 2.3.1 Continu

Dans un procédé continu le produit fini est élaboré d'une façon ininterrompue, c'est le cas du dépropaniseur de la figure suivante où à partir d'une charge (alimentation) butane/propane, introduite en continu dans la colonne de fractionnement, on soutire de façon continue, le propane en tête et le butane en fond de colonne.

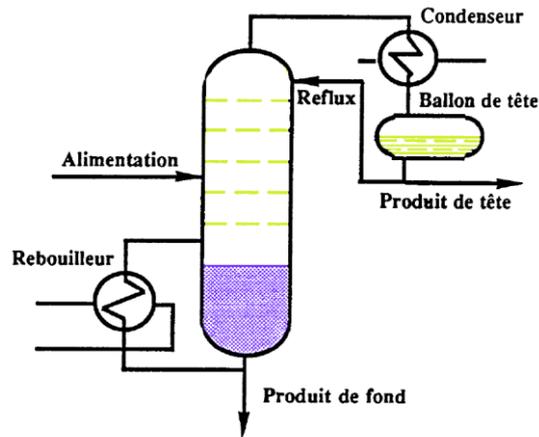


Figure 2.8 : Un exemple de procédé continu (Dépropaniseur)

### 2.3.2 Discontinu

Les procédés discontinus sont également appelés « BATCH » ou « par lot », installation où le produit fini est obtenu en quantité déterminée lors d'une seule procédure de fabrication. En fait, ces procédés sont « continus par partie » puisqu'il s'agit des suites d'opérations discrètes (chargement soutirage...) et de phases continues (réaction...). L'exemple type est le réacteur : chargement, réaction, vidange, puis à nouveau, déroulement du cycle suivant.

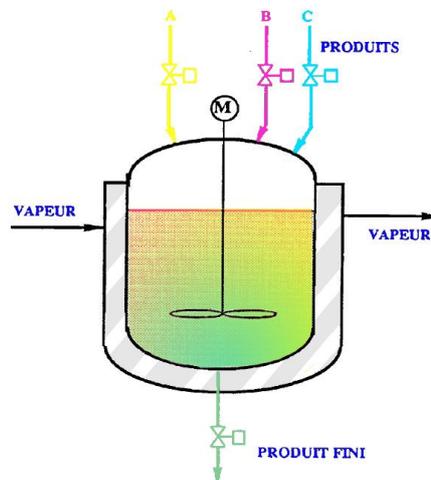
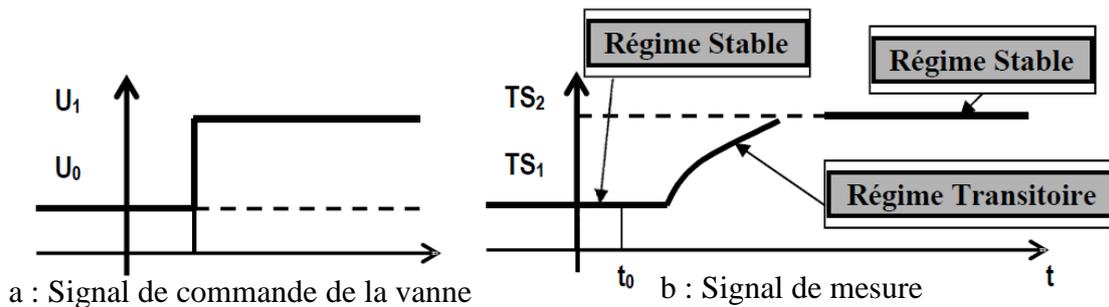


Figure 2.9 : Un exemple de procédé discontinu (Autoclave)

### 2.3.3 Stable [3]

Procédé naturellement stable qui fournit, pour un échelon de commande, une variation finie de la mesure repositionnant le procédé dans un état constant.

Exemple : échangeur thermique. En cas de variation de la commande, la mesure prend une nouvelle valeur stable.



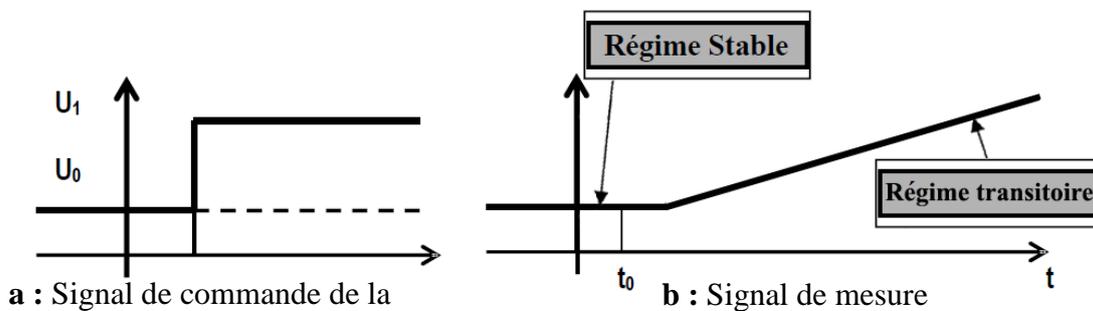
**Figure 2.10 :** Régulateur en mode manuel d'un process stable

### 2.3.4 Instable [3]

Procédé instable dans le cas où le procédé ne reste pas dans un état stable suite à un échelon de commande.

Exemple : le remplissage de cuve. Si l'on fait un échelon sur la commande (ouverture de la vanne de remplissage), la mesure de niveau augmente infiniment.

La stabilité ou l'instabilité d'un procédé se détermine sans régulation. Il est clair que la régulation a pour but de rendre stable un procédé instable ou d'accélérer un procédé stable.



**Figure 2.11 :** Régulateur en mode manuel d'un process instable

## 2.4 Type de système dynamique

### 2.4.1 Mono variable [3]

Un système mono variable est un système présentant un bouclage entre une commande (et une seule) et une mesure.

Exemple : pour arroser sa pelouse, la pression en sortie du jet se règle avec le robinet d'arrivée.

Rien d'autre ne fait varier la pression.

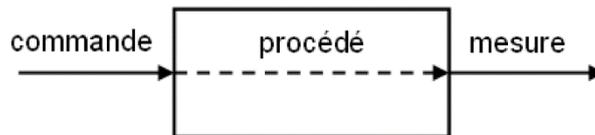


Figure 2.12 : Système mono variable

### 2.4.2 Multi variable [3]

C'est le cas d'un procédé où une mesure peut être affectée par plusieurs commandes.

Exemple : la température d'un jet d'eau se règle grâce à un robinet d'eau chaude et à un robinet d'eau froide ; il s'agit d'une configuration multi-variable simple (la même température pouvant être atteinte par des choix différents sur l'eau chaude et sur l'eau froide).

En revanche, le problème est plus complexe si l'on désire respecter la température et le débit. Dans ce cas, on ne peut plus régler chaque actionneur sans tenir compte de l'autre. Toute variation sur un actionneur fait varier la température et le débit.

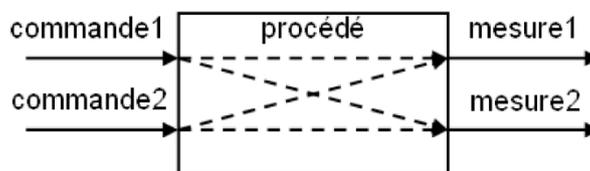


Figure 2.13 : Système multi variable

## 2.5 La régulation PID [3]

### 2.5.1 Les actions

#### 2.5.1.1 Action Proportionnelle

Cette action est la plus classique. Elle consiste à appliquer une correction en rapport avec la différence instantanée entre la mesure et la consigne courante.

Le rapport Erreur/Commande s'appelle le gain (ou 1/Bande proportionnelle).

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

On a donc l'équation :  $S = Gr * (M-C)$ .

Dans le cas d'un régulateur inverse, l'équation sera  $S = -Gr * (M-C)$ .

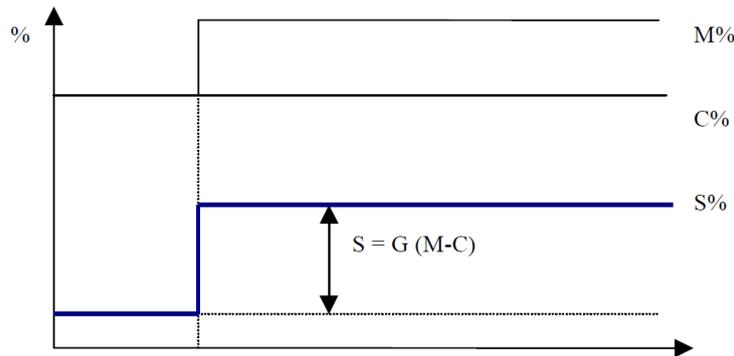
Les termes employés pour désigner le gain pourront être :  $G, K, Kp \dots$

Dans le cas de la bande proportionnelle, on aura :  $BP, PB, XP \% \dots$

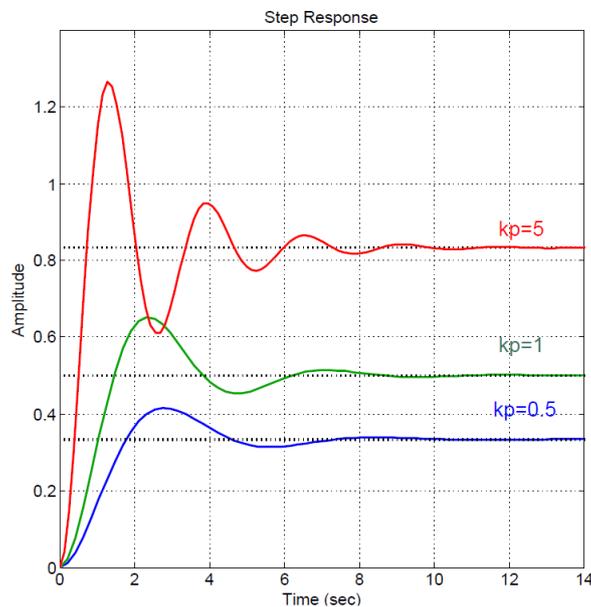
L'équation de transfert du régulateur s'écrit  $S = +/- G (M-C) + So$

$So$ , accessible sur certains régulateurs, est appelé décalage ou bande centrale. C'est la valeur du signal de sortie lorsque  $M=C$ .

A l'instant  $t=0$ , on crée un écart (mesure – consigne) sous forme d'échelon en donnant une impulsion sur la consigne.



**Figure 2.14 :** Impulsion sur la consigne



**Figure 2.15 :** Action Proportionnelle [6]

L'action proportionnelle seule ne permet pas en général à la mesure de rejoindre la valeur de consigne. Une augmentation du gain entraîne une réduction de l'écart résiduel mais peut entraîner une instabilité de la mesure. Celle-ci peut devenir oscillante, voire même divergente

### 2.5.1.2 Action Intégrale

Cette action calcule sa contribution à la commande en n'utilisant pas l'écart instantané mais une mémoire des écarts intervenus lors des précédents échantillonnages. Ceci permet d'éliminer à la fois les erreurs statiques trop petites pour que l'action proportionnelle les contre et les erreurs de suivi de consignes en pente. La limite de cette commande intervient à partir du moment où le système ne réagit plus à la commande passée (blocage d'organe, limitation de sécurité, etc.). Dans ce cas précis, le régulateur va amplifier à l'infini sa contribution intégrale.

On parle de saturation d'intégrale. La résolution de problèmes passe de l'utilisation d'un PID à intégrale externe.

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

Le réglage de la contribution intégrale peut se faire de deux façons : soit réglage du  $T_i$ , soit réglage de « n » qui est le nombre de fois où l'on somme l'entrée durant l'unité de temps (1 minute ou 1 seconde). Nous avons la relation  $T_i = 1/n$ .

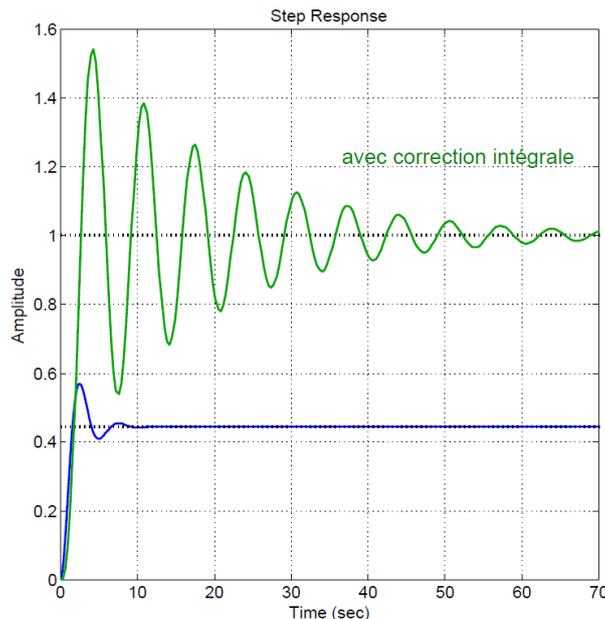


Figure 2.16 : Action Intégrale [6]

### 2.5.1.3 Action Dérivée

Cette action surqualifie la commande à passer en la calculant à partir de la variation de l'erreur. Sur variation de l'erreur, l'action dérivée va intervenir mais s'annulera pour une erreur constante. L'utilisation de l'action dérivée impose d'éliminer tous les bruits entachant la mesure sous peine de voir son système en mouvement permanent.

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

Le réglage de la dérivée se fera en donnant  $T_d$  qui est un coefficient de multiplication de la valeur de la dérivée instantanée. La commande sera donc  $S = T_d (dE/dt)$ .  $T_d$  est exprimé en unité de temps car  $S$  est une amplitude alors que  $dE/dt$  est une vitesse.

### 2.5.2 Les différents types d'implémentations

Plusieurs implémentations des calculs sont possibles : les structures Série, Parallèle, Mixte.

2.5.2.1 Structure PID Série

Les calculs sont réalisés en série.

$$S = Gr \left( \frac{Ti + Td}{Ti} \right) (M - C) + \frac{Gr}{Ti} \int (M - C) dt + Gr * Td \frac{d(M - C)}{Td} \quad (2.1)$$

Il existe un lien fort entre les paramètres puisque le réglage de l'action Proportionnelle Gr influe (par exemple) sur l'action Intégrale et que Ti influe sur l'action Proportionnelle.

En peu écrire aussi :

$$S = Kp \left( \frac{Ti + Td}{Ti} \right) (PV - SP) + \frac{Kp}{Ti} \int (PV - SP) dt + Kp * Td \frac{d(PV - SP)}{Td} \quad (2.2)$$

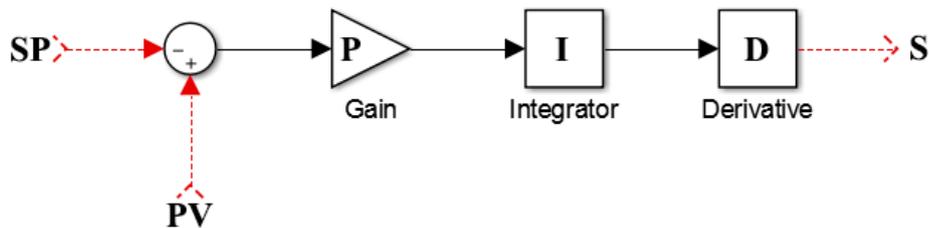


Figure 2.17 : Structure PID Série

2.5.2.2 Structure PID Parallèle

Les calculs sont réalisés en parallèle et une somme est effectuée sur les différentes contributions pour avoir la commande à appliquer.

$$S = Gr(M - C) + \frac{1}{Ti} \int (M - C) dt + Td \frac{d(M - C)}{Td} \quad (2.3)$$

Une grande liberté de choix de réglages est offerte.

Cependant, cette structure est peu utilisée car elle est plus complexe à régler.

On peut écrire aussi :

$$S = Kp(PV - SP) + \frac{1}{Ti} \int (PV - SP) dt + Td \frac{d(PV - SP)}{Td} \quad (2.4)$$

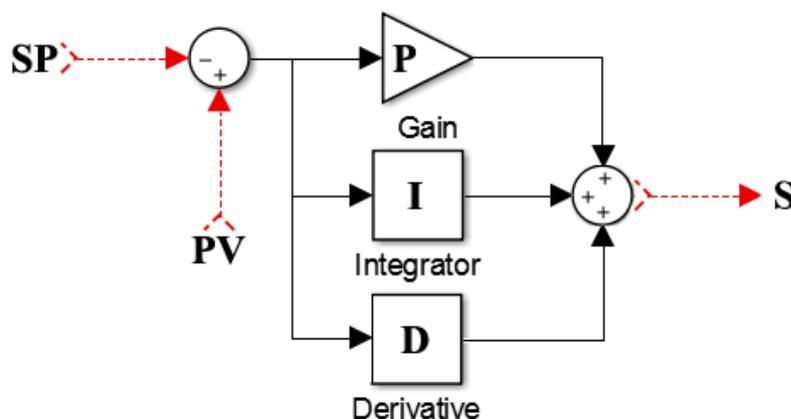


Figure 2.18 : Structure PID Parallèle

2.5.2.3 Structure PID Mixte

Les calculs sont réalisés en parallèle pour les contributions Intégrale et Dérivée et la somme de ces actions est mise en série avec la contribution Proportionnelle.

$$S = Gr(M - C) + \frac{Gr}{Ti} \int (M - C)dt + Gr * Td \frac{d(M - C)}{Td} \quad (2.5)$$

C'est la structure la plus classique qui peut être écrite comme suit :

$$S = Kp(PV - SP) + \frac{Kp}{Ti} \int (PV - SP)dt + Kp * Td \frac{d(PV - SP)}{Td} \quad (2.6)$$

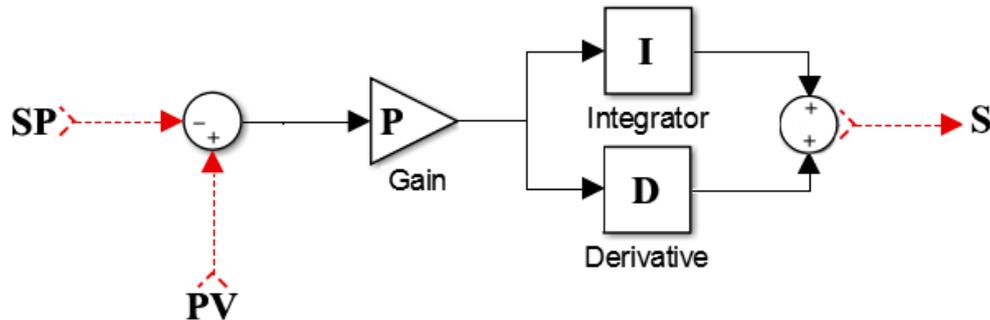


Figure 2.19 : Structure PID Mixte

2.5.3 L'influence des coefficients [7]

Kp, Ki et Kd sont les coefficients d'un régulateur PID, le tableau dessous présente l'influence des coefficients.

Tableau 2.1 : Récapitulatif de l'action des coefficients

Coefficient	Temps de Montée	Temps de Stabilisation	Dépassement	Erreur Statique
Kp	Diminue	Augmente	Augmente	Diminue
Ki	Diminue	Augmente	Augmente	Annule
Kd	-	Diminue	Diminue	-

2.6 Les différents types de calculs

Deux types de calculs peuvent être effectués : soit en calculant la commande à appliquer en ne tenant compte que de l'écart instantané Mesure Consigne, soit en calculant l'incrément (suivant la variation de l'écart) à appliquer à la commande actuelle.

La première technique s'appelle Absolu, la seconde Incrémental.

2.6.1 L'algorithme Absolu

A chaque période Δ, toute l'action PID est recalculée.

$$out = P(t) + I(t) + D(t) \quad (2.7)$$

avec

$$e(t) = PV - SP$$

$$P(t) = Kp * e(t)$$

$$I(t) = I(t - \Delta) + \frac{Kp * \Delta * e(t)}{Ti}$$

$$D(t) = \frac{(Kp * Td)}{Td + Kp * \Delta} * [e(t) - e(t - \Delta) + \frac{D(t - \Delta)}{Kd}]$$

On voit que si  $e(t)$  est constant,  $I(t)$  va s'incrémenter jusqu'à l'infini. Un algorithme spécial gère ce cas et bloque  $I(t)$  à une valeur limite.

Cet algorithme ne permet pas d'assumer des contraintes sur la commande. C'est le mode de calcul le plus classique.

### 2.6.2 L'algorithme Incrémental

A chaque période  $\Delta$ , seul l'incrément de commande à appliquer est calculé.

$$out(t) = out(t - \Delta) + outd \quad (2.8)$$

avec

$$outd = \Delta P(t) + \Delta I(t) + \Delta D(t)$$

$$e(t) = PV - SP$$

$$\Delta P(t) = Kp * (e(t) - e(t - \Delta))$$

$$\Delta I(t) = Kp * \frac{\Delta e(t)}{Ti}$$

$$\Delta D(t) = D(t) - D(t - \Delta)$$

Remarque : les deux algorithmes sont similaires tant que la commande passée suit le "out" du PID, mais on voit que l'anti-saturation est implicite.

Ce type de PID est dit à intégrale externe car certaines implémentations ne fournissent que "outd" et le développeur doit intégrer ces incréments de façon à avoir la commande à passer.

## 2.7 Les boucles de régulations [8]

### 2.7.1 Régulation en boucle ouverte

Autres appellations : Régulation à priori, Régulation prédictive et Feedforward control...

Un système en boucle ouverte ou contrôle ouvert est une forme de contrôle d'un système qui ne prend pas en compte la réponse de ce système.

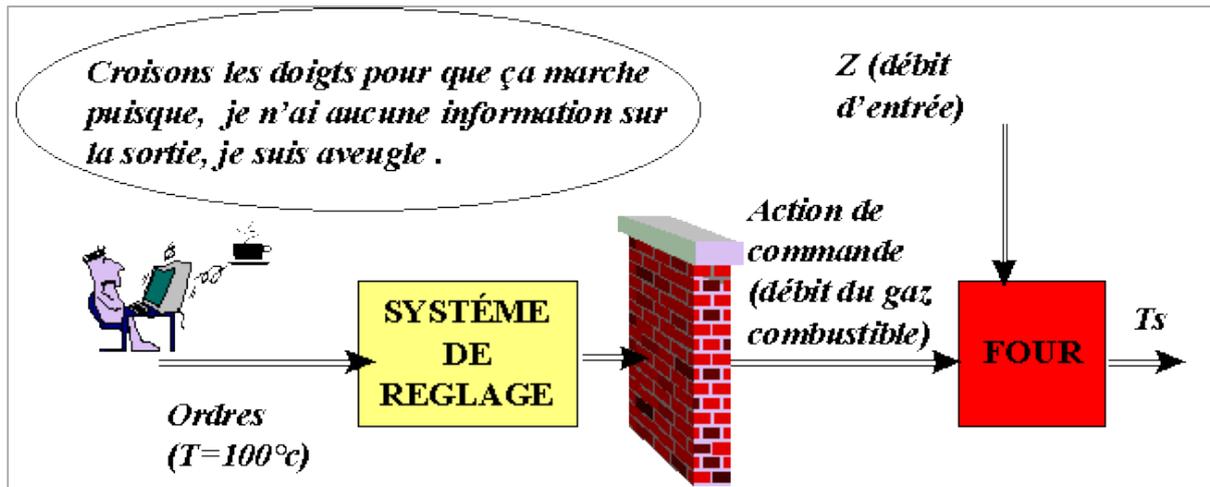


Figure 2.20 : Régulation de la température d'un four en boucle ouverte [9]

Ce contrôle, simple en principe, est à utiliser avec précaution si le système est naturellement instable. Pour le mettre en place il faut au préalable avoir parfaitement modélisé le système, que la commande soit parfaitement adaptée et qu'il n'y ait aucune perturbation.

L'opposé du contrôle en boucle ouverte est le contrôle en boucle fermée, qui lui intègre la rétroaction du système qui est en général négative (contre-réaction)

### 2.7.2 régulation en boucle fermée

Dans ce type de régulation, l'action correctrice s'effectue après que les effets des grandeurs perturbatrices aient produit un écart entre la mesure et la consigne. Cet écart peut être également provoqué par un changement de consigne. Dans les deux cas, le rôle de la boucle fermée est d'annuler l'écart.

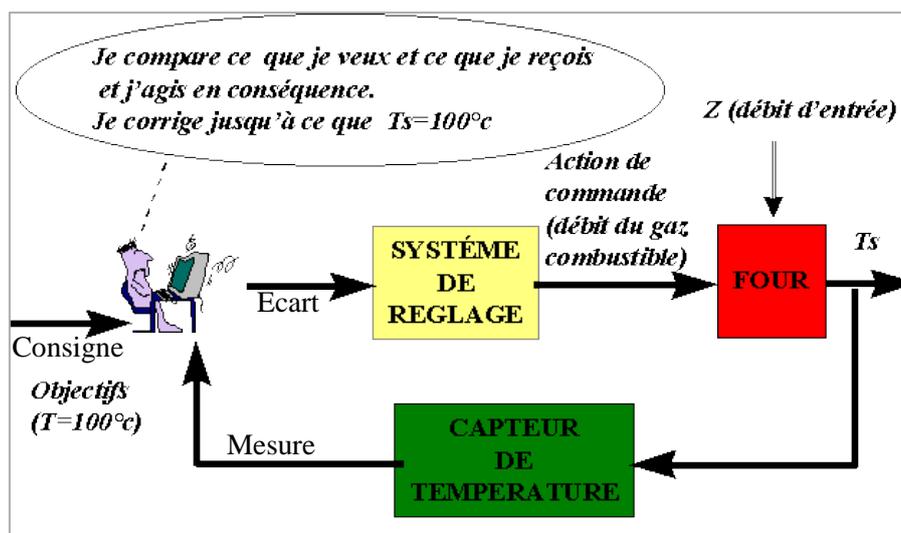


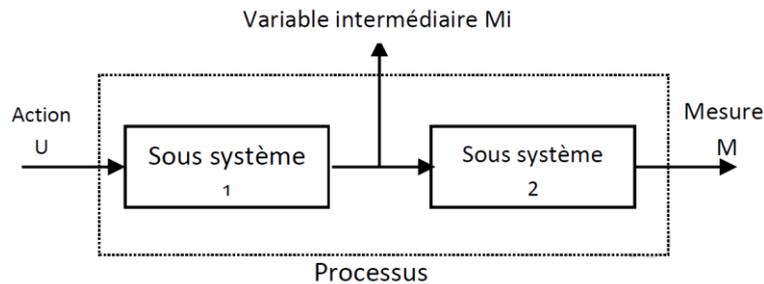
Figure 2.21 : Régulation de la température d'un four en boucle fermée [9]

Il y a des autres appellations comme : Boucle fermée simple, Régulation à posteriori, Feedback control ...

### 2.7.3 régulation cascade [10]

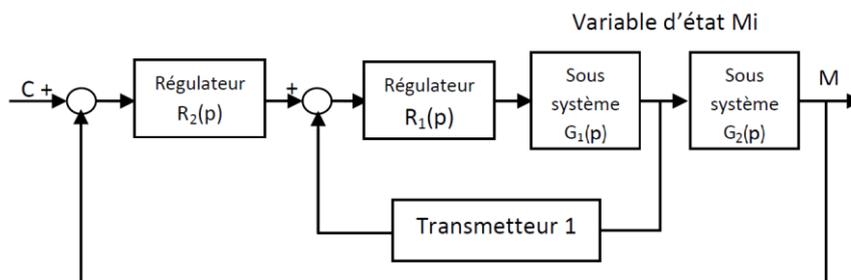
La régulation cascade est une technique utilisée pour permettre aux procédés qui ont une dynamique lente d'avoir une réponse rapide face aux perturbations extérieures ainsi qu'aux changements de consigne.

L'idée repose sur la décomposition d'un processus complexe en plusieurs sous-systèmes. Sans perte de généralité, on suppose pour fixer les idées une décomposition en deux sous-systèmes comme le montre la figure suivante :



**Figure 2.22 :** processus complexe en deux sous-systèmes

La variable intermédiaire  $M_i$  généralement appelée variable d'état, possède la propriété d'être en avance temporelle par rapport à la grandeur de mesure  $M$ . Si par exemple une modification sur l'action  $U$  ou une perturbation affectant le sous-système 1, la variable d'état  $M_i$  sera la première à être affectée avant la variable de mesure  $M$ . En d'autres termes, la grandeur d'état  $M_i$  permet de renseigner sur l'état futur de  $M$ . Cette forme de prédiction peut être exploitée judicieusement pour réaliser une régulation cascade.



**Figure 2.23 :** Régulation cascade

### 2.7.4 régulation de rapport [11]

La régulation de rapport est une technique utilisée pour réguler une variable de procédé sur une consigne qui est calculée comme étant une proportion d'une entrée quelconque (grandeur menante).

La consigne de rapport détermine la proportion de la valeur menante à utiliser comme consigne de régulation en cours. La consigne de rapport peut être appliquée comme une entrée plus grande ou bien comme une fraction sur l'entrée menante.

Il existe plusieurs applications qui adoptent cette technique à l'industrie comme l'application des fours à gaz, les traitements des eaux, les incinérateurs de déchets, les industries chimiques, les mélangeurs ... etc.

La figure 2.24 est une application typique du contrôle d'un mélangeur où pour réaliser correctement le mélange de deux produits, l'entrée 1 et l'entrée 2 doivent être maintenues dans un rapport constant.

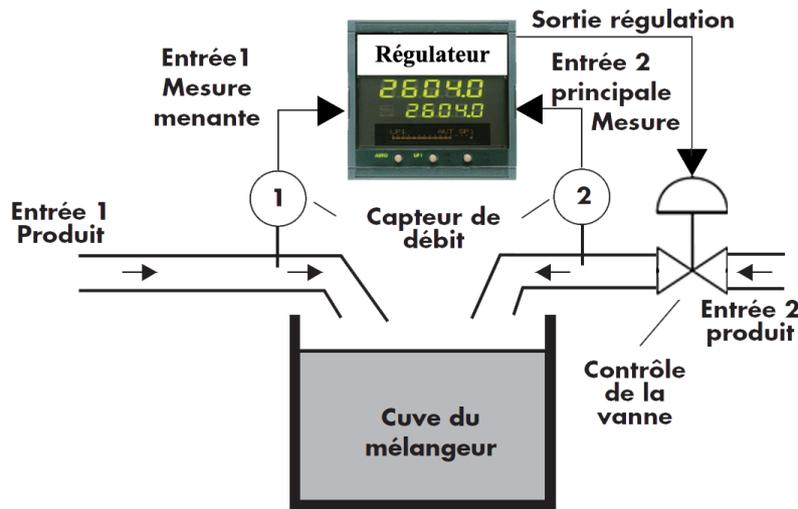


Figure 2.24 : Régulation de rapport

### 2.7.5 régulation split range [12]

Il existe de nombreuses applications de contrôle de processus dans l'industrie où il est souhaitable que plusieurs vannes de contrôle répondent à la sortie d'un contrôleur commun. Les vannes de régulation configurées pour suivre la commande du même contrôleur sont dites split range ou séquencées. Les vannes de contrôle à échelle partagée peuvent prendre différentes formes de séquençage.

Différents modes de séquençage des vannes de contrôle sont couramment observés dans l'industrie : complémentaires, exclusifs et progressifs.

#### 2.7.5.1 Séquençage de vanne complémentaire

Le premier type de séquençage de vanne de commande est un mode dans lequel deux vannes servent à doser un mélange de deux courants de fluide, comme dans cet exemple où les liquides de base et les pigments sont mélangés pour former une peinture colorée :

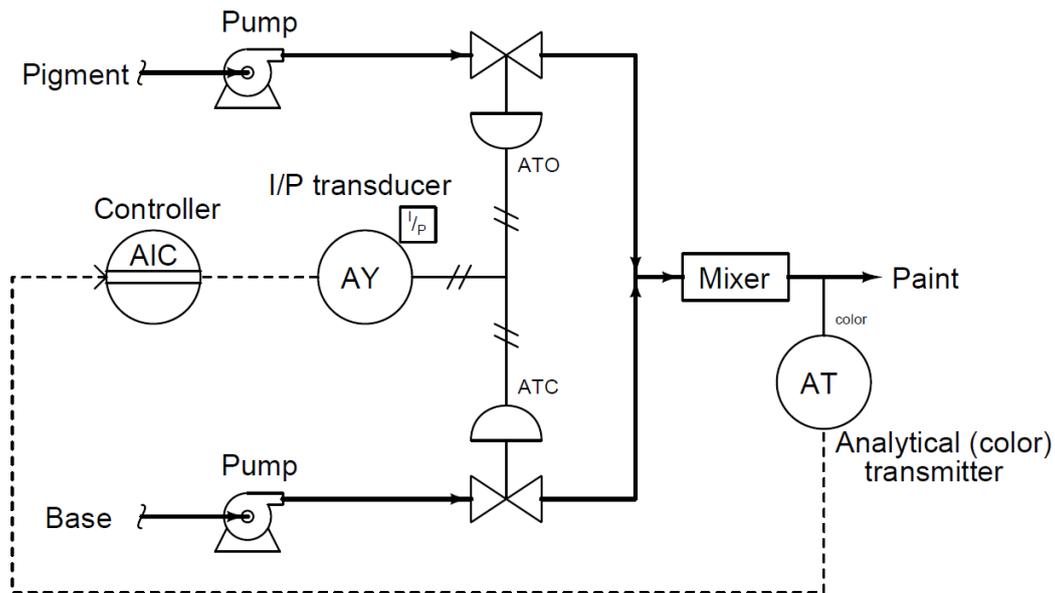


Figure 2.25 : Mélange de deux liquides pour former une peinture colorée [12]

Les vannes à base et à pigments fonctionnent à partir du même signal pneumatique de 3 à 15 PSI fourni par le transducteur I/P (AY), mais l'une des vannes est Air-To-Open tandis que l'autre est Air-To-Close.

Le tableau suivant montre la relation entre l'ouverture de la vanne pour chaque vanne de régulation et la sortie du contrôleur :

Tableau 2.2: la relation entre l'ouverture de chaque vanne et la sortie du contrôleur [12]

Sortie du contrôleur (%)	I/P sortie (PSI)	Vanne à pigment (Position de la tige)	Vanne à base (Position de la tige)
0	3	Complètement fermer	Complètement ouverte
25	6	25% ouverte	75% ouverte
50	9	50% ouverte	50% ouverte
75	12	75% ouverte	25% ouverte
100	15	Complètement ouverte	Complètement fermer

Une expression alternative pour ce comportement de vanne à split range est une représentation graphique montrant chaque ouverture de vanne sous la forme d'une bande colorée de largeur variable (plus large représentant une ouverture complet). Pour ce mode particulier de split-range, le graphique ressemblerait à ceci :

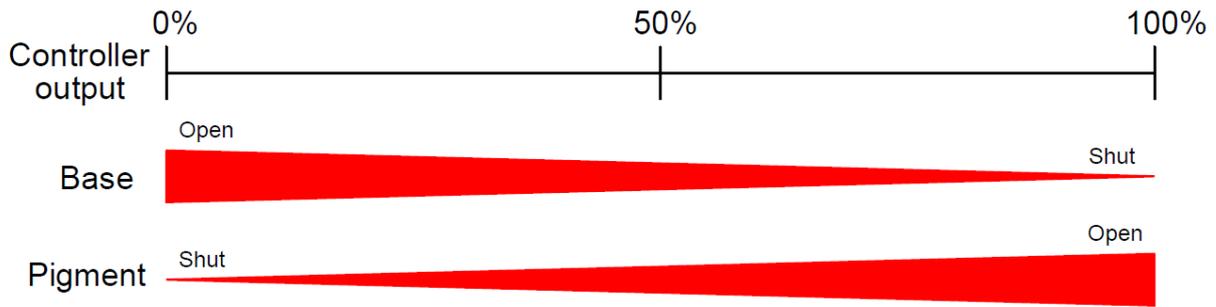


Figure 2.26 : Représentation de chaque ouverture de vanne [12]

2.7.5.2 Séquençage de vanne exclusive

D'autres applications pour les vannes de commande à split range requièrent une forme de séquençage des vannes où les deux vannes sont complètement fermées à un signal de contrôleur de sortie est 50%, une ouverture complète d'une vanne si la sortie de signale de contrôleur à 100% et une ouverture complète de l'autre vanne si le signal de sortie du contrôleur passe à 0%. La nature de ce séquençage de vanne est d'avoir un "ou-ou" étranglé chemin pour fluide de processus. Si un fluide circule à travers une vanne ou l'autre fluide qui circule à travers l'autre vanne, mais jamais travers par les deux en même temps.

Un exemple pratique de cette forme de scission est l'alimentation en réactif pour un procédé de neutralisation du pH, où la valeur du pH du liquide de traitement est ramenée au neutre par l'addition de liquides réactifs acides ou caustiques :

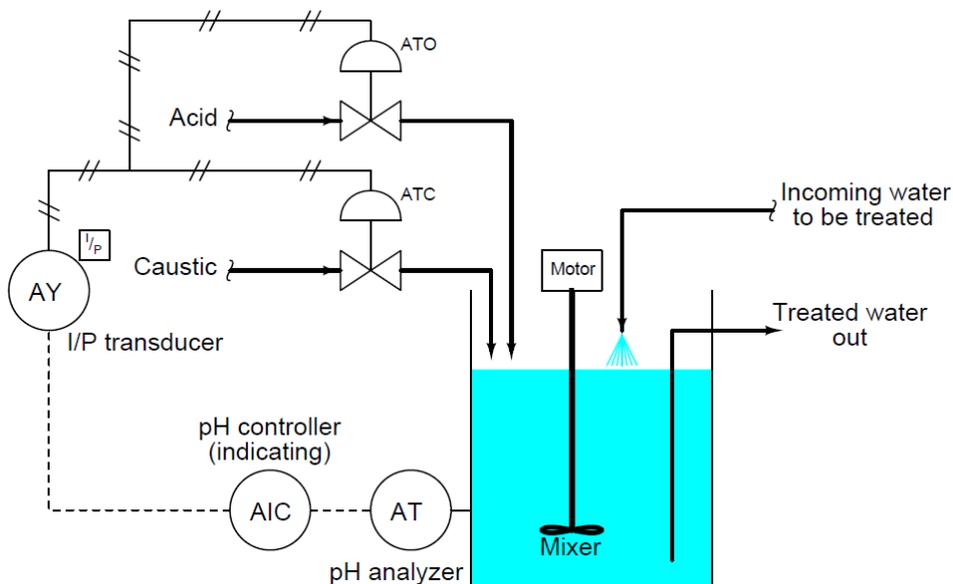


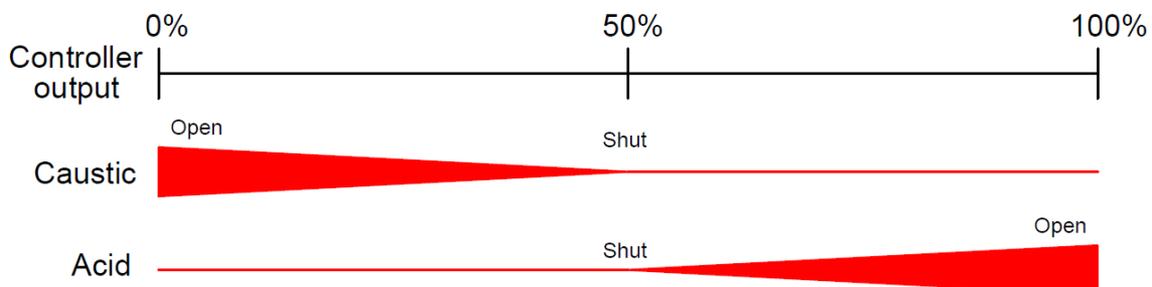
Figure 2.27 : Traitement des eaux par l'addition d'acides ou de caustiques [12]

Ici, un analyseur de pH surveille la valeur du pH de la solution liquide et un seul contrôleur de pH commande deux vannes de réactif pour ouvrir si nécessaire. Si le pH du processus commence à augmenter, le signal de sortie du régulateur augmente également (action directe) pour ouvrir la vanne d'acide. L'addition d'acide au mélange aura pour effet d'abaisser le pH du mélange. Inversement, si le pH du processus commence à diminuer, le signal de sortie du contrôleur diminuera également, fermant la vanne d'acide puis ouvrant la vanne de caustique. L'addition de caustique au mélange aura pour effet d'augmenter la valeur du pH du mélange. Les deux vannes de contrôle de réactif fonctionnent à partir du même signal pneumatique de 3 à 15 PSI fourni par le transducteur I/P (AY), mais les plages calibrées des deux vannes ne sont pas les mêmes. La vanne d'acide Air-To-Open a une plage de fonctionnement de 9 à 15 PSI, tandis que la vanne caustique Air-To-Close a une plage de fonctionnement de 9 à 15 PSI. Le tableau suivant montre la relation entre l'ouverture de la vanne pour chaque vanne de régulation et la sortie du contrôleur :

**Tableau 2.3 :** la relation entre l'ouverture de chaque vanne et la sortie du contrôleur

Sortie du contrôleur (%)	I/P sortie (PSI)	Vanne d'acide (Position de la tige)	Vanne de caustique (Position de la tige)
0	3	Complètement fermer	Complètement ouverte
25	6	Complètement fermer	50% ouverte
50	9	Complètement fermer	Complètement fermer
75	12	50% ouverte	Complètement fermer
100	15	Complètement ouverte	Complètement fermer

Encore, nous pouvons exprimer la relation exclusive des deux vannes sous la forme d'un graphique, avec des bandes colorées représentant l'ouverture de la vanne :



**Figure 2.28 :** Représentation de chaque ouverture de vanne

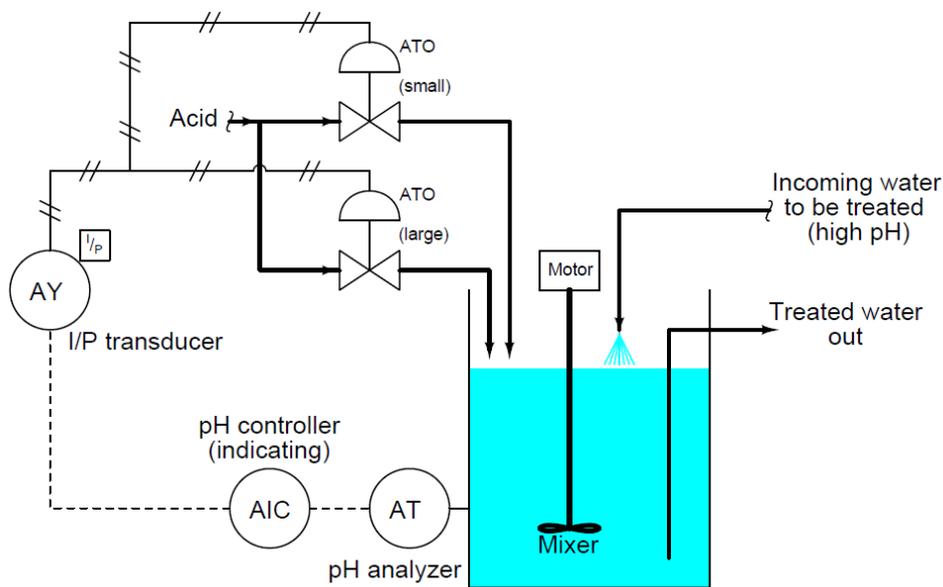
Les vannes de régulation à séquence exclusive sont utilisées dans des applications où il serait souhaitable d'avoir les deux vannes ouvertes simultanément. Dans l'exemple d'un procédé de

neutralisation du pH, le but est de contrôler l'acidité du réactif causal pour "pousser" la valeur du pH ou la direction selon les besoins. Cependant, ce serait un processus de gaspillage, car il serait simplement neutralisé sans aucun bénéfice pour le processus de liquidité lui-même.

**2.7.5.3 Séquençage de vanne progressive**

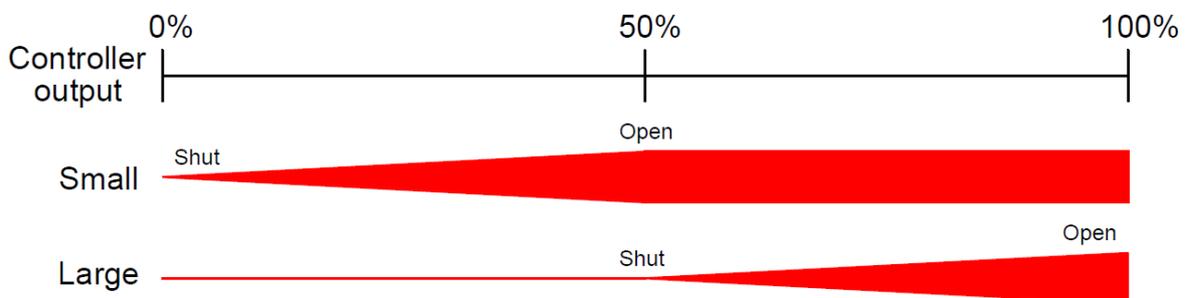
Une troisième forme de séquençage de vanne de commande est utilisée pour élargir la plage de fonctionnement du contrôle de débit pour un certain fluide au-delà de ce qu'une vanne de commande unique pourrait rassembler. Une fois de plus, le contrôle du pH fournit un exemple approprié pour illustrer une application de cette forme de séquençage.

En peut appliquer cette solution à un processus de contrôle du pH où le liquide entrant a toujours un pH élevé et doit être neutralisé avec de l'acide :



**Figure 2.29 :** Traitement des eaux ph élevé est ramenée au neutre par l'addition d'acides

Le séquençage correct des petites et grandes vannes de régulation d'acide est indiqué dans le graphique et le tableau :



**Figure 2.30 :** Représentation de chaque ouverture de vanne

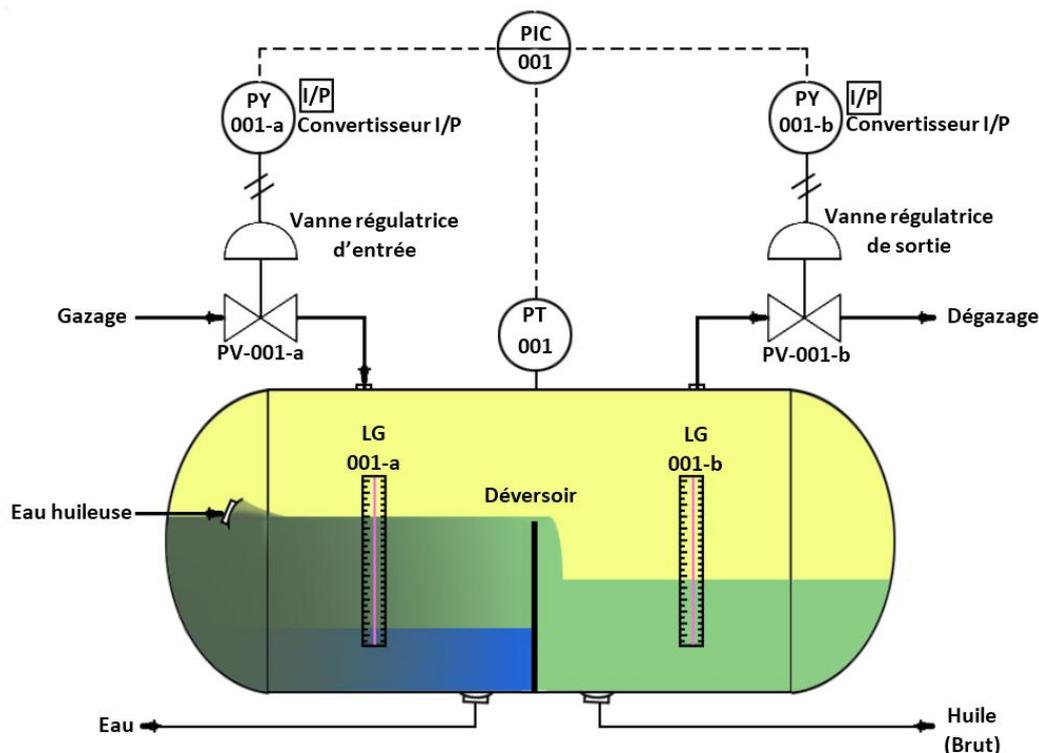
**Tableau 2.4 :** la relation entre l'ouverture de chaque vanne et la sortie du contrôleur

Sortie du contrôleur (%)	I/P sortie (PSI)	Petite vanne d'acide (Position de la tige)	Grande vanne d'acide (Position de la tige)
0	3	Complètement fermer	Complètement fermer
25	6	50% ouverte	Complètement fermer
50	9	Complètement ouverte	Complètement fermer
75	12	Complètement ouverte	50% ouverte
100	15	Complètement ouverte	Complètement ouverte

**2.7.5.4 Application split-range sur notre dégazeur**

Cette application est un cas particulier au mode de séquençage de vanne exclusive, les vanne de commande requièrent une forme de séquençage des vannes où les deux vannes sont complètement fermées à un signal de contrôleur entre 60% et 65% nommé bande morte.

Une ouverture complète de la vanne régulatrice d'entrée si la sortie de signal de contrôleur à 100% et une ouverture complète de la vanne régulatrice de sortie si le signal de sortie du contrôleur passe à 0%.



**Figure 2.31 :** Séparateur d'eau huileuse et dégazeur par deux vannes régulatrices

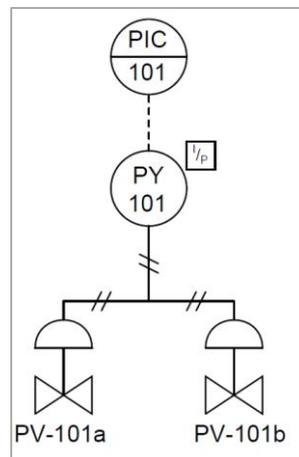
Dans les premiers exemples de séquençage de vannes de régulation précédents, les deux vannes de régulation ont reçu le même signal pneumatique d'un convertisseur commun I/P

(courant-pression). Cela signifie que chaque vanne a reçu exactement le même signal de pression du transducteur pour toute valeur de sortie du contrôleur donnée. Le séquençage des deux vannes (c'est-à-dire faire en sorte que chacune répond différemment au même signal de pression d'air) consistait donc à régler chaque vanne sur une plage de pression de banc différente.

Il faut cependant comprendre que la mise en place de deux vannes de régulation avec différentes plages de banc d'essai n'est pas la seule façon de diviser une gamme de vannes. D'autres voies existent également, chacune avec ses propres avantages et inconvénients.

Les illustrations suivantes illustrent plusieurs méthodes alternatives pour le séquençage des vannes de régulation (split range) :

- L'approche du signal pneumatique commun (un contrôleur, un transducteur I/P) est simple mais souffre de l'inconvénient de la réponse lente, car un transducteur I/P doit commander deux actionneurs pneumatiques. Le temps de réponse peut être amélioré en ajoutant un surpresseur pneumatique entre le transducteur I/P et les actionneurs de vanne, ou en ajoutant un positionneur à au moins une des vannes. Chacune de ces solutions fonctionne par le même principe : réduire la demande de volume d'air sur le transducteur I/P commun.

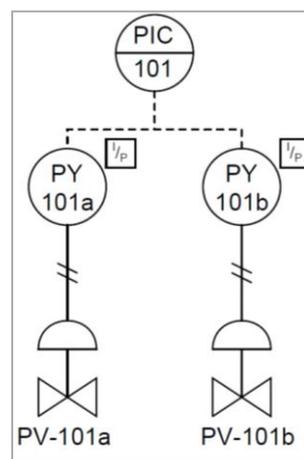


**Figure 2.32 :** Le signal pneumatique commun

- Le câblage de deux transducteurs I/P en série afin qu'ils partagent un courant de signal commun est une autre façon de diviser deux vannes de régulation. Cette approche ne souffre pas d'une réponse lente, car chaque vanne possède son propre transducteur I/P dédié pour l'alimenter en air d'actionnement. Nous avons maintenant un choix où nous implémentons les split ranges : nous pouvons le faire dans les transducteurs I/P (i.e. chaque transducteur I/P ayant un étalonnage différent) ou dans les vannes de contrôle (i.e. chaque vanne de contrôle ayant une plage de pression différente). Comme il est

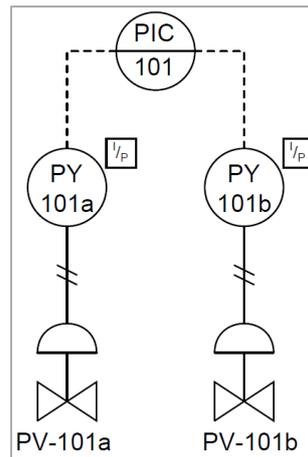
généralement plus facile de modifier la plage d'un I/P que de reconstruire une vanne de régulation avec un ressort différent (pour lui donner une plage de pression d'actionnement différente), cette approche présente l'avantage d'une configuration pratique.

Un inconvénient de la stratégie I/P câblée en série est la charge supplémentaire imposée aux circuits du signal de sortie du contrôleur : il faut veiller à ce que les deux convertisseurs I/P connectés en série ne tombent pas trop à pleine intensité, sinon le contrôleur peut avoir de la difficulté à piloter les deux appareils en série. Un autre inconvénient (potentiel) des dispositifs de vanne connectés en série dans une boucle de courant est l'incapacité d'installer des instruments « intelligents » communiquant avec le protocole HART, car plusieurs périphériques sur la même boucle rencontreront des conflits d'adresse. Les appareils HART ne peuvent fonctionner en mode hybride analogique / numérique que s'il y a un appareil par circuit 4-20 mA.



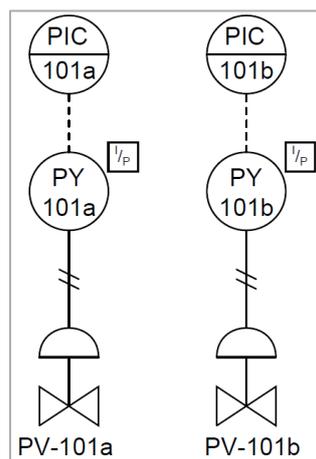
**Figure 2.33** : Un courant de signal commun

- Un moyen populaire d'implémenter la split-range est d'utiliser plusieurs sorties 4-20 mA sur le même contrôleur. Ceci est très facile à faire si le contrôleur fait partie d'un grand système (par exemple un DCS ou un PLC) avec plusieurs canaux de sortie analogiques. Si plusieurs sorties sont configurées sur un contrôleur, chaque vanne aura sa propre paire de fils dédiée pour le contrôle. Cela tend à simplifier le câblage par rapport aux transducteurs ou positionneurs I/P câblés en série, car chaque boucle de vanne est un circuit 4-20 mA standard, comme tout autre circuit de boucle de régulation (non séparé). C'est peut-être aussi la manière la plus pratique de mettre en œuvre un split-range lorsque des positionneurs de vanne "intelligents" sont utilisés, car les circuits de boucle dédiés permettent un fonctionnement normal du protocole HART sans conflits d'adresses.



**Figure 2.34 :** plusieurs sorties sont configurées sur un contrôleur

- Un avantage des sorties à deux contrôleurs est la possibilité d'effectuer le séquençage à split range dans le contrôleur lui-même, ce qui est souvent plus facile que de modifier un transducteur I/P ou de calibrer un positionneur de vanne. Ainsi, les signaux 4-20 mA vont chaque vanne sera unique pour toute valeur de sortie de contrôleur donnée. Si elles sont séquençées en tant que telles, les valeurs d'étalonnage du transducteur I/P et de banc de vannes peuvent être standard plutôt que personnalisées. Bien sûr, le fait que le contrôleur soit capable d'effectuer le séquençage nécessaire ne signifie pas que le séquençage doit être effectué dans le contrôleur. Il est possible de programmer les deux sorties analogiques du contrôleur pour envoyer exactement le même signal de courant à chaque vanne, en configurant chaque vanne (ou chaque positionneur, ou chaque transducteur I/P) pour répondre différemment aux signaux de courant identiques.



**Figure 2.35 :** le séquençage à split range par deux contrôleurs

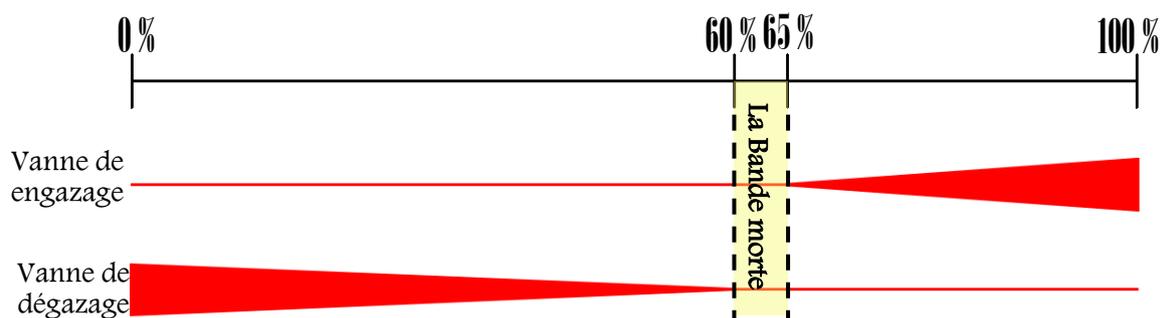
On à utiliser dans cette application un contrôleur (Automate Programmable Industriel) qui contient plusieurs sorties analogique, chaque vanne aura sa propre paire de fils dédiée pour le contrôle. Cela tend à simplifier le câblage par rapport aux transducteurs ou positionneurs I/P

(convertisseur I/P) câblés en série, car chaque boucle de vanne est un circuit 4-20 mA standard, comme tout autre circuit de boucle de régulation (non séparé). Le tableau suivant montre la relation entre l'ouverture de chaque vanne, la sortie de contrôleur et la sortie des deux transducteurs I/P :

**Tableau 2.5 :** la relation entre l'ouverture des vannes, la sortie du contrôleur et la sortie des deux transducteurs I/P

Sortie du contrôleur (%)	I/P 001-a (PSI)	Vanne d'entrée PV-001-a (Position de la tige)	I/P 001-b (PSI)	Vanne de sortie PV-001-b (Position de la tige)
0	3	Complètement fermer	15	Complètement ouverte
15	3	Complètement fermer	12	75% ouverte
30	3	Complètement fermer	9	50% ouverte
45	3	Complètement fermer	6	25% ouverte
60	3	Complètement fermer	3	Complètement fermer
65	3	Complètement fermer	3	Complètement fermer
73,75	6	25% ouverte	3	Complètement fermer
82,5	9	50% ouverte	3	Complètement fermer
91,25	12	75% ouverte	3	Complètement fermer
100	15	Complètement ouverte	3	Complètement fermer

Le graphique suivant exprime l'ouverture de chaque vanne régulatrice par des bandes colorées et présente leur relation avec la sortie du contrôleur (API).

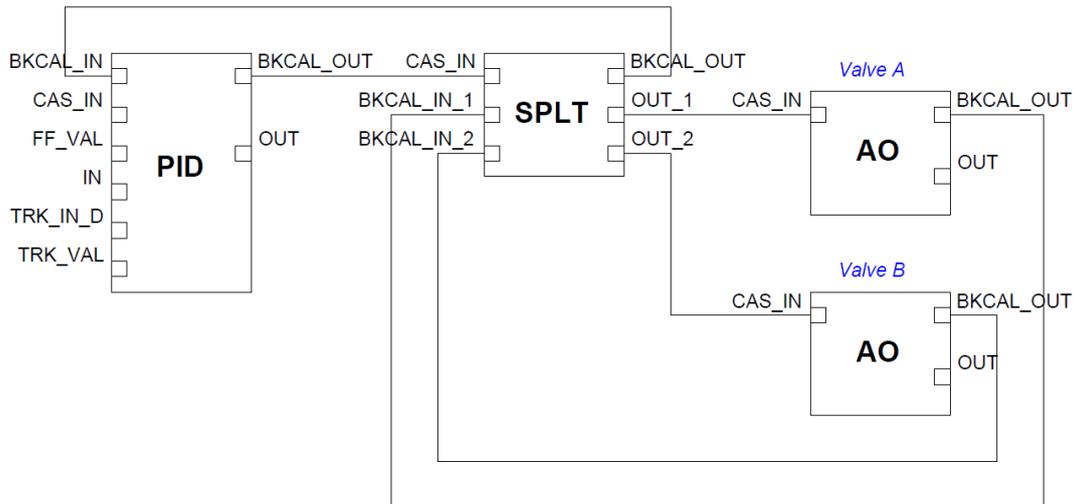


**Figure 2.36 :** Représentation l'ouverture des vannes et la bande morte

- Une adaptation numérique de la méthode de séquençage du contrôleur à deux sorties est observée dans les systèmes fondation bus de terrain, où un bloc fonctionnel logiciel spécial appelé « SPLT » ou « splitter » existe pour fournir un séquençement séparé à deux vannes. Le bloc fonctionnel "SPLT" reçoit un signal de commande unique et émet

deux signaux, un signal de sortie pour chaque vanne dans une paire à intervalle divisé.

Le diagramme fonctionnel d'un tel système apparaît ici :



**Figure 2.37** : Le splitter reçoit un signal de commande unique et émet deux signaux

Dans ce système bus de terrain, un seul bloc de contrôle PID émet un signal vers le bloc SPLT, qui est programmé pour commander deux signaux de positionnement uniques aux blocs AO (sortie analogique) des deux vannes. Il convient de noter que, bien que chaque bloc AO soit propre à sa propre vanne de commande, les blocs SPLT et même PID peuvent être situés dans n'importe quel dispositif capable dans le réseau de bus de terrain. Avec fondation bus de terrain, les fonctions du système de contrôle ne sont pas nécessairement reléguées à des appareils séparés. Il est possible, par exemple, d'avoir une vanne de régulation équipée d'un positionneur de bus de terrain effectuant ses propres calculs de contrôle PID et séquençage à split range en localisant ces blocs fonctionnels dans cet appareil physique.

## Conclusion

Dans ce chapitre nous avons présenté la relation entre les deux vannes régulatrices du dégazeur (la vanne d'engazage et la vanne de dégazage) et le signal de sortie du contrôleur dans une boucle split range.

Dans le chapitre suivant, on s'intéressera aux rôles essentiels de l'automate programmable industriel (API), la programmation, les calculs des coefficients du régulateur PID, Ainsi la partie de la supervision.

# Chapitre 3

Automate Programmable industriel

## Introduction

Face à l'évolution dans le monde industriel, l'automatisation devient nécessaire. Elle remplace l'action de l'être humain dans des tâches pénibles, répétitives et dangereuses. Quand il n'y a pas d'automate, on utilise de nombreux relais pour commander les différents équipements intervenant dans certains procédés industriels. L'ajout d'un automate, en raison de son logiciel, permet de réduire considérablement le nombre de relais et d'accroître la fiabilité des procédés industriels. Un autre avantage des automates programmables est qu'en facilitant la modification de leur programmation logique et de leurs paramètres, ils favorisent une amélioration en continu des procédés.

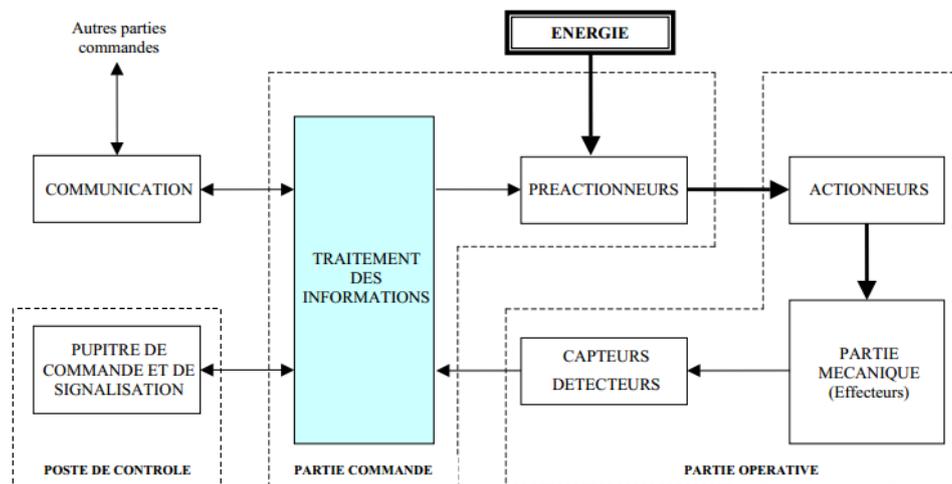
Un très grand nombre d'industries utilisent des automates programmables pour augmenter leur productivité et la qualité des produits.

Le présent chapitre donne un rappel indispensable sur la commande séquentielle et présente le nouveau système qui est basé sur l'automate S7-300 CPU 313C ainsi que sa programmation sur le SIMATIC Manager et la supervision avec le progiciel WinCC flexible 2008.

### 3.1 Structure des systèmes automatisés [13]

Chaque système automatisé comporte deux parties :

- Une Partie Opérative (PO) dont les actionneurs agissent sur le processus automatisé.
- Une Partie Commande (PC) qui coordonne les actions de la Partie Opérative.



**Figure 3.1 :** Structure d'un système automatisé [13]

#### ➤ Partie Opérative

C'est elle qui opère sur la matière d'œuvre et le produit. Elle comporte en général :

- Des outillages et moyens divers mettant en œuvre le processus d'élaboration, par exemple, moules, poinçons, outils de coupe, pompes, têtes de soudure, de marquage.

- Des actionneurs destinés à mouvoir ou mettre en œuvre ces moyens, exemple :  
moteur électrique pour actionner une pompe, vérin hydraulique pour fermer un moule, vérin pneumatique pour mouvoir une tête de marquage.
- Partie Commande
- C'est elle qui émet des ordres vers la Partie Opérative et en reçoit les signaux en retour, afin de coordonner ses actions, au centre de la Partie Commande, le "traitement" est la convergence de 3 dialogues qu'il coordonne :
- A) Le dialogue avec la machine
- Commande des actionneurs (moteurs, vérins, vanne) via les pré-actionneurs (contacteur, distributeurs, variateur...).
- Acquisition des signaux en retour par les capteurs qui rendent compte de révolution de la machine.
- B) Le dialogue homme-machine
- Pour exploiter, régler, dépanner la machine, le personnel émet des consignes et reçoit des informations en retour.
- C) Le dialogue avec d'autres machines
- Plusieurs machines peuvent coopérer dans une même production Leur coordination est assurée par le dialogue entre leurs Parties Commandes.

### 3.2 Les systèmes de commande [13]

Pour réaliser un système de commande, nous disposons à l'heure actuelle de deux moyens :

- les solutions câblées.
- les solutions programmées.

#### 3.2.1 Les solutions câblées

Dans les solutions câblées, la mise en œuvre nécessite uniquement l'établissement de liaisons matérielles (câblage) selon un schéma établi à partir de la théorie ou de l'expérience. En électricité ou en électronique, les liaisons sont assurées par des câbles électriques, tandis qu'en fluide, il s'agit de canalisations reliant les différents composants de l'installation.

Elles comportent des contraintes :

- le poids et le volume des composants (non négligeable)
- la rentabilité financière
- la complexité de l'installation
- des risques d'erreurs de câblage

- la recherche de pannes

Toute modification dans le choix du fonctionnement de l'installation entraîne :

- une intervention dans le câblage (main-d'œuvre)
- une augmentation du nombre de relais et de fils (dimensions de l'armoire à remettre en cause)
- un coût élevé (main-d'œuvre, relais, fils, armoire plus grande si nécessaire)
- le dialogue avec de telles installations est très limité (voyants, synoptique).

Avantage des solutions câblées :

- technologie simple, connue et maîtrisée pour les installations simples.
- réalisation, mise en service et maintenance de ces installations assurées par un personnel ne nécessitant pas une formation spécifique.

### 3.2.2 Les solutions programmées (utilisation d'un automate programmable)

Elles éliminent tous les relais auxiliaires, les relais temporisés, les pendules et toute la filerie assurant les liaisons entre ces différents éléments. L'encombrement se trouve réduit et la recherche de panne est facilitée.

Inconvénients :

- utilisation d'un personnel formé à cette technologie.
- le coût à la réalisation reste élevé.

Avantage :

- encombrement de l'armoire électrique réduit lorsque la complexité de l'installation augmente.
- main-d'œuvre réduite lors du câblage.
- terminal de programmation pouvant être commun à plusieurs automates.
- modifications possibles sans interventions sur le câblage (à partir d'un terminal de programmation).
- Performances techniques (fréquence de traitement, type de signaux traités ... etc.)

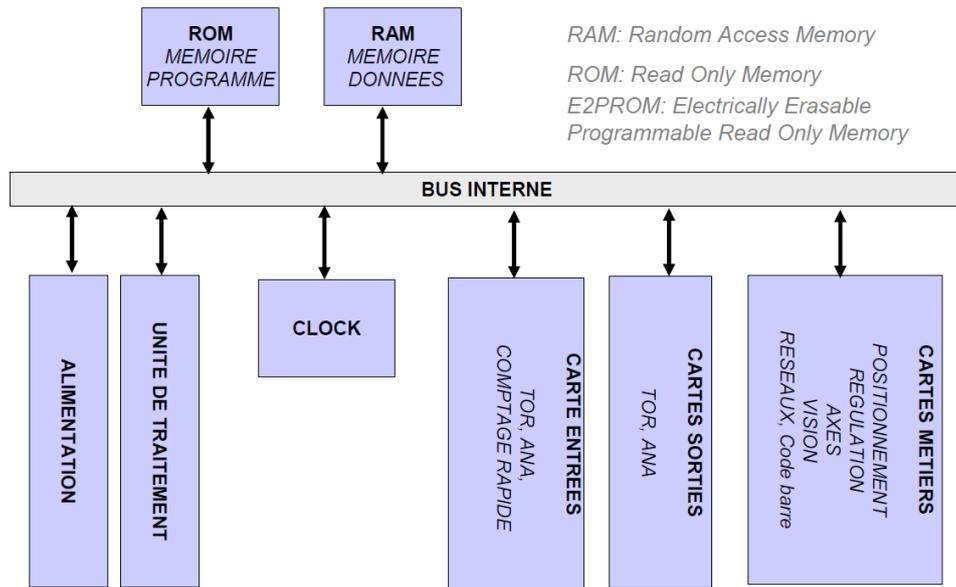
## 3.3 L'automate programmable industriel [13]

### 3.3.1 Définition

Un automate programmable industriel est un dispositif électronique programmable similaire à un ordinateur destiné à la conduite et la surveillance en temps réel de processus industriels par un traitement séquentiel, à l'aide d'un langage adapté. Il envoie des ordres vers les pré-actionneurs (partie opérative côté actionneur) à partir de données d'entrées (capteurs) (partie commande côté capteur)

### 3.3.2 Architecture des automates programmables industriels

La structure d'un API peut se représenter comme suit :



**Figure 3.2 :** Structure interne d'un API [14]

#### 3.3.2.1 Le processeur

Le processeur, ou unité centrale (UC), a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement de l'application (les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul PID, etc.). Mais en dehors de cette tâche de base, il réalise également d'autres fonctions :

- Gestion des entrées/sorties.
- Surveillance et diagnostic de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement.
- Dialogue avec le terminal de programmation, aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications des données.

Un ou plusieurs processeurs exécutent ces fonctions grâce à un micro logiciel préprogrammé dans une mémoire de commande, ou mémoire système. Cette mémoire morte définit les fonctionnalités de l'automate. Elle n'est pas accessible à l'utilisateur.

#### 3.3.2.2 La mémoire

La zone mémoire va permettre de recevoir les informations issues des capteurs d'entrées, de recevoir les informations générées par le processeur et destinées à la commande des sorties (valeur des compteurs, des temporisations, ...etc.) et de recevoir et conserver le programme du processus.

Deux familles de mémoires sont utilisées dans les automates programmables :

A) Les mémoires vives

Mémoires à accès aléatoire « Random Access Memory (RAM) » Le contenu de ces mémoires peut être lu et modifié à volonté, mais il est perdu en cas de manque de tension (mémoire volatiles). Elles nécessitent par conséquent une sauvegarde par batterie. Les mémoires vives sont utilisées pour l'écriture et la mise au point du programme, et pour le stockage des données.

B) Les mémoires mortes

Elles sont à lecture seule, les informations ne sont pas perdues lors de la coupure de l'alimentation des circuits. On peut citer les types suivants :

- ROM « Read Only Memory » : Elle est programmée par le constructeur et son programme ne peut être modifié.
- PROM « Programmable ROM » : Elle est livrée non enregistrée par le fabricant. Lorsque celle-ci est programmée, on ne peut pas l'effacer
- EPROM « Erasable PROM » : C'est une mémoire PROM effaçable par un rayonnement ultraviolet intense.
- EEPROM « Electrically EPROM » : C'est une mémoire PROM programmable plusieurs fois et effaçable électriquement.
- Mémoire Flash : C'est une mémoire EEPROM rapide en programmation. L'utilisateur peut effacer un bloc de cases ou toute la mémoire.

La mémoire morte est destinée à la mémorisation du programme après la phase de mise au point. La mémoire programme est contenue dans une ou plusieurs cartouches qui viennent s'insérer sur le module processeur ou sur un module d'extension mémoire.

### 3.3.2.3 Interface Entrées-Sorties

Des cartes d'entrées - sorties (en anglais Input - Output, I/O) numériques (tout ou rien) pour des signaux à 2 états ou analogiques pour des signaux à évolution continue

- Cartes d'entrées pour brancher des capteurs, boutons poussoirs, etc.
- Cartes de sorties pour brancher des actionneurs, voyants, vannes, etc.

Les cartes d'E/S ont une modularité de 8,16 ou 32 voies. Elles admettent ou délivrent des tensions continues 0-24 Vcc.

### 3.3.2.4 Les bus

Ils sont composés d'un certain nombre de conducteurs (pistes en cuivre) et destinés pour le transport des informations binaires (suite de 0 ou 1), c'est-à-dire (0V ou 5V) sur chaque fil.

L'unité centrale dispose de trois bus :

- Bus de donnée : Il permet de véhiculer des données du microprocesseur vers un composant ou d'un composant vers le microprocesseur. Il est donc bidirectionnel.
- Bus d'adresse : La mémoire est composée de nombreuses cases mémoires. Chaque case est repérée par une adresse. Le bus adresses est unidirectionnel (du microprocesseur vers les autres composants).
- Bus de commande : Il assure la commande des boîtiers mémoires et entrées-sorties par le microprocesseur.

### 3.3.2.5 Le module d'alimentation

Permet de fournir à l'automate l'énergie nécessaire à son fonctionnement. Ils délivrent, à partir du 220 V alternatif, des sources de tension nécessaires à l'automate tels que : +5V, 12V et 24V en continu.

Afin d'assurer le niveau de sûreté requis, elle comporte des dispositifs de détection de baisse ou de coupure de la tension réseau, et de surveillance des tensions internes. En cas de défaut, ces dispositifs peuvent lancer une procédure prioritaire de sauvegarde.

### 3.3.3 Principe de fonctionnement d'un API [13]

Le cycle de fonctionnement de l'API est décrit ci-dessous (Figure 3.3).

Traitement interne : L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).

Lecture des entrées : L'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.

Exécution du programme : L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.

Ecriture des sorties : L'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

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

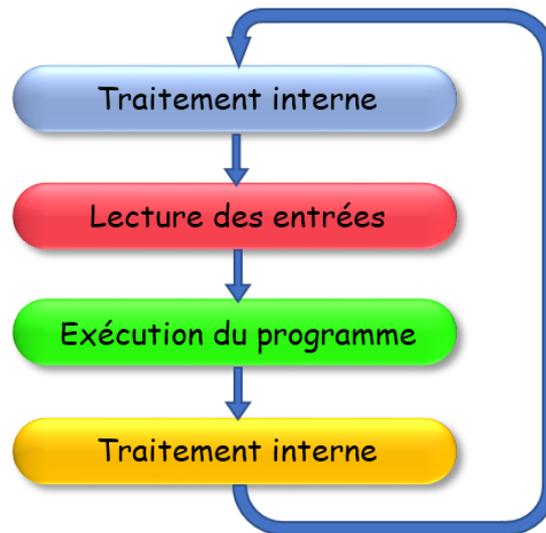


Figure 3.3 : Fonctionnement d'un API [13]

### 3.3.4 Les langages de programmation des API

Les langages de programmation utilisés pour les API ont évolué depuis l'introduction des automates à la fin des années 60. Actuellement la norme CEI 61131-3 (la partie 3 sur 8 de la norme notée précédemment 1131, apparue en 1993 et en deuxième édition en 2003 et spécifiant les langages de programmation), spécifie 5 langages se résumant en trois langages graphiques et deux littéraux :

#### 3.3.4.1 Langages graphiques

##### ➤ Langage LD (Ladder Diagram)

Langage graphique développé pour les électriciens. Il utilise les symboles tels que : Contacts, relais et blocs fonctionnels et s'organise en réseaux (labels). C'est le plus utilisé.

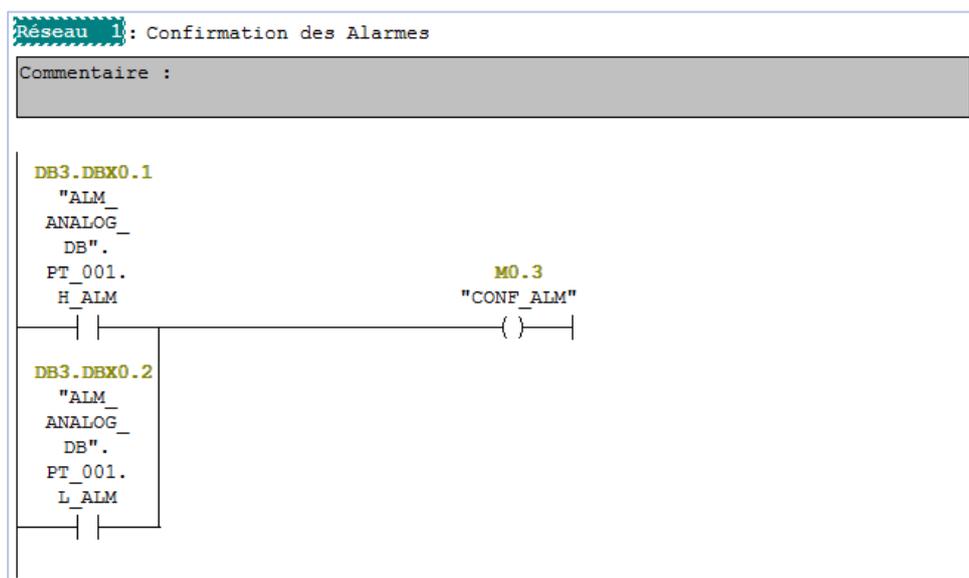


Figure 3.4 : Exemple d'un langage à contacts

➤ **Blocs Fonctionnels (FBD : Function Bloc Diagram)**

Langage graphique où des fonctions sont représentées par des rectangles avec les entrées à gauche et les sorties à droites. Les blocs sont programmés (bibliothèque) ou programmables. Utilisé par les automaticiens.

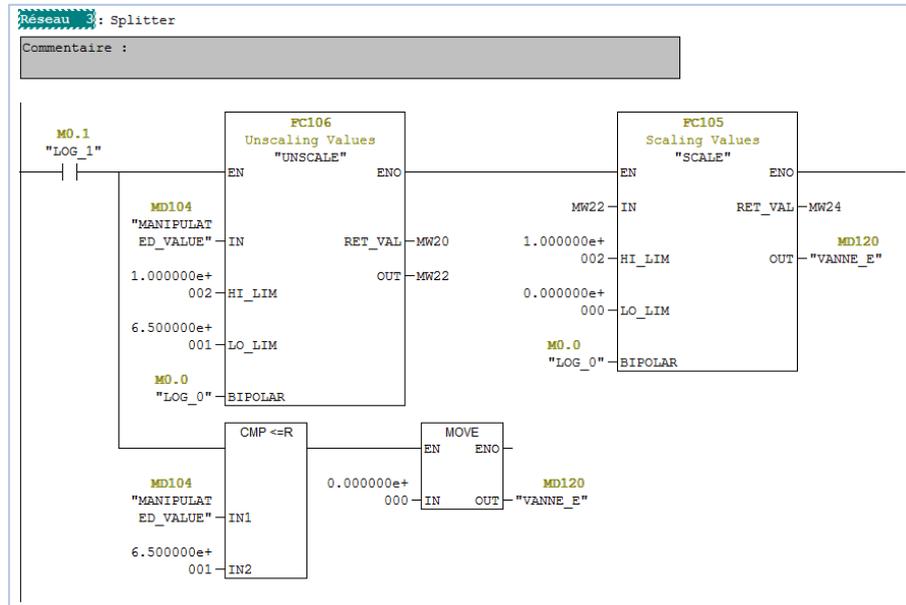


Figure 3.5 : Exemple d'un langage FBD

➤ **Le langage SFC (Sequential Function Chart)**

Le langage SFC (Sequential Function Chart), ou GRAFCET, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD.

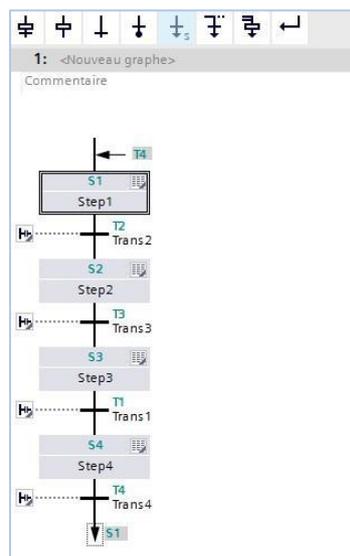


Figure 3.6 : Exemple d'un langage SFC



### 3.3.5 Comparaison des langages [14]

Le tableau suivant présente avantages et inconvénients de différents langages de programmation des APIs.

**Tableau 3.1 : Comparaison des langages**

LANGAGE	AVANTAGES	INCONVENIENTS
LD	Facile à lire et à comprendre par la majorité des électriciens langage de base de tout PLC	Suppose une programmation bien structurée
FBD	Très visuel et facile à lire	Peut devenir très lourd lorsque les équations se compliquent
ST	Langage de haut niveau (langage pascal) Pour faire de l'algorithmique	Pas toujours disponible dans les ateliers logiciels
IL	Langage de base de tout PLC type assembleur	Très lourd et difficile à suivre si le programme est complexe Pas visuel.
SFC	Description du fonctionnement (séquentiel) de l'automatisme. Gestion des modes de marches Pas toujours accepté dans l'industrie...	Peu flexible

### 3.4 Critères de choix d'un automate [13]

Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe et les contacts commerciaux et expériences vécues sont déjà un point de départ. Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir "se retourner" en cas de "perte de vitesse" de l'une d'entre elles.

Le personnel de maintenance doit toutefois être formé sur ces matériels et une trop grande diversité des matériels peut avoir de graves répercussions. Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions.

La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables.

Il faut ensuite quantifier les besoins :

- 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.
- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- Fonctions ou modules spéciaux : certaines cartes (commande d'axe, pesage ...) permettront de "soulager" le processeur et devront offrir les caractéristiques souhaitées (résolution, ...).
- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ...).
- La fiabilité et la robustesse.
- La durée de garantie et le service après-vente.

### 3.5 L'automate Siemens S7-300

Le système d'automatisation S7-300 de Siemens est un automate modulaire fabriqué par la famille SIMATIC (Figure 3.9). Tous ces éléments logiques, temporisations, compteurs, ...etc. nécessaires à l'automatisation sont prévus par le fabricant et sont intégrés dans l'automate. C'est un automate qui offre des performances et ne nécessite pas d'entretien. Il peut supporter jusqu'à 512 E/S tout ou rien (TOR) et 64 E/S analogique, comme il peut être configuré avec un maximum de 32 modules de signaux pouvant être répartis sur un châssis de base et trois châssis d'extensions.

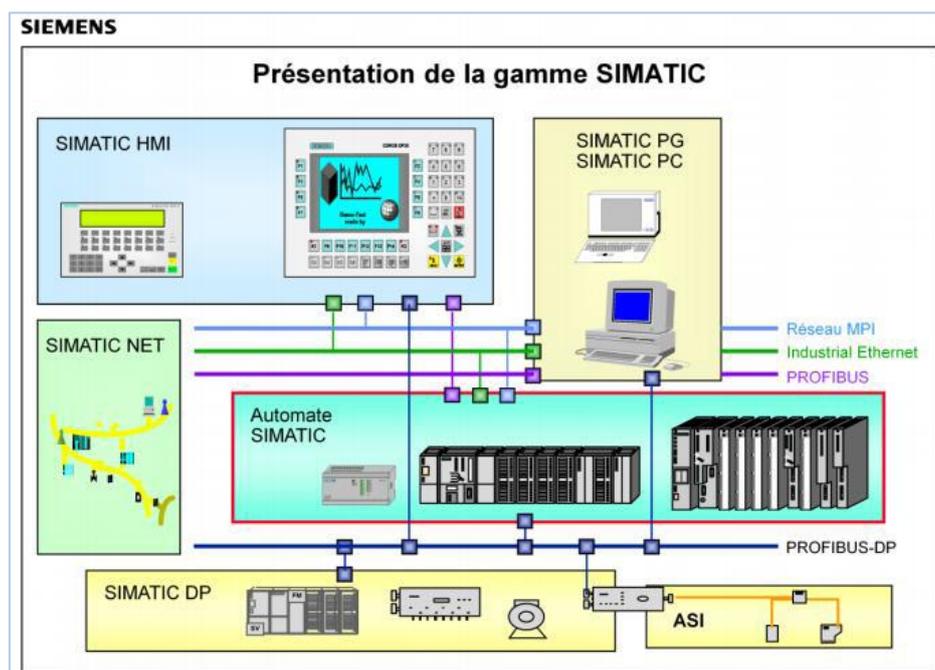


Figure 3.9 : Présentation de la gamme SIMATIC [13]

### 3.5.1 Caractéristiques de l'API S7-300 [13]

Mini-automate modulaire pour les applications d'entrée et de milieu de gamme.

- Gamme diversifiée de CPU.
- Gamme complète de modules.
- Possibilité d'extension jusqu'à 32 modules.
- Bus de fond de panier intégré aux modules.
- Possibilité de mise en réseau avec : l'interface multipoint (MPI), PROFIBUS ou Industriel Ethernet.
- Raccordement central de la PG avec accès à tous les modules.
- Liberté de montage aux différents emplacements.
- Configuration et paramétrage à l'aide de l'outil « configuration matérielle ».

### 3.5.2 Constitution

Dans l'automate programmable S7-300, le profilé support (châssis) contient 11 emplacement au maximum ou la gamme des modules est fixée dans l'ordre suivant (Figure 3.10)

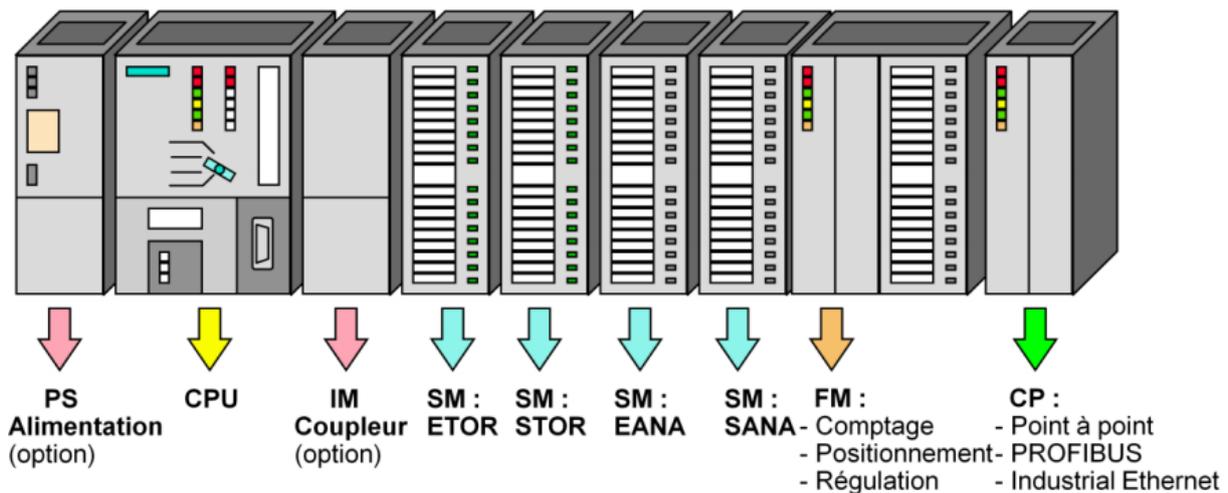


Figure 3.10 : Modules de l'API S7-300 [13]

#### a) Module d'alimentation (PS)

Il convertit la tension secteur (AC120/230 V) en une tension d'alimentation (24 VCC) pour les modules de l'automate. Les tensions pour les capteurs, actionneurs, ... qui dépassent le 24 V sont fournies par des blocs d'alimentation ou transformateurs supplémentaires.

**b) Unité centrale (CPU)**

L'unité centrale a pour fonction la lecture des états des signaux d'entrées et la commande des sorties, l'exécution de programme d'utilisateur et permettre aussi le réglage du comportement au démarrage et le diagnostic des défauts par les LEDS.

Elle est disponible en différentes versions dont certains :

- A utilisateur standard : CPU 313, CPU 314, CPU 315, CPU 316.
- Avec fonctions intégrées : CPU 312 IFM, CPU 314 IFM.
- Avec interfaces PROFIBUS-DP : CPU 315-DP, CPU 316-2 DP, CPU 318-2 DP.

CPU 31xC sont des CPU avec une série des entrées et des sortie intégrées.

La CPU 313C utilisée dans notre étude est caractérisée par :

- Raccordements des entrées et sorties intégrées (DI24/DO16 et AI5/AO2).
- Une mémoire de travail RAM 32 KO.
- Un logement pour une cartouche mémoire EPROM, EEPROM (MMC max 8Mbyte).
- Temps de traitement 0.1 ms/kilo-instructions.
- Un connecteur pour console de programmation, terminal de dialogue (MPI).
- 3 sorties impulsions (2,5kHz).
- Comptage et mesure sur 3 voies avec codeurs incrémentaux 24V (30kHz).
- Configuration multi-rangées jusqu'à 31 modules.
- Atelier logiciel de programmation et simulation SIMATIC Manager V5.5+SP2.
- Interface homme-machine (Supervision) SIMATIC WinCC flexible SP3 2008.

Le module CPU 313C comporte les éléments suivants (Figure 3.11) et les caractéristiques sont données dans le :

❖ Les LEDS pour la signalisation d'état et de défauts

SF : signalisation groupée de défauts, erreurs dans le programme ou défauts sur un module de diagnostic.

DC5V : signalisation de la tension d'alimentation interne 5V.

FRCE : forçage, signalisation qu'au moins une entrée ou une sortie est forcée de manière permanente.

RUN : Clignotement à la mise en route de la CPU, allumage continu en mode RUN.

STOP : Allumage continu en mode STOP, clignotement lent lorsqu'un effacement général est requis et clignotement rapide lorsqu'un effacement général est en cours.

❖ Un commutateur de mode de fonctionnement

MRES = Effacement général (Module Reset).

STOP = Arrêt, le programme n'est pas exécuté.

RUN = Le programme est exécuté, accès en lecture seule avec une PG.

❖ Connexion MPI

Le port interface MPI se présente sous la forme d'une connexion à neuf ponts, protégée par un cache. Elle permet de réaliser des liaisons multipoints entre plusieurs appareils S7.

❖ Logement pour la carte mémoire

Une carte mémoire peut être à cet emplacement. Elle conserve le contenu du programme en cas de coupure de courant, même en l'absence de pile.

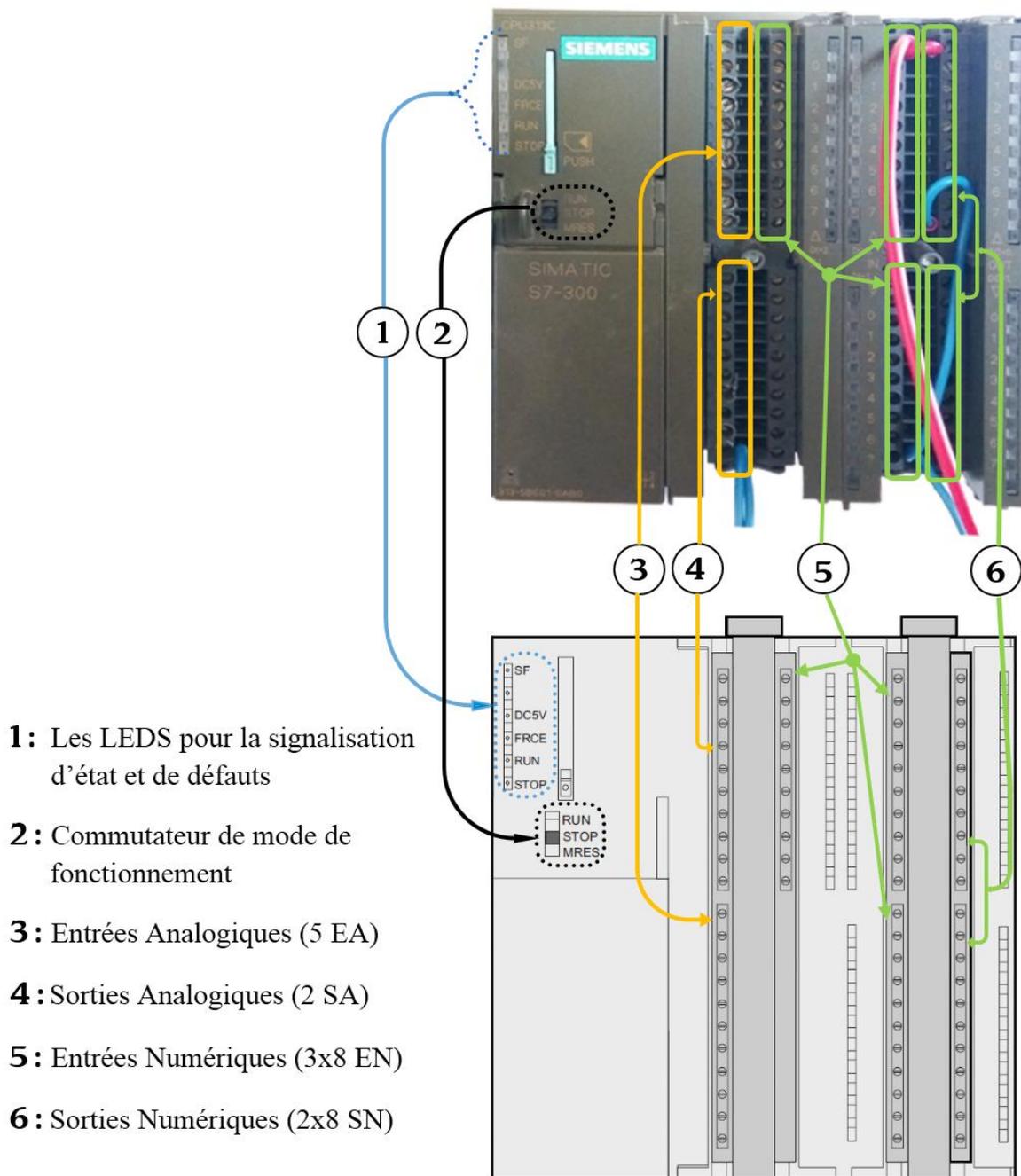


Figure 3.11 : Présentation de la CPU 313C

**c) Coupleurs (IM) :**

Les coupleurs IM360/IM361 ou IM365 permettent de réaliser des configurations à plusieurs châssis, ces modules d'interface servent donc à interconnecter le châssis de base ZG au châssis d'extension EG dans une configuration à plusieurs rangées.

**d) Modules de signaux (SM) :**

ETOR = Modules d'entrées tout ou rien permet le raccordement de l'API aux capteurs logiques.

STOR = Modules de sorties tout ou rien permettent le raccordement de l'API aux pré-actionneurs, chaque sortie est identifiée par une LED.

EANA = Modules d'entrées analogiques convertissent les signaux analogiques en valeurs numériques traitées par la CPU en utilisant des CAN.

SANA = Modules de sorties analogiques fournissent des signaux analogiques à partir des valeurs numériques transmises par la CPU en utilisant des CNA.

**3.6 Logiciel de programmation SIMATIC Manager STEP7 V5.5****3.6.1 Présentation du logiciel**

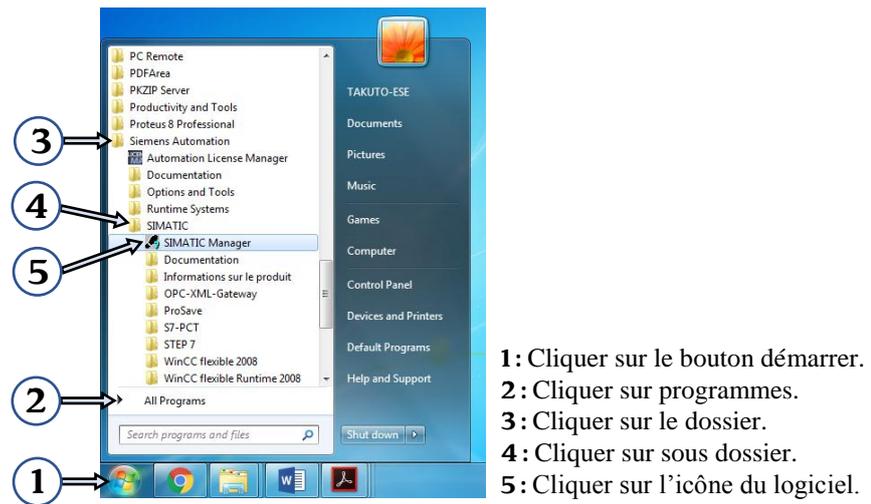
L'atelier logiciel pour automates SIMATIC offre un environnement d'ingénierie intégré, avec des outils performants adaptés aux méthodes de travail et aux tâches les plus diverses. Sont des outils de configuration et de programmation des contrôleurs, SIMATIC disponibles sur une base système homogène, offrent des interfaces ouvertes, génèrent des blocs réutilisables et font gagner du temps d'ingénierie.

STEP 7 est un progiciel d'ingénierie complet permettant d'exécuter toutes les opérations d'ingénierie d'un projet. Les utilisateurs bénéficient ainsi d'un gain de productivité tout en réduisant les coûts d'ingénierie.

**3.6.2 Démarrer le SIMATIC STEP7 V5.5**

Pour démarrer logiciel SIMATIC Manager STEP 7, procédez de la manière suivante :

1. Sélectionnez sous Windows la commande "Démarrer > Programmes > Siemens Automation > SIMATIC > SIMATIC Manager" (Figure 3.12).



**Figure 3.12 :** Lancement de SIMATIC Manager en utilisant le menu Démarrer

2. On peut également utiliser le raccourci présent sur le bureau (Figure 3.13).



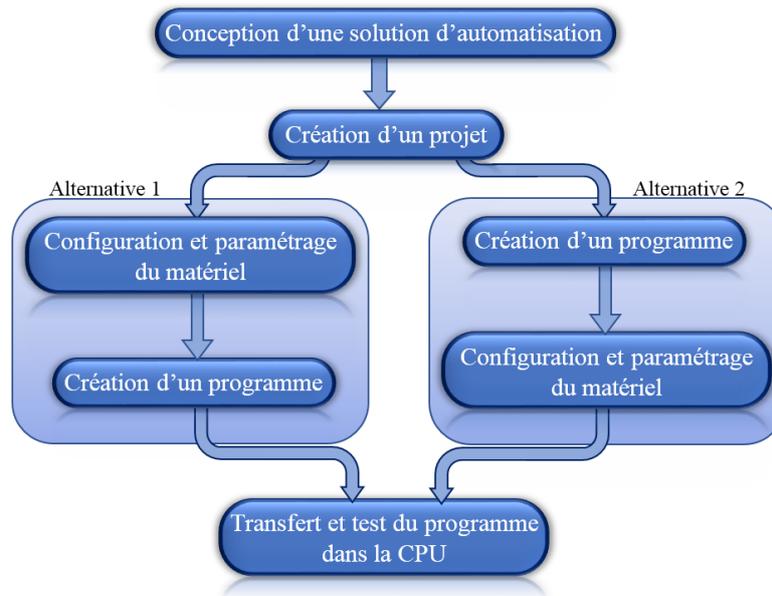
**Figure 3.13 :** Lancement de SIMATIC Manager en utilisant le raccourci du bureau

### 3.6.3 La conception d'un programme avec SIMATIC Manager STEP7 V5.5

La stratégie à suivre pour faire la conception d'un programme en utilisant la plate-forme SIMATIC Manager est :

- La création d'un nouveau projet.
- La configuration matérielle.
- Compilation et chargement de la configuration.
- La création de la table des mnémoniques.
- L'élaboration du programme.
- La simulation avec le logiciel.
- La visualisation d'état du programme (le test).

La conception d'une solution d'automatisation se fait par deux alternatives, soit on commence par la programmation ou par la configuration matérielle, dans notre cas on a commencé par la configuration.

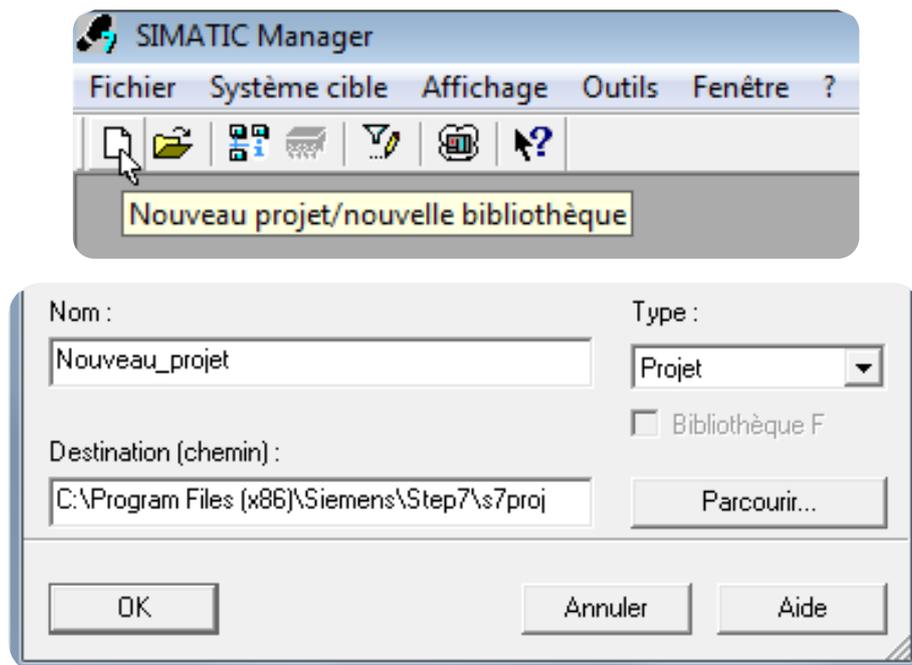


**Figure 3.14 :** Organisation pour la création de projets sous SIMATIC Manager STEP7

### 3.6.3.1 Création d'un nouveau projet

Pour créer un projet dans SIMATIC Manager STEP7, on procède de la manière suivante :

- 1) Appuyer sur l'icône « SIMATIC Manager » pour ouvrir le logiciel STEP7, en peut faire cette étape avec deux méthodes (3.6.2 Démarrer le SIMATIC STEP7 V5.5).
- 2) Cliquer sur l'icône « Nouveau projet ».
- 3) Choisir un nom et le chemin souhaités pour le projet ou utiliser les données proposées.
- 4) Valider par le clique sur le bouton « OK » (Figure 3.15).
- 5) Enfin le nouveau projet est créé et affiché dans le navigateur du projet.

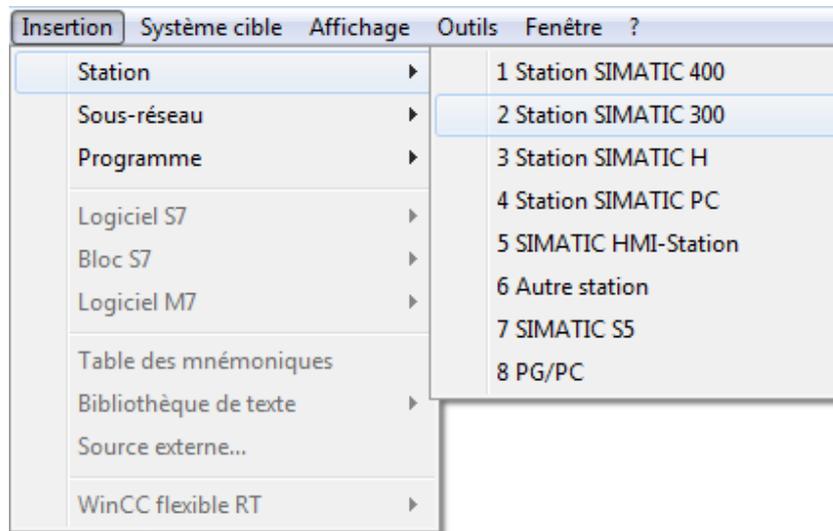


**Figure 3.15 :** Création d'un nouveau projet.

### 3.6.3.2 Configuration matériels

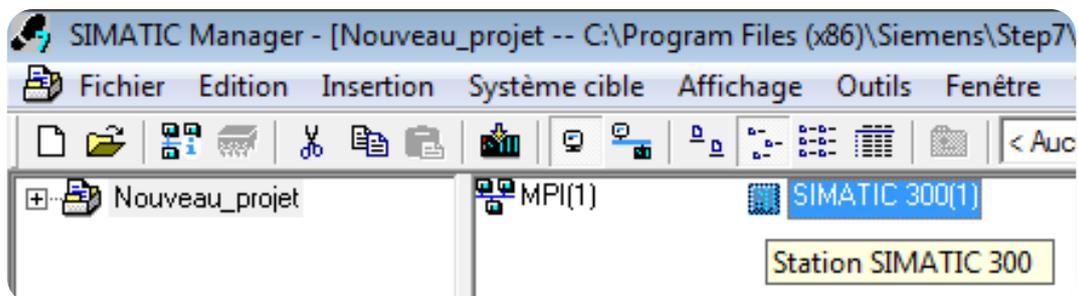
Une fois notre projet crée, on peut faire la configuration avec les étapes suivante :

- 1) Insérer une station « Station SIMATIC 300 » (Figure 3.16).



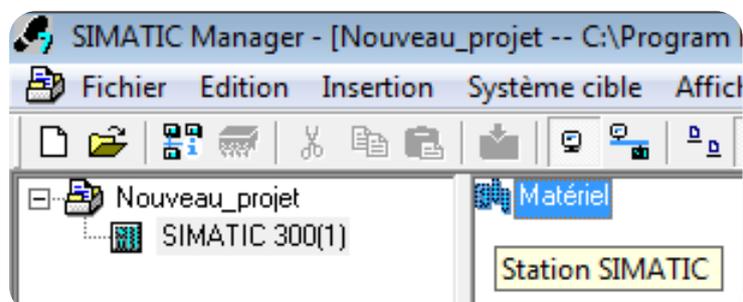
**Figure 3.16 :** Insertion Station SIMATIC 300

- 2) La « station » apparaît dans la partie droite de l'écran, double-clique sur cette icône (Figure 3.17).



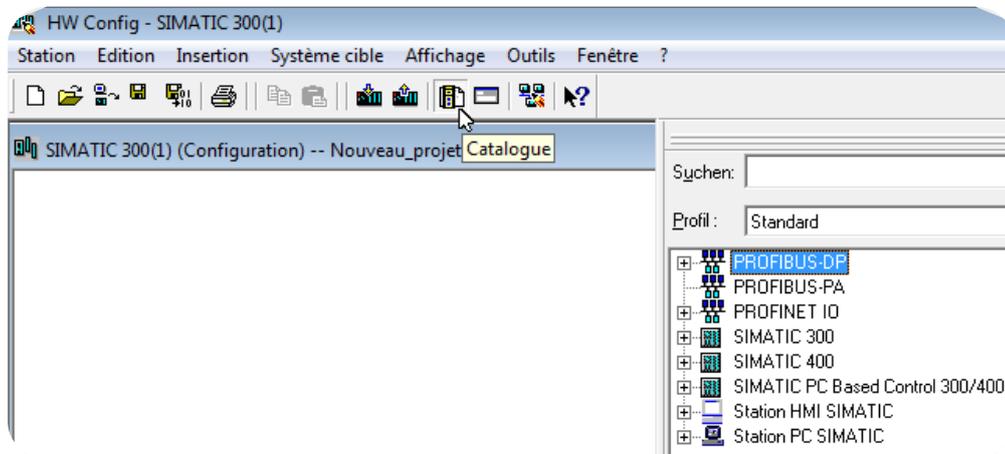
**Figure 3.17 :** Icône « Station SIMATIC 300 »

- 3) Afin d'effectuer la configuration matérielle, double-clique sur l'icône matériel dans la partie droite de l'écran (Figure 3.18).



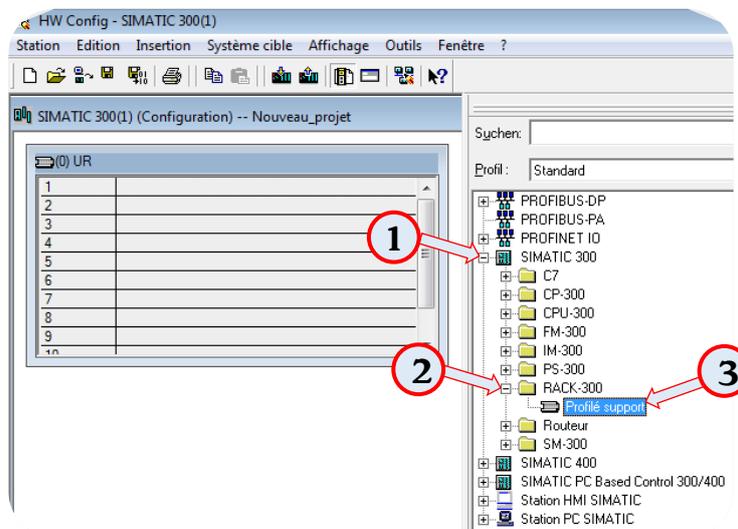
**Figure 3.18 :** Icône « matériel »

- 4) La nouvelle fenêtre qui s'ouvre va nous permettre de configurer l'automate. Si la liste des « stations » (sur la droite) n'apparaît pas, clique sur l'icône « catalogue » (ou menu « Affichage », puis cocher « catalogue ») (Figure 3.19).



**Figure 3.19 :** La liste des « stations »

- 5) Dans la liste des « stations » développer « SIMATIC 300 » (cliquer sur le [+] devant SIMATIC 300), puis développer « RACK-300 » et double-cliquer sur « Profilé support ». Celui-ci apparaît dans la partie gauche de l'écran (Figure 3.20).



**Figure 3.20 :** Profilé support du RACK-300

- 6) Développer « PS-300 » et double-cliquer sur « PS 307 5A ». Le module alimentation de 5A se place sur le premier emplacement du RACK.

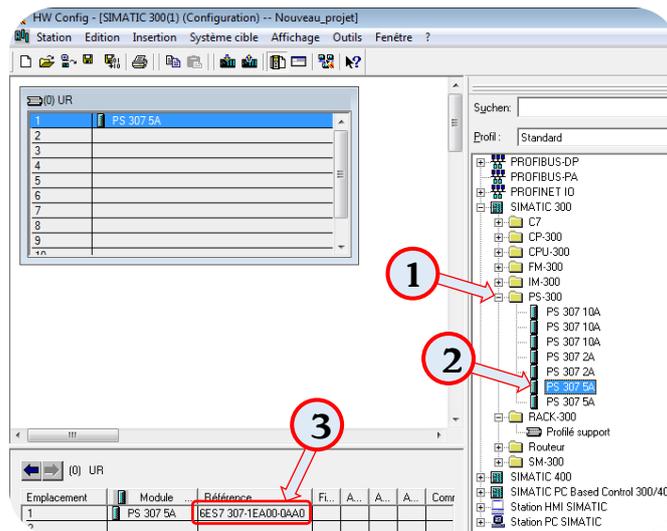


Figure 3.21 : Placement du module alimentation de 5A

Le module alimentation est porté sur la « PS 307 5A »

Tableau 3.2 : Caractéristique technique du module alimentation

Numéro de référence	6ES7 1EA00-0AA0
Tension d'entrée	120/230 VCA
Tension de sortie	24 VCC
Courant de sortie	5 A

7) Développer « CPU-300 », puis « CPU 313C » et double-cliquer sur « 6ES7 313-5BE01-0AB0 ». Le module Unité de traitement avec les modules d'entrées et de sorties et le module de comptage intégrés se place sur le deuxième emplacement du RACK.

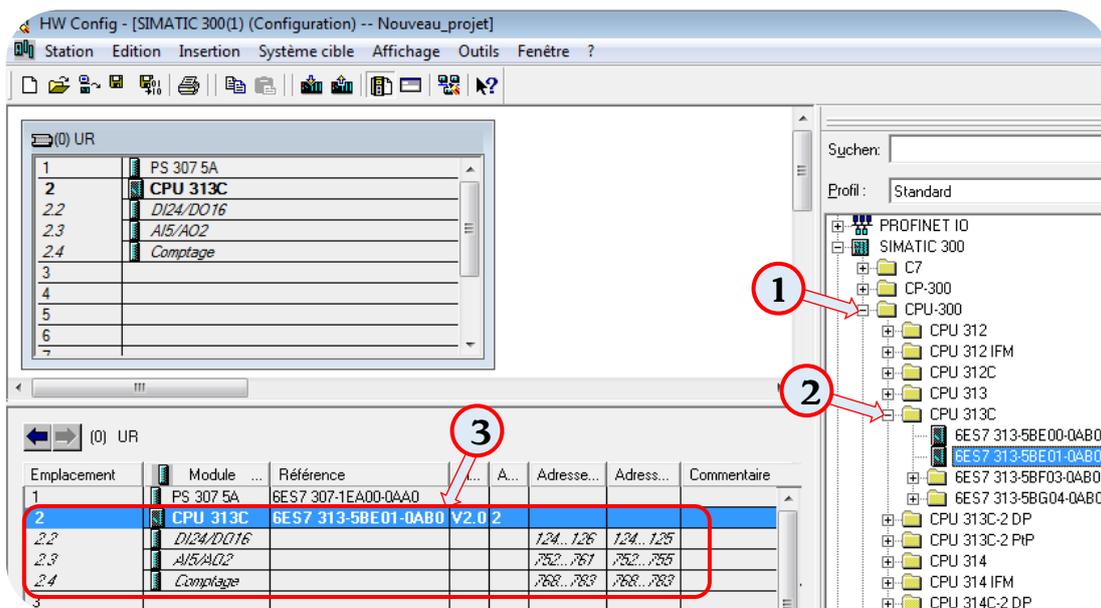


Figure 3.22 : Choix de la CPU

- 8) Enregistrement les modifications. Avec cette étape la configuration matérielle est terminée.

On a choisi la CPU 313C, ayant pour référence 6ES7 313-5BE01-0AB0 à base des critères suivants :

- Nombre d'entrées-sorties tout ou rien intégrer.
- Nombre d'entrées-sorties analogique intégrer.
- Mémoire de travail
- L'extensibilité de la CPU

**Tableau 3.3 :** Caractéristique technique de la CPU 313C

Numéro de référence	6ES7 313-5BE01-0AB0
Nombre d'entrées-sorties tout ou rien intégrer	24 entrées / 16 sorties
Nombre d'entrées-sorties analogique intégrer	5 entrées / 2 sorties
Mémoire de travail	32 Ko
Comptage et mesure sur 3 voies	Sur 3 voies
Codeurs incrémentaux	24V (30kHz)
Configuration multi rangée	Jusqu'à 31 modules
Version de firmware	V2.0
Temps de surveillance de cycle	150 ms
Tension d'alimentation	24 VCC
La vitesse de traitement des données	0,1ms/kilo-instructions

### 3.6.3.3 Compilation et chargement de la configuration [15]

Une fois la configuration matérielle est réalisée, il faut compiler et charger dans l'automate. La compilation matérielle et logicielle se fait en sélectionnant l'API dans le projet puis en cliquant sur l'icône « compiler » de la barre de tâche.

Avant la compilation et le chargement de la configuration matérielle, il faut choisir le type de connexion entre l'automate et le computer (PG/PC), le câblage approprié et faire la configuration d'interface PG/PC avec STEP7.



**Figure 3.23 :** Les différents câbles utilisés

### 3.6.3.4 La création de la table des mnémoniques

Afin de faciliter la programmation, il est intéressant de créer une table de variables. C'est par laquelle que l'on va pouvoir déclarer toutes les variables et les constantes utilisées. Lorsque l'on définit une variable API, il suffit d'indiquer :

- Son nom : C'est l'adressage symbolique de la variable.
- Son type de donnée : BOOL (1 bit), Word (8bits), ...
- Son adresse absolue : Indication d'opérande (Par exemples I1.3, Q3.5, M0.0, ...).
- Un commentaire : Pour qu'il nous renseigne sur cette variable.

On édite notre table de mnémoniques en respectant notre cahier de charge de la manière suivante (Figure 3.24) :

- 1) Dans le navigateur du projet et dans la partie gauche de l'écran, développer le « nom du projet » et développer « SIMATIC 300 », puis développer « CPU 313C » et un seul-clic sur le « Programme S7(1) ». Ainsi dans la partie droite de l'écran se trouve le fichier « Mnémoniques » (Figure 3.24).

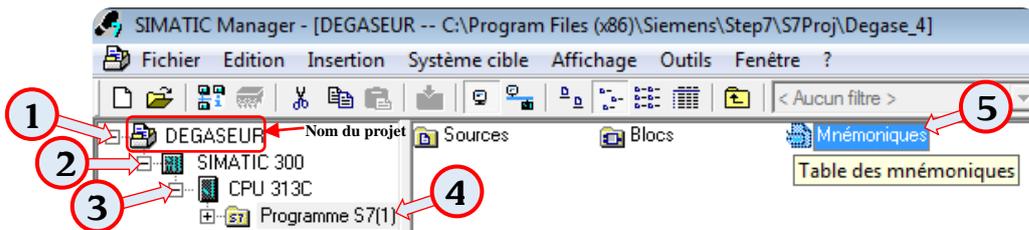


Figure 3.24 : Le fichier « Mnémoniques »

- 2) Double clic sur le fichier « Mnémoniques », on peut entrer des mnémoniques maintenant ou en cours de programmation.
- 3) On clique sur Enregistrer et on ferme l'éditeur de mnémoniques (Figure 3.25).

	Etat	Mnémonique	Opérande	Type de d	Commentaire
1		ACK	E 124.0	BOOL	
2		ACK_T	M 0.2	BOOL	
3		ALARM	FC 2	FC 2	
4		ALARM_UDT_2	UDT 2	UDT 2	
5		ALM_ANALOG_DB	DB 3	DB 3	
6		ANALOG INPUT	FB 1	FB 1	
7		BUZZER	A 124.1	BOOL	
8		CONF_ALM	M 0.3	BOOL	
9		CONT_C	FB 41	FB 41	Continuous Control

Figure 3.25 : La création de la table de mnémonique

3.6.3.5 Ecriture du programme

On écrit le programme en utilisant différent blocs :

Dans notre travail on a utilisé deux blocs d'organisation (OB1 et OB35), deux fonctions (FC), un seul bloc fonctionnel (FB), cinq blocs de données (DB), deux types de données utilisateur (UDT) et trois blocs prêts à l'emploi (FC105, FC106 et FB41) (figure 3.26), en langage Logigramme, Ladder Diagram (LD) et Function Bloc Diagram (FBD).

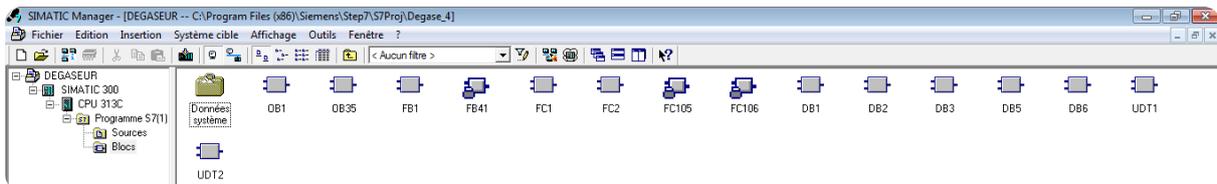


Figure 3.26 : Les blocs utiliser dans notre travail

- Bloc d'organisation OB : il commande le traitement du programme. Il est possible par l'intermédiaire des OB de réagir aux événements cycliques, temporisés ou déclenchés par alarme durant l'exécution du programme. Le programme de OB sera un appel aux différentes fonctions (bloc CALL).
  - OB1 : Le système d'exploitation de la CPU S7 exécute l'OB1 de manière cyclique, aussitôt son traitement achevé, il le démarre à nouveau. L'exécution cyclique de l'OB1 commence quand la mise en route est terminée. Le bloc OB1 a la priorité la plus basse parmi tous les OB à durée d'exécution surveillée, son traitement peut donc être interrompu par tous les autres OB, sauf par l'OB90. [16]
  - OB35 : Il y a neuf OB d'alarme cyclique (OB30 à OB38) dans STEP 7, permettant de démarrer des programmes à des intervalles de temps équidistants. Notre CPU 313C contient à l'OB35. Le (Tableau 3.4) présente les caractéristiques de ce dernier.

Tableau 3.4 : caractéristiques du bloc d'alarme cyclique OB35 [16]

OB d'alarme cyclique	Période par défaut	Classe de priorité par défaut
OB35	100 ms	12

- Bloc fonctionnel FB : est un bloc avec rémanence. Les paramètres transmis au FB ainsi que les variables statiques sont sauvegardés dans le bloc de données d'instance (DB). Les variables temporaires sont rangées dans la pile des données locales.
  - FB41 : Le bloc SFB41/FB41 "CONT\_C" (continuous controller) sert à régler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système réglé.

En peut l'utiliser le régulateur comme régulateur PID de maintien individuel, mais aussi comme régulateur en cascade, proportionnel ou de rapport dans des régulations à plusieurs boucles. Sa méthode de travail se base sur l'algorithme PID du régulateur d'échantillonnage à sortie analogique, complété le cas échéant par un niveau de formateur d'impulsions assurant la formation de sorties à impulsions modulées en durée pour régulations à deux ou trois échelons avec actionneurs proportionnels.

➤ Fonction FC : est un bloc de code sans mémoire. Les variables temporaires d'une fonction sont sauvegardées dans la pile des données locales. Ces données sont perdues à l'achèvement de la fonction. Les fonctions peuvent faire appel à des blocs de données globaux (DB) pour la sauvegarde des données. [16]

- FC105 : La fonction Mise à l'échelle (SCALE) prend une valeur entière (IN) et la convertit selon l'équation ci-après en une valeur réelle exprimée en unités physiques, comprises entre une limite inférieure (LO\_LIM) et une limite supérieure (HI\_LIM).  

$$OUT = [((FLOAT(IN) - K1) / (K2 - K1)) * (HI\_LIM - LO\_LIM)] + LO\_LIM$$
 Le résultat est écrit dans OUT (Figure 3.27).

Les constantes K1 et K2 sont définies selon que la valeur d'entrée est bipolaire ou unipolaire.

Bipolaire : La valeur entière d'entrée est supposée être comprise entre (-27648 et 27648) donc : K1 = -27648.0 et K2 = +27648.0

Unipolaire : La valeur entière d'entrée est supposée être comprise entre (0 et 27648) 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. Une mise à l'échelle inversée peut être obtenue en programmant une limite inférieure supérieure à la limite supérieure (LO\_LIM > HI\_LIM). Dans ce cas, la valeur de la sortie diminue quand la valeur de l'entrée augmente. [16]

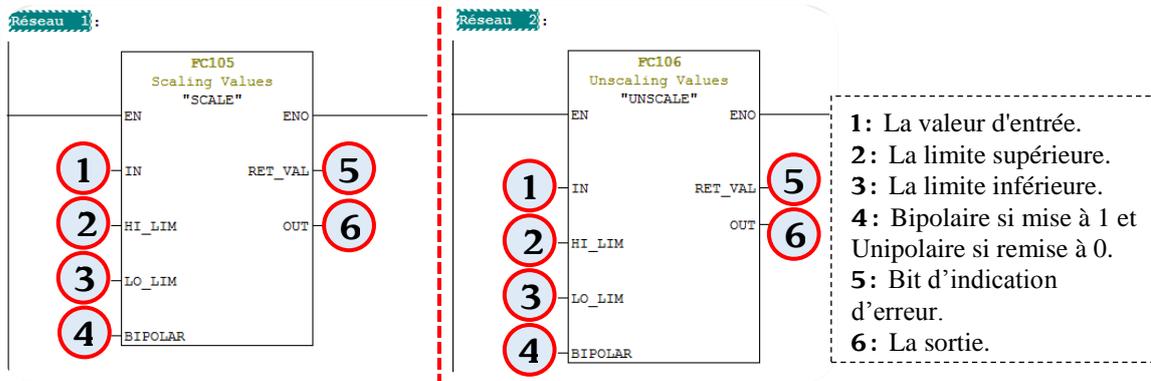
- FC106 : La fonction Annuler la mise à l'échelle (UNSCALE) prend une valeur d'entrée réelle (IN) exprimée en unités physiques comprises entre une limite inférieure (LO\_LIM) et une limite supérieure (HI\_LIM) et la convertit selon l'équation ci-après en une valeur entière.

$$OUT = [((IN - LO\_LIM) / (HI\_LIM - LO\_LIM)) * (K2 - K1)] + K1$$

Le résultat est écrit dans OUT (Figure 3.27).

La même définition du K1 et K2 dans la fonction FC105 (SCALE).

Si la valeur entière d'entrée se situe en dehors de la plage définie par les limites inférieure (LI\_LIM) et supérieure (HI\_LIM), la sortie (OUT) est saturée à la valeur la plus proche de la limite inférieure ou supérieure de la plage indiquée pour son type (bipolaire ou unipolaire) et une erreur est signalée. [16]



**Figure 3.27 :** Les fonctions FC105 et FC106 (SCALE et UNSCALE)

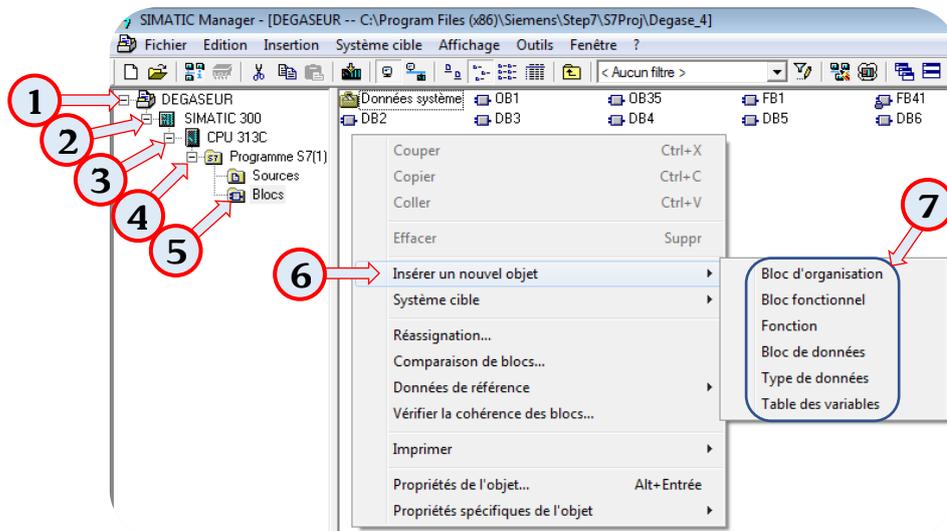
➤ Bloc de données DB :

- Bloc de données d'instance : est associé à chaque appel de bloc fonctionnel transmettant des paramètres. Ce bloc de données d'instance contient les paramètres effectifs et les données statiques du FB.
- Bloc de données globaux : Contrairement aux blocs de code, les blocs de données ne contiennent pas d'instructions STEP 7. Ils servent à l'enregistrement de données utilisateur, ils contiennent des données variables que le programme utilisateur utilise. Les blocs de données globaux servent à l'enregistrement de données utilisateur pouvant être utilisées par tous les autres blocs.

➤ Types de données utilisateur UDT : Les types de données utilisateur (user data type, UDT) sont des structures de données particulières. En peut les utiliser, une fois définis, dans l'ensemble du programme utilisateur S7. En peut utiliser les UDT comme types de données simples ou comme types de données complexes dans la déclaration des variables de blocs de code (FC, FB, OB) ou encore comme types de données pour des variables dans un bloc de données (DB). [16]

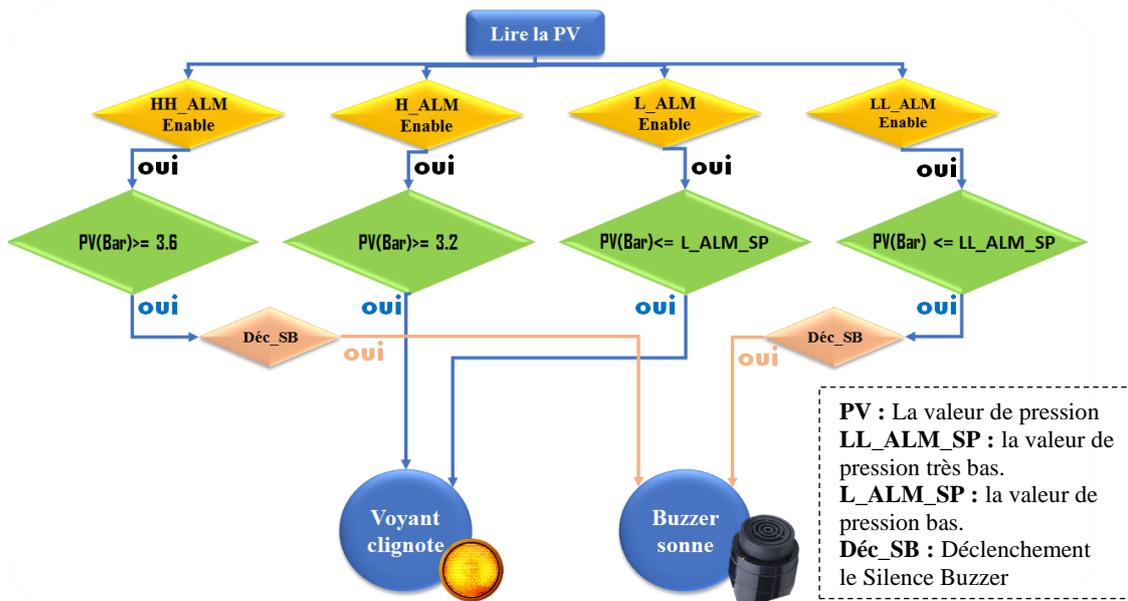
Dans le navigateur du projet et dans la partie gauche de l'écran, développer successivement le « nom du projet », « SIMATIC 300 », « CPU 313C », « Programme S7(1) » et un seul-cliquer sur « Blocs ». Ainsi dans la partie droite de l'écran et sur le vide cliquer un clique-droite pour afficher la liste de commande, puis sélectionner « Insérer un nouvel objet » comme ça en obtenu sous-liste, cliquer sur « Bloc d'organisation », « Bloc fonctionnel », « Fonction », « Bloc de

données », « Type de données » ou « Table des variables » à savoir notre cahier de charge (Figure 3.28).



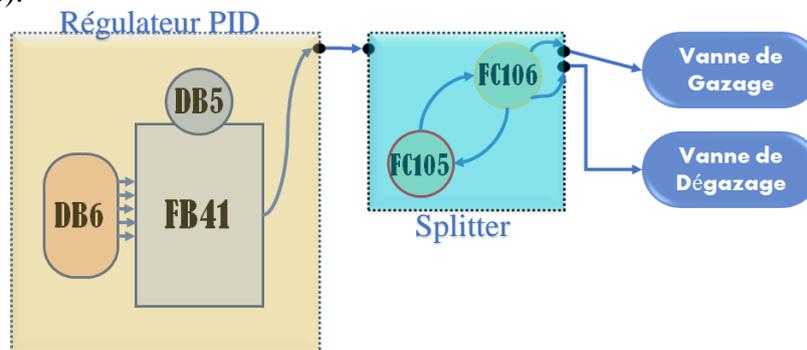
**Figure 3.28 :** Création des blocs et des fonctions

Dans notre projet en à programmer dans la partie « Alarmes » du programme les fonctions FC1 et FC2. FC1 comme un transmetteur de pression, le bloc fonctionnel FB1 comme un bloc d'entrée analogique « Analogue input » et le DB1 c'est le bloc de donnée d'instance de FB1, les types de données utilisateur (UDT1 et UDT2) joue le rôle de types de données pour des variables dans les blocs de données (DB2 et DB3). Cette stratégie permette d'utilise le bloc FB1 dans plusieurs bloc fonction (FC) comme un autre transmetteur de pression ou comme (un transmetteur de température, de niveau, ...), en peut également paramétrer les données du DB2 dans tout le temps de chaque transmetteur pour choisie les différents degrés d'alarme, soit « LL\_ALM » si la pression très bas, « L\_ALM » si bas, « H\_ALM » si haut ou « HH\_ALM » si très haut, peut-on aussi désactiver les alarmes indésirables si ne pas nécessaire. La fonction FC105 (SCALE) convertie la valeur d'entrée analogique (mémoire image de la carte d'entrée analogique d'automate) à une valeur qui présente la grandeur physique (pression). La fonction FC2 « Alarme » peut faire l'écriture sur la carte de sortie numérique d'automate pour actionner les alarmes (voyant et buzzer).



**Figure 3.29 :** L'organigramme présentant les alarmes de notre boucle split range

Dans la partie « Régulation split range » du programme, en à programmer le bloc fonctionnel FB41 avec leur bloc de donnée d'instance DB5 comme un régulateur PID et paramétrer le bloc de donnée globaux DB6 pour obtenir des paramétrages de fonctionnement du régulateur puis le chargement sur le FB41. Pour programmer le splitter (SPLT) en a utilisé les deux fonctions FC105 et FC106 (SCALE et UNSCALE). En fin le FC106 (UNSCALE) faire l'écriture sur la carte de sortie analogique d'automate pour actionner les vannes régulatrice (vanne de engagement ou de dégazage).



**Figure 3.30 :** Le régulateur PID et le Splitter

- Choisir le langage Ladder Diagram (LD) : Nous arrivons enfin à la programmation du Ladder. Pour choisir la programmation en Ladder sur le projet, il faut ajouter un bloc au programme de l'automate en choisissant le langage CONT (Ladder) avant d'ajouter le bloc (Figure 4.31). Pour programmer avec Ladder, il suffit de double-cliquer sur ce bloc, s'ouvre une fenêtre, sélectionner un réseau et cliquer sur les éléments désirés dans la partie haut ou la partie gauche de la fenêtre, le composant sélectionné pourra être déposé.

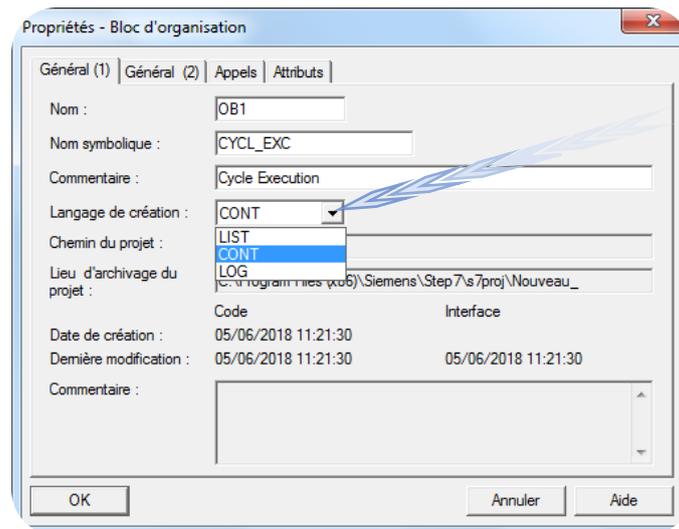


Figure 3.31 : Choisir le langage de programmation Ladder (CONT)

### 3.6.4 Structure d'un programme [17]

L'utilisation des différents blocs permet plusieurs structures de programme

- Programmation linière : Toutes les instructions sont contenues dans un seul bloc (OB1).
- Programmation segmentée : Les instructions pour les différentes fonctions contenues dans différents blocs (FC, FB). Le bloc d'organisation (OB1, OB35 ...) appelle les blocs les uns après les autres.
- Programmation structurée : Les fonctions réutilisable sont programmées dans différents blocs (FC, FB). Le bloc d'organisation (OB1, OB35 ...) appelle ces blocs et transmet les données utiles à l'exécution du programme.

L'automate met à disposition différents types de blocs qui contiennent le programme et les données correspondantes. Selon les exigences et la complexité du processus, il est possible de structurer le programme en différents blocs : OB, FB et FC (Figure 3.32).

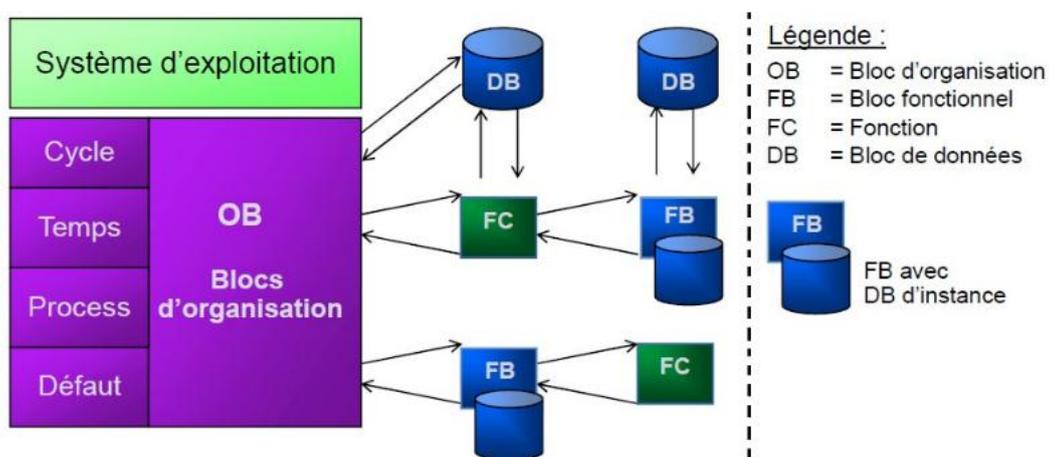


Figure 3.32 : Le degré d'imbrication maximum des blocs dépend la CPU [17]

La structure de notre programme est présente dans la (Figure 3.), le bloc d'organisation OB1 appelle les fonctions FC1 et FC2 l'un après l'autre. La fonction FC1 appelle le bloc fonctionnel FB1. Le bloc de données DB1 est associé à chaque appel de bloc fonctionnel (FB1) transmettant des paramètres, le bloc de données globaux DB2 contiennent des données variables d'entrée du bloc FB1, ce dernier prend les données du DB2 et enregistre les résultats après leurs opérations sur le bloc de données globaux DB3. Pour faire des conversions, le bloc FB1 appelle le bloc FC105 et prend les résultats.

Chaque 100ms le système d'exploitation d'automate pause l'OB1 et donne le droit d'exécution vers le bloc d'alarme cyclique (OB35), ce dernier appelle les fonctions FB41, FC105 et FC106. DB5 est associé à chaque appel de bloc fonctionnel (FB41) transmettant des paramètres. Le bloc de données globaux DB6 émet les données variables vers les entrées du bloc FB41. Le FB41 reçoit les résultats des conversions de FC105 et envoie leurs résultats vers FC106. Il y a toujours un changement de données entre FC105 et FC106.

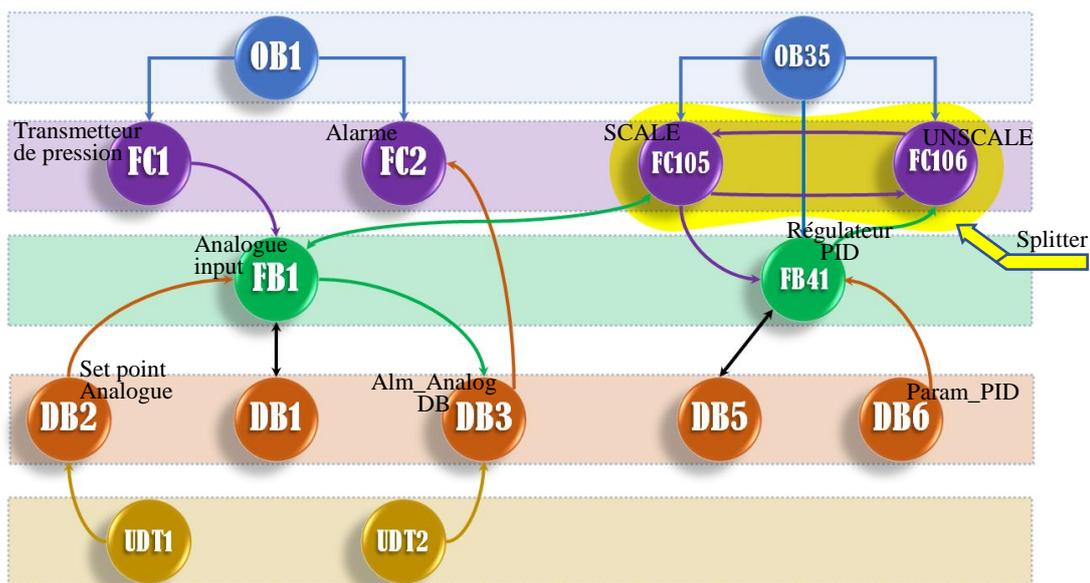


Figure 3.33 : L'organigramme présentant la programmation structurée de notre projet

### 3.6.5 Simulation avec S7-PLCSIM

Pour utiliser l'automate de simulation, il suffit de cliquer sur l'icône « Activer/désactiver la simulation » dans la partie haut du navigateur du projet, on aura la fenêtre du simulateur s'ouvre (Figure 3.34).

S7-PLSIM permet de tester le programme avant son implantation définitive dans l'automate.

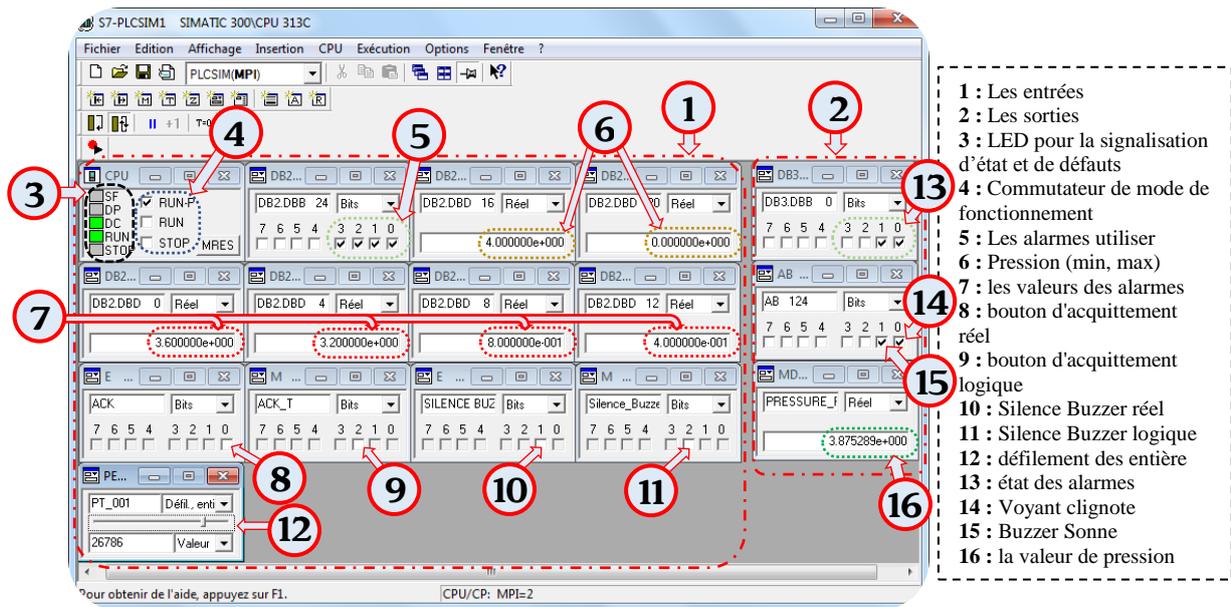


Figure 3.34 : Simulation de la partie « Alarmes » de notre projet avec S7-PLCSIM

### 3.6.6 Test du programme

La visualisation permet de tester le bloc d'un programme, pour cela :

- On ferme l'automate de simulation.
- On vérifie si l'automate est en mode « STOP », sinon on place le bouton situé sur l'unité centrale vers le mode « STOP ».
- On clique sur l'icône « En ligne » dans la partie haut du navigateur et on le charge complètement dans l'appareil par le cliquer sur l'icône « charger ».
- On transfère notre programme et la configuration de l'API dans la mémoire de l'API.
- On place l'automate en mode « RUN » avec le même bouton situé sur l'unité centrale.
- Enfin on teste notre programmation.

#### 3.6.6.1 Calcul des paramètres du bloc FB41 « le régulateur PID »

Comme nous l'avons mentionné plus tôt, on va utiliser le bloc FB41 comme un régulateur PID. Pour cela on a besoin de paramétrer le bloc FB41, pour sélectionner le type de régulateur approprié (régulateur P, PI, PID ...) et aussi pour calculer leur paramètre de fonctionnement dans notre boucle de régulation de notre dégazeur.

- 1) Pour sélectionner le type de régulateur approprié (régulateur P, PI, PID ...) dans notre boucle de régulation de notre dégazeur on va paramétrer le FB41 comme le suivant :
  - Activer le mode manuel du bloc FB41 « MAN\_ON » (mise à 1).
  - Fixer la période d'échantillonnage « SYCLE » à 100ms (suffisante tant que la Période par défaut du bloc d'alarme cyclique OB35 est 100ms).

- Entrer l'adresse de l'écriture de la valeur pression à l'entrée « PV\_IN » du bloc FB41.
- Entrer l'adresse d'une vanne régulatrice (l'adresse sur la carte sortie d'automate) à la sortie « LMN\_PER » du bloc FB41 (en à choisir la vanne de engazage dans notre projet).
- On agissant sur la valeur de manipulation « LMN » du bloc FB41 et on se ramène en condition de service (2 Bar dans notre projet).
- On envoie un échelon «  $\Delta$ LMN » assez faible de l'ordre de (2%).
- Il faut enregistrer l'opération complet avec le traçage des courbes pour faire les calculs.

Le progiciel SIMATIC Manager fourni « Paramétrage du régulateur PID », cette dernière est une application pour tracer les différentes variations des valeurs des entrées et des sorties désirables en temp réel avec des courbes colorées. Pour utilise cette application, il suffit de cliquer sur le bouton démarrer de Windows et suivre le lien suivant : Programmes > Siemens Automation > SIMATIC > STEP 7 > Paramétrage du régulateur PID. Une fenêtre « Régulation PID » s'ouvre. Cliquer sur l'icône « ouvrir » dans la partie haut de la fenêtre, une autre fenêtre « Ouvrir » s'ouvre et dans cette dernière cocher le choix « En ligne » dans la partie haut de la fenêtre, choisi le nom du projet par sélectionner leur nom dans le rectangle de choix « Nom » (partie gauche de cette fenêtre), ci-dessous de ce rectangle se trouve le nom du projet apparait, développer successivement le « nom du projet », « SIMATIC 300 », « CPU 313C », « Programme S7(1) » et un seul-cliquer sur « Blocs », les blocs de données sont apparus au milieu de la fenêtre, un seul cliquer sur le bloc de donnée d'instance du bloc FB41, puis valider par le cliquer sur le bouton « OK » (figure 3.35).

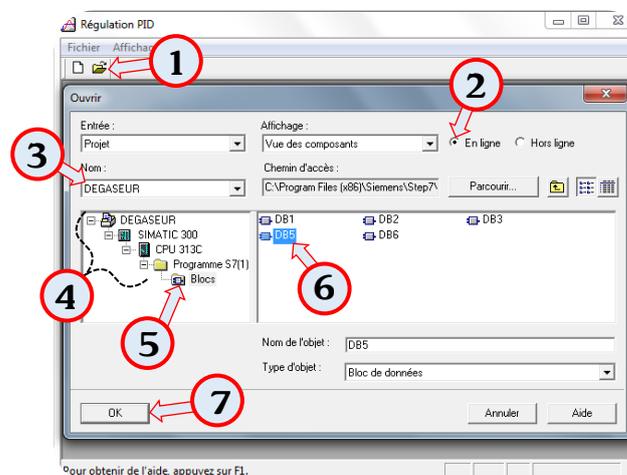


Figure 3.35 : L'application « Paramétrage du régulateur PID »

La fenêtre « Ouvrir » se ferme et l'icône « Traceur de courbes » apparaît dans la partie haut de la fenêtre principale « Régulation PID », un seul-clic sur cette icône pour présenter la fenêtre « Traceur de courbes ». Il faut cliquer sur le bouton « paramétrer » pour choisir les valeurs de consigne et leur couleur de courbe avant de cliquer sur le bouton « Activer » (figure 3.36).

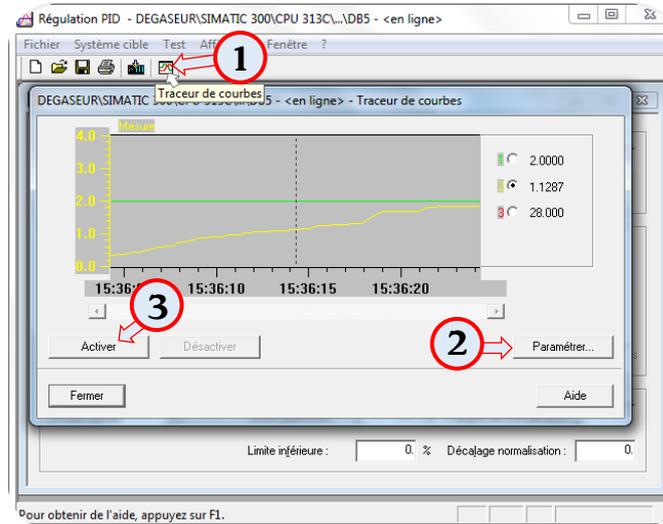


Figure 3.36 : Traceur de courbes

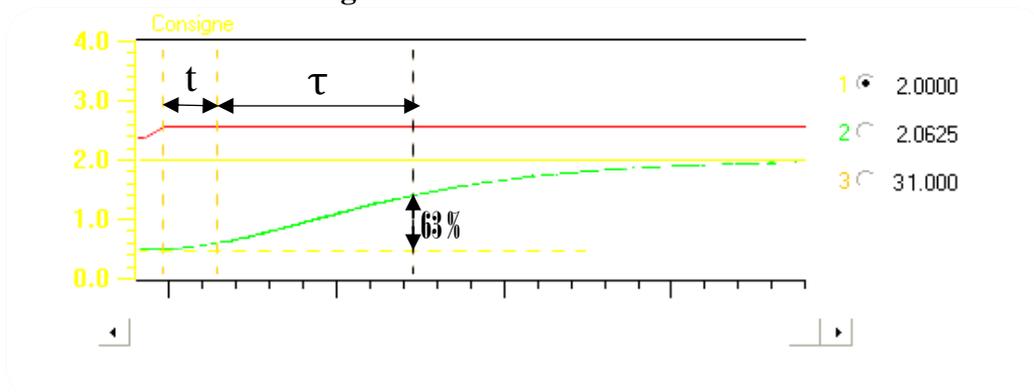


Figure 3.37 : Présentation t et  $\tau$  avec le traceur de courbes

t : le temp de retard.

$\tau$  : le temp a 63% du  $\Delta PV$ .

$\Delta PV$  : la différence entre la pression au début d'opération et jusqu'au la stabilité.

- On calcule le taux de réglabilité « r » pour identification du système, selon l'équation suivante :

$$r = \frac{t}{\tau} \quad (3.1)$$

Dans notre projet  $t = 4s$  et  $\tau = 150s$  donc  $r \approx 0.026$ .

Tant que  $r < 0,3 \Rightarrow$  le régulateur PI est approprié pour notre système.

- 1) « Ziegler et Nichols » proposent de calculer les paramètres du régulateur P, PI et PID à l'aide des équations présentées dans le tableau suivant :

**Tableau 3.5 :** Les paramètres du régulateur P, PI et PID

Type de régulateur	Kp	Ti	Td
P	$\frac{\tau}{G_s t}$	*	*
PI	$0,9 \frac{\tau}{G_s t}$	$3,33 t$	*
PID	$1,2 \frac{\tau}{G_s t}$	$2,0 t$	$0,5 t$

On peut calculer le gain statique « Gs » selon l'équation suivante :

$$G_s = \frac{\Delta PV}{\Delta LMN} \quad (3.2)$$

A partir des équations précédentes, on a trouvé les paramètres de notre régulateur PI.

- 2) Enfin, on teste notre boucle de régulation en mode auto, il faut suivre les étapes suivantes :

- Mise à (1) les entrées « P\_SEL », « I\_SEL » et mise à (0) l'entrée « D\_SEL » du bloc FB41 pour le paramétrer comme un régulateur PI.
- Entrer les valeurs Kp et Ti dans les entrées « GAIN » et « Ti » du bloc FB41.
- Pour activer le mode auto du bloc FB41, il suffit de désactiver le « MAN\_ON » et le mettre à 0.

Remarque : pour utiliser les paramètres calculés Kp et Ti dans la boucle split range, on a ajouté 60% à ces deux valeurs, tant que l'échelle de la boucle est 60%.

### 3.6.6.2 Le splitter

Le principe de notre splitter est de recevoir un signal de commande unique « LMN » du bloc FB41 et émettre deux signaux vers la carte sortie d'automate, un signal de sortie pour chaque vanne dans une paire à intervalle divisé.

Pour un signal de « LMN » entre (0 et 60%) il émet un autre signal inverse entre (100 et 0%) vers la vanne de dégazage, on ne peut pas faire cette conversion de façon directe, mais on peut le faire à l'aide des fonctions FC106 et FC105.

Comme nous l'avons mentionné plus tôt, avec FC106 peut-on convertir des valeurs réelles entre (LO\_LIM et HI\_LIM) à des valeurs entières entre (0 et 27648 si configuré à unipolaire) ou entre (-27648 et 27648 si configuré à bipolaire). FC105 peut-on de fonctionner contrairement, et convertir des valeurs entières (de même intervalle si la même configuration « unipolaire » ou « bipolaire ») à des valeurs réelles (LO\_LIM et HI\_LIM).

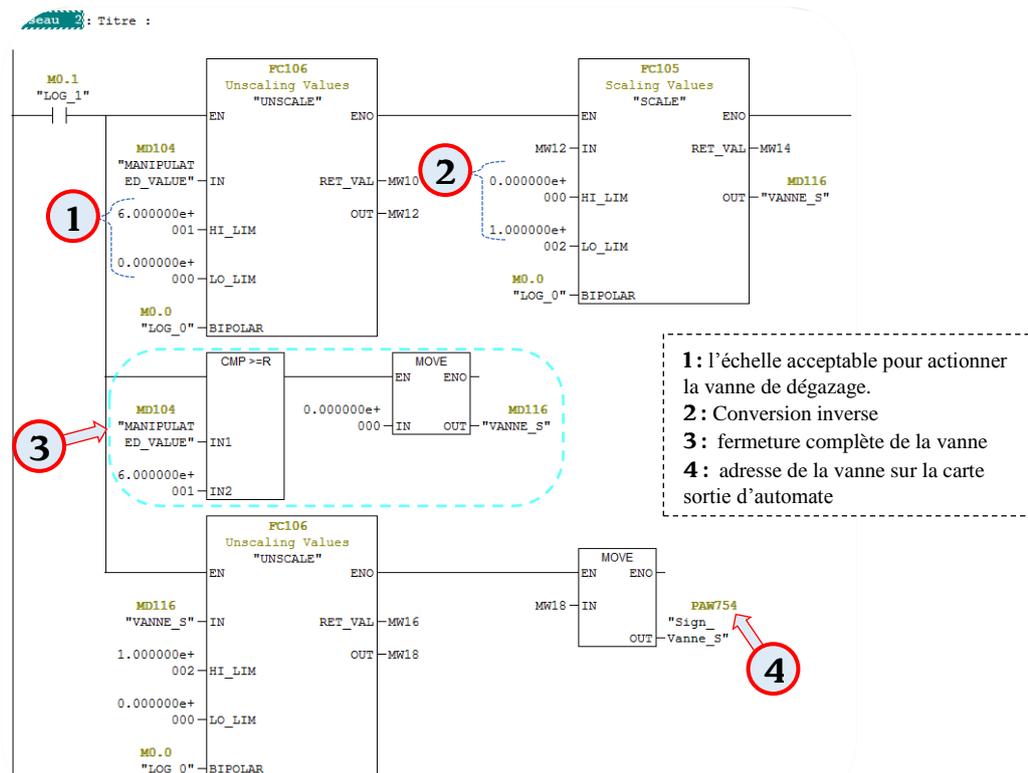
Donc, pour créer un convertisseur réel-réel a notre splitter, il suffit de paramétrer les fonctions FC106 et FC105 comme le (tableau 3.6) et utilise les résultats de FC106 à l'entrée de FC105 (utilise même adresse).

**Tableau 3.6 :** Limite de conversion des FC106 et FC105

Limite de conversion	FC106	FC105
LO_LIM	0%	100%
HI_LIM	60%	0%

On a fait maintenant le partage d'échelle, nous avons sélectionné l'intervalle d'actionnement de la vanne de dégazage à partir de signal « LMN » (la valeur de manipulation) et on a trouvé le pourcentage d'ouverture. Mais pour l'écriture sur la carte sortie d'automate, il faut convertir la valeur entière (ouverture %) a la valeur réelle correspondant. Pour cela on a utilisé autre fois la fonction FC106.

Afin que les deux vannes ne soient pas ouvertes en même temps, on a fermé complètement si hors l'intervalle de fonctionnement.



**Figure 3.38 :** Splitter « Partie d'actionnement la vanne de dégazage »

On a utilisé la même stratégie pour programmer la partie d'actionnement de la vanne d'engazage (figure 3.39).

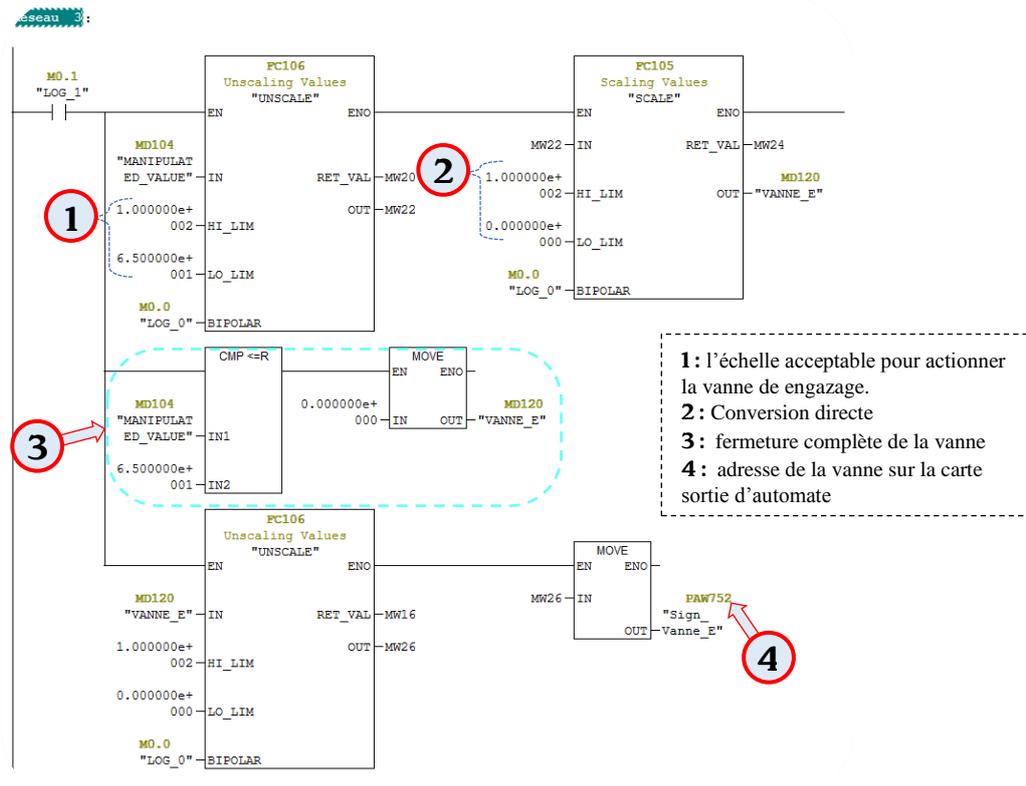


Figure 3.39 : Splitter « Partie d'actionnement la vanne d'engagement »

### 3.7 Le logiciel de supervision WinCC flexible 2008

WinCC est un système HMI (Human Machine Interface) très performant, il constitue l'interface entre l'homme (opérateur) et la machine (installation/processus).

La supervision industrielle consiste à surveiller l'état de fonctionnement en temps réel d'un procédé pour l'amener à son point de fonctionnement optimal.

#### 3.7.1 Avantage de la supervision

La supervision industrielle permet de faire :

- La surveillance du processus à distance.
- La détection des défauts.
- Le diagnostic et le traitement des alarmes.
- Le traitement des données.

#### 3.7.2 Composants de WinCC

Le système de base se subdivise en :

- Logiciel de configuration WinCC flexible Engineering System qui permet de créer un projet.
- Logiciel Runtime permet de mettre le projet en œuvre dans le cadre du processus.

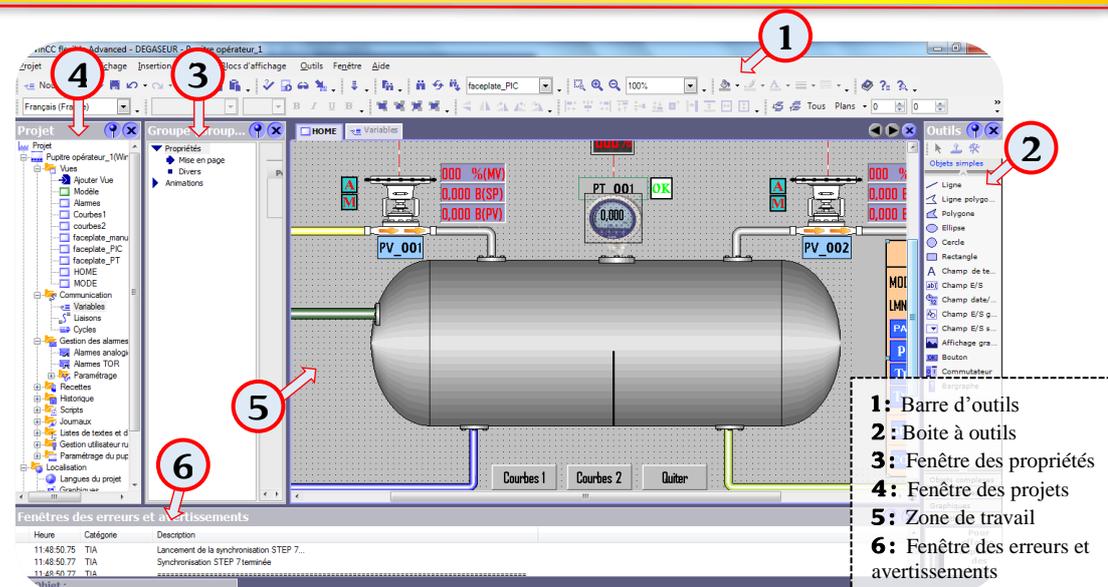


Figure 3.40 : notre boucle split range sur le WinCC

### 3.7.3 Configuration

De manière générale, WinCC permet de réaliser les configurations système suivantes :

- Système monoposte : lorsqu'un pupitre operateur est relie directement à un automate via le bus système.
- Système multiposte : avec un serveur et plusieurs clients.
- Système IHM : avec fonctions centrales.

### 3.7.4 Elément de WinCC flexible 2008

Logiciel WinCC flexible constitue les éléments suivants :

- Les liaisons.
- Les variables.
- Les bibliothèques.
- Les vues.
- Les systèmes de signalisation.
- L'archivages et visualisation des variables.
- L'utilisation de journaux.

L'API qu'on a utilisée S7-300 CPU313C ne contient pas « Profibus » et le câble « MPI », ne permis pas de faire un déplacement et contrôlé notre système (procédé) a langue distance.

Pour cela, on a :

- Configurer notre PC avec un smartphone dans un réseau WI-FI qui couvre tout l'espace de travail.
- Installer un programme de contrôle smartphone-PC, dans notre cas « PC Remote ».

Comme ça, on peut de gérer tout le procédé, fait les tests nécessaires par notre HMI sur le WinCC flexible et par distance.

3.7.5 Résultat obtenu

Les figures suivantes présentent les réactions de notre boucle split-range avec les perturbations, qui nous avons appliqué par les vannes manuels d'engazage et de dégazage :

- 1) On a fait une perturbation par la vanne manuelle de dégazage et on a observé comment le signale de commande « LMN » augmente pour ouvrir la vanne régulatrice d'engazage et reste de gérer l'ouverture jusqu'à l'obtention d'une pression stable vers la consigne.

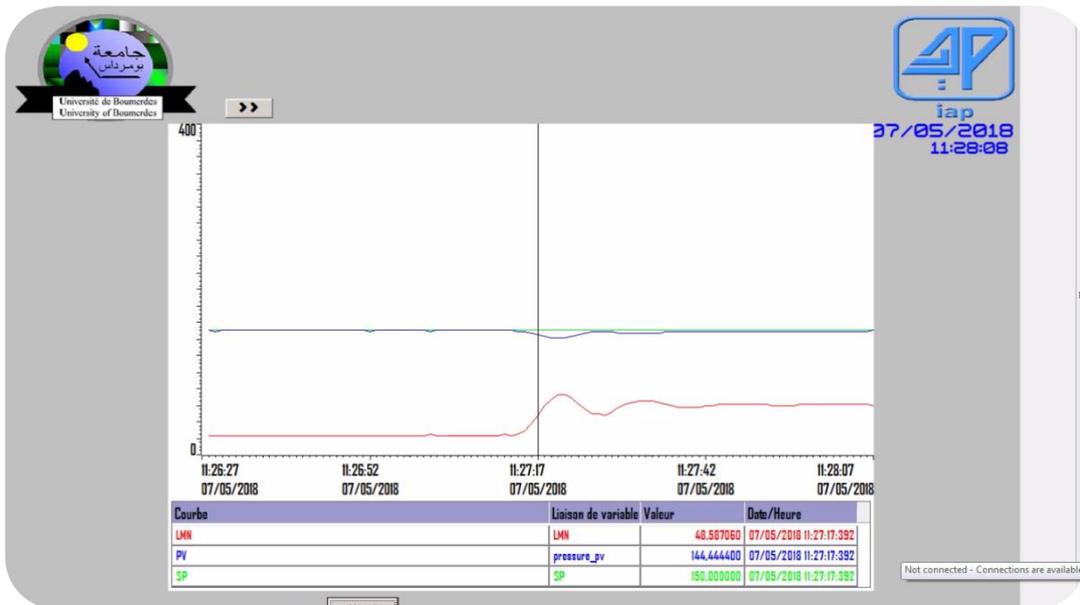


Figure 3.41 : perturbation par la vanne de dégazage manuelle

- 2) Encore on a fait une perturbation, mais cette fois par la vanne manuelle d'engazage, on a observé que le « LMN » chute, tant que l'ouverture de la vanne régulatrice de dégazage s'inverse avec le « LMN », on a vu la chute de pression jusqu'à la stabilité.

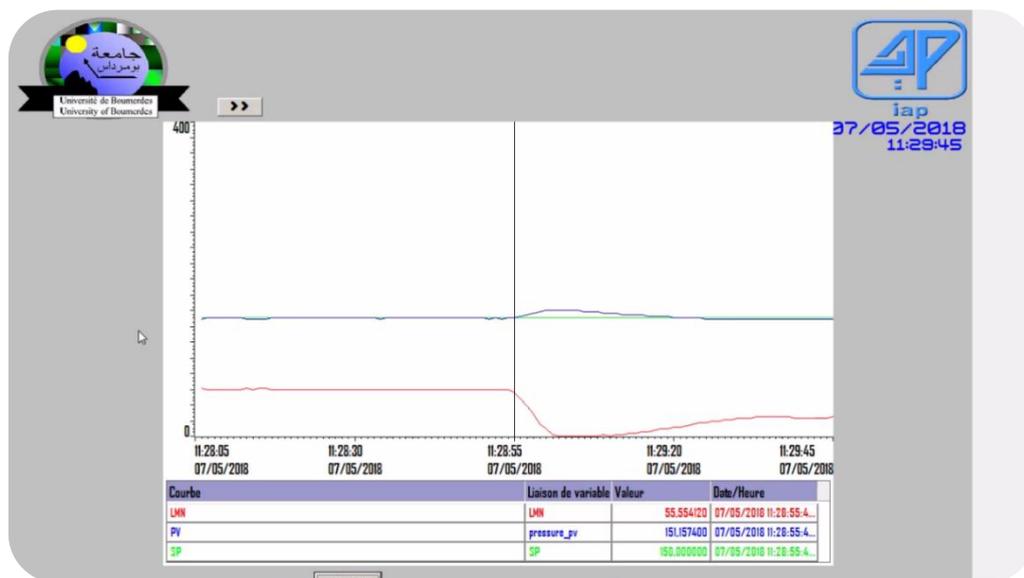


Figure 3.42 : perturbation par la vanne de dégazage manuelle

Les figures suivantes présentes la réaction des vannes avec les perturbations, qui nous avons appliqué :

- 1) La pression est inférieure à la consigne, donc le régulateur calcule un signal de « LMN = 86% » et le contrôleur par le splitter fait une ouverture de 61% de la vanne d'engazage et ferme l'autre complètement.

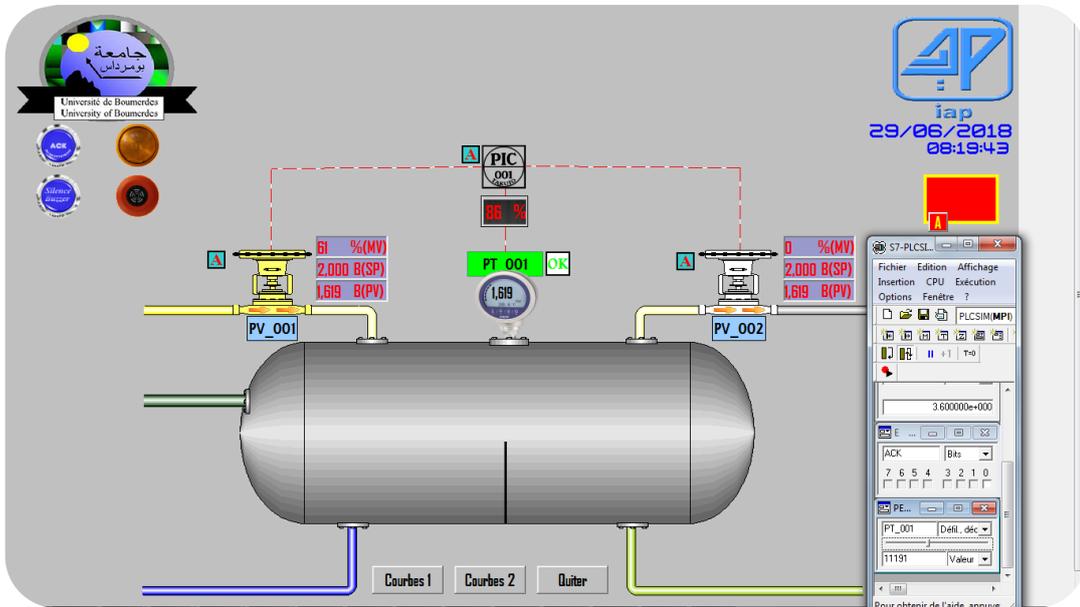


Figure 3.43 : cas d'engazage en boucle split-range

- 2) La pression est supérieure à la consigne, donc le régulateur calcule un signal de « LMN = 34% » et le contrôleur fait une ouverture de 44% de la vanne de dégazage et ferme l'autre complètement.

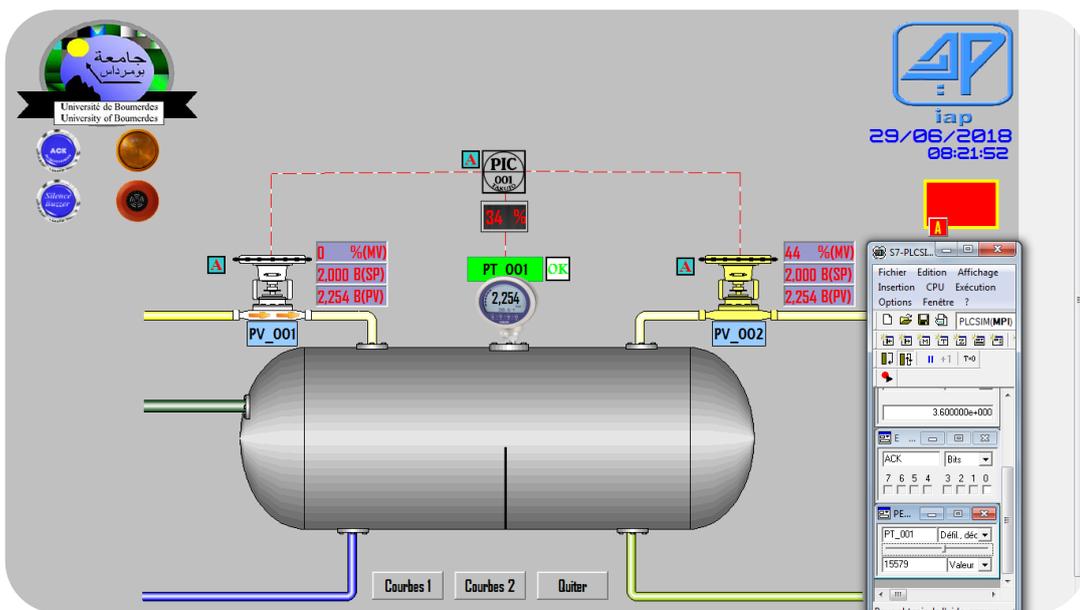


Figure 3.44 : cas de dégazage en boucle split-range

**Conclusion**

Le programme que nous avons édité a été implémenté dans notre PC et ce pour des raisons de simulation sous le progiciel SIMATIC Manager.

A l'aide du PLCSIM on a testé et simulé toutes les étapes de la régulation split range du processus cadre de notre étude, on a utilisé aussi le WinCC flexible 2008 pour superviser complètement les étapes du processus.

## Conclusion générale

---

La majorité des processus industriels nécessitent le contrôle d'un certain nombre de grandeurs physiques telles que la température, la pression, le niveau, le débit, le pH, la concentration, etc. Il appartient à la chaîne de régulation (et plus généralement à la chaîne d'asservissement) de maintenir ces grandeurs à des niveaux prédéterminés.

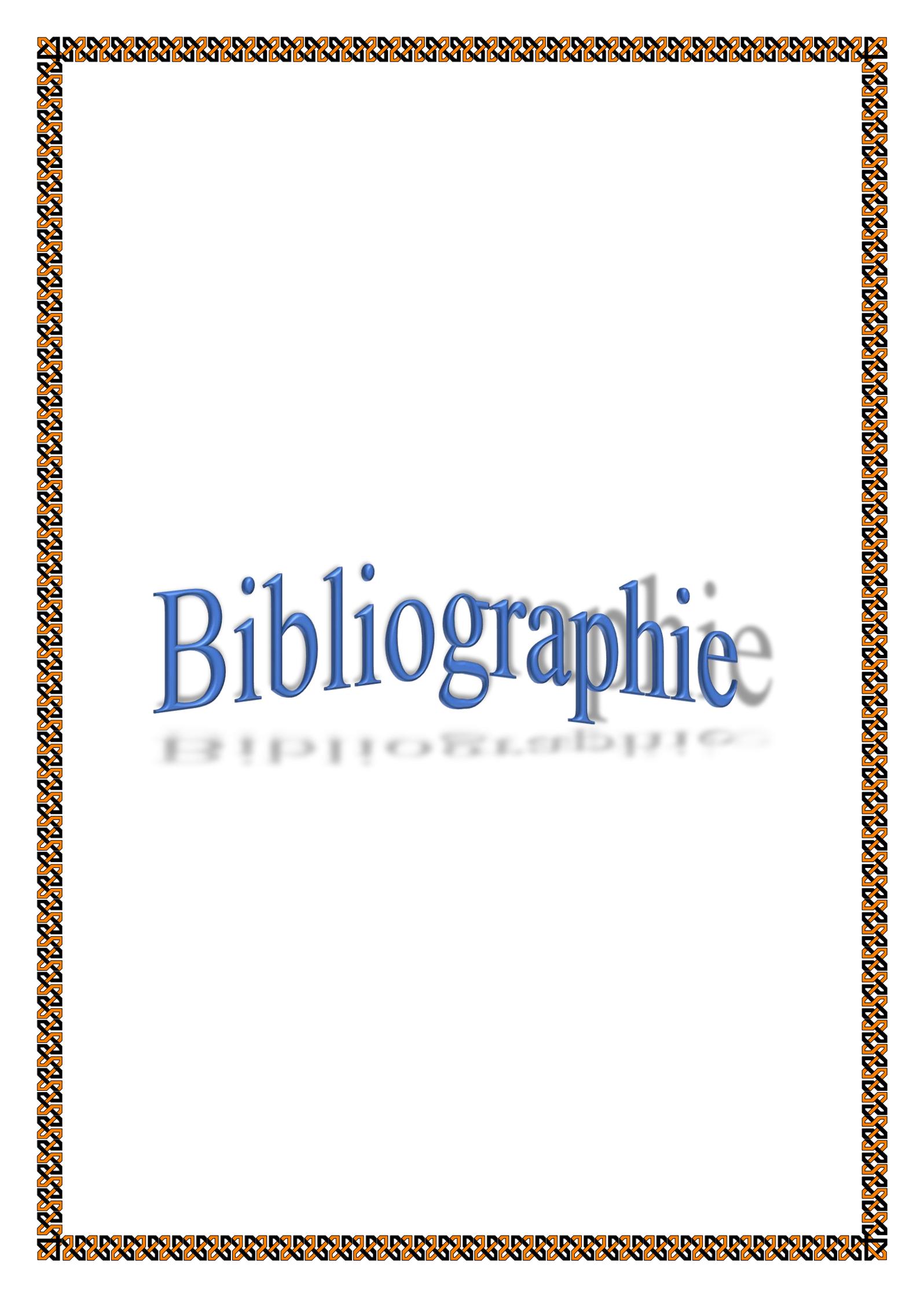
La boucle de régulation split-range est un type qu'on le trouve dans plusieurs domaines industriels. Pour cela, on a présenté dans le deuxième chapitre « Régulation industrielle et boucle de régulation split range » les différentes applications des différents domaines.

Notre travail est une expérience intéressante, qui nous a permis de réaliser un prototype d'un dégazeur, d'appliquer une boucle split-range pour satisfaire la régulation à base d'un automate S7-300 et la belle chose, notre participation à toutes les étapes du processus, aussi le développement d'algorithmes et la programmation.

L'élaboration de ce travail nous a permis, d'une part, d'approfondir nos connaissances dans le domaine électronique des systèmes embarqués et le savoir-faire acquis durant les années de notre cycle d'études et d'autre part, de bien préparer notre intégration dans la vie professionnelle.

Le travail que nous avons réalisé pourrait être complété et poursuivi sous différents aspects, notamment :

- L'application de notre projet dans une unité de production.
  - Tester la fiabilité et le rendement de notre boucle dans plusieurs domaines de l'industrie.
-



# Bibliographie

BIBLIOGRAPHIE

## Bibliographie

---

- [1] Mise en œuvre de l'IAP, Chapitre 1 : Présentation du lieu de stage (document interne de l'Institut Algérien de Pétrole).
  - [2] Mémoire présenté en vue de l'obtention du diplôme de Master en : Génie des Procédés, Option : Génie Chimique, Titre : Valorisation et traitement des eaux huileuses dans l'industrie pétrolière, Présenté par : Necib Sana, Université Mohamed Khider – Biskra, Faculté des Sciences et de la technologie, Département : Chimie Industrielle, Promotion Juin 2015.
  - [3] Guide-technique-régulation-Le magazine Schneider Electric de l'enseignement technologique et professionnel-juin 2004.
  - [4] Assemblage à Google image et traitement sur PC.
  - [5] [www.technologuepro.com/cours-genie-electrique/cat-3-cours-automatisme-informatique-industrielle](http://www.technologuepro.com/cours-genie-electrique/cat-3-cours-automatisme-informatique-industrielle), 11 Mai 2018 – 02h : 13m.
  - [6] Régulation et Asservissement - Yassine Ariba - Dpt GEI - Icam, Toulouse.
  - [7] Le PID utilisé en régulation de position et/ou de vitesse de moteurs électriques, Christophe Le Lann 2007, <http://www.totofweb.net/projets/pid/rapport.pdf>.
  - [8] Boucles de régulation ch8, <https://www.technologuepro.com/Regulation-industrielle/Boucles-de-Regulation.pdf>, 20 Juin 2018 – 09h :37m.
  - [9] [www.polytech-lille.fr/cours-regulation-automatique](http://www.polytech-lille.fr/cours-regulation-automatique), 28 Avril 2018 – 01h :45m.
  - [10] [www.specialautom.net](http://www.specialautom.net), 13 Mai 2018 – 03h : 16m.
  - [11] EURO THERM AUTOMATION SERVICE RÉGIONAL-année 2000.
  - [12] Lessons In Industrial Instrumentation c 2008-2016 by Tony R. Kuphaldt – under the terms and conditions of the Creative Commons Attribution 4.0 International Public License, Version 2.19 (development) – Last update July 4, 2016.
  - [13] Mémoire de fin d'étude en vue d'obtention du diplôme de master 2, Thème : Automatisation de la mise en marche, arrêt et protection du circuit AIR/FUMÉE de la chaudière par un API SIEMENS S7-300, Réalisé par : Mr RIAL Islam et Mr AMARA Said, Université M'hamed Bougara Boumerdes, Faculté des Sciences de l'Ingénieur, Département : Maintenance Industrielle, Option : Génie Electrique, Spécialité : Automatique, Année 2015-2016.
  - [14] Les Automates Programmables Industriels (API), Philippe RAYMOND, Notes de cours - BTS MAI - octobre 2005.
  - [15] [www.automation-sense.com/blog/automatisme/pg-pc-et-configuration-des-interfaces-pg-pc-sous-step7-siemens.html](http://www.automation-sense.com/blog/automatisme/pg-pc-et-configuration-des-interfaces-pg-pc-sous-step7-siemens.html), 25 Avril 2018 – 01h :21m.
-

## Bibliographie

---

[16] [www.siemens.com](http://www.siemens.com), 12 Juin 2018 – 02h : 04m.

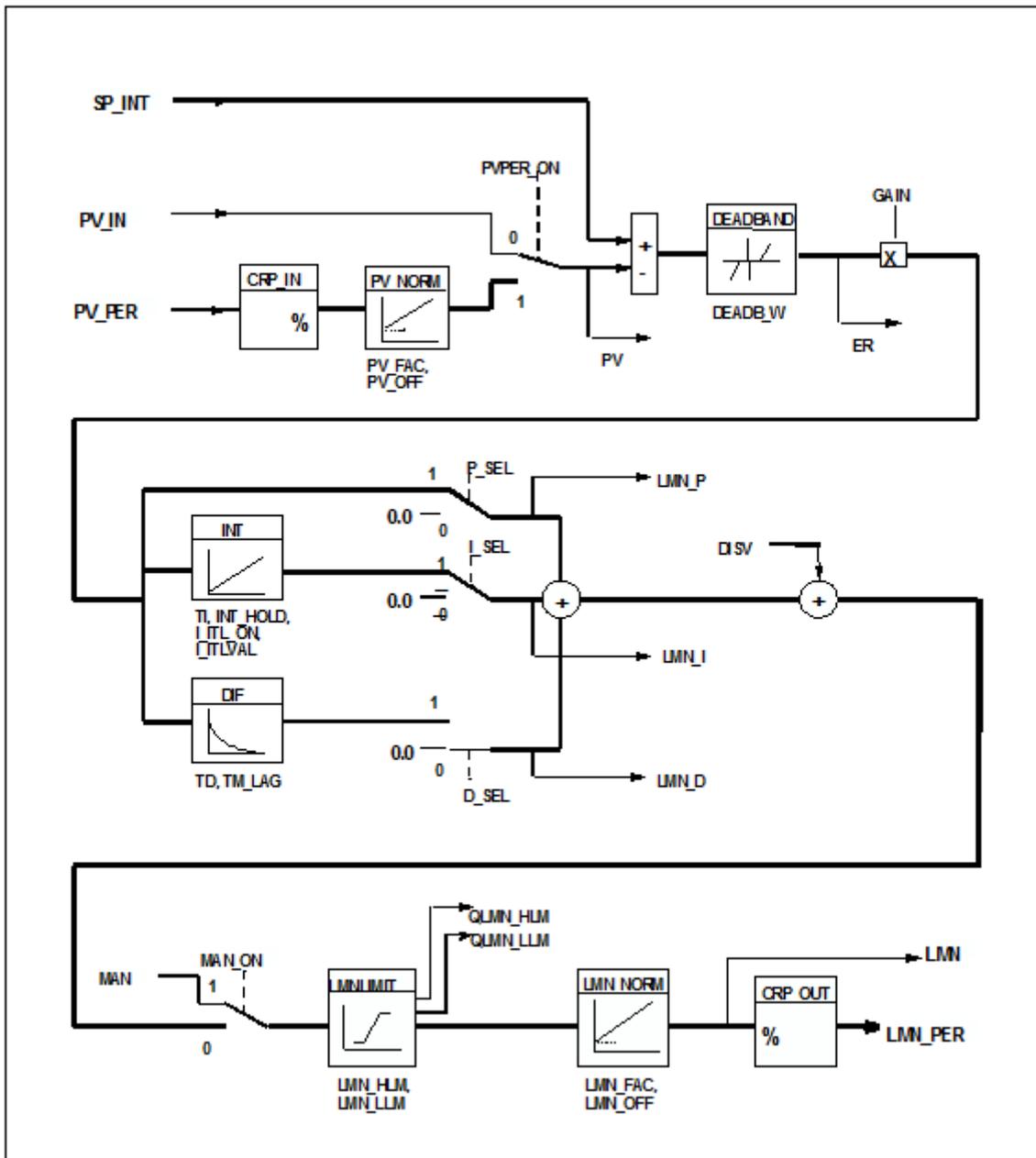
[17] <http://tia-portal.weebly.com/10-types-de-blocs-et-structure.html>, 11 Juin 2018 - 02h :30m.

---



# Annexe

## Schéma fonctionnel du bloc FB41



## Paramètres d'entrée

Les tableaux suivants présentent les paramètres d'entrée du bloc SFB41/FB41 "CONT\_C".

Paramètre	Type de données	Valeurs admises
COM_RST	BOOL	
MAN_ON	BOOL	
PVPER_ON	BOOL	
P_SEL	BOOL	
I_SEL	BOOL	
INT_HOLD	BOOL	

## Annexe

I_ITL_ON	BOOL	
D_SEL	BOOL	
CYCLE	TIME	>= 1 ms
SP_INT	REAL	-100,0 à 100,0 (%)ou grandeur physique 1
PV_IN	REAL	-100,0 à 100,0 (%)ou grandeur physique 1
PV_PER	WORD	
MAN	REAL	-100,0 à 100,0 (%)ou grandeur physique 2
GAIN	REAL	
TI	TIME	>= CYCLE
TD	TIME	>= CYCLE
TM_LAG	TIME	>= CYCLE/2
DEADB_W	REAL	>= 0,0 (%)ou grandeur physique 1
LMN_HLM	REAL	LMN_LLM à 100,0 (%)ou grandeur physique 2
LMN_LLM	REAL	-100,0 à LMN_HLM (%)ou grandeur physique 2
PV_FAC	REAL	
PV_OFF	REAL	
LMN_FAC	REAL	
LMN_OFF	REAL	
I_ITLVAL	REAL	-100,0 à 100,0 (%)ou grandeur physique 2
DISV	REAL	-100,0 à 100,0 (%)ou grandeur physique 2

Paramètre	Par défaut	Description
COM_RST	FALSE	COMPLETE RESTARTLe bloc a un sous-programme d'initialisation qui est exécuté quand cette entrée est à 1.
MAN_ON	TRUE	MANUAL VALUE ON / Activation du mode manuelQuand cette entrée est à 1, la boucle de régulation est interrompue. C'est une valeur manuelle qui est introduite comme valeur de réglage.
PVPER_ON	FALSE	PROCESS VARIABLE PERIPHERY ON / Activation de la mesure de périphérie Pour que la mesure soit lue dans la périphérie, il faut relier l'entrée PV_PER à la périphérie et mettre à 1 l'entrée PVPER_ON.
P_SEL	TRUE	PROPORTIONAL ACTION ON / Activation de l'action proportionnelleDans 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	TRUE	INTEGRAL ACTION ON / Activation de l'action par intégrationDans 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.
INT_HOLD	FALSE	INTEGRAL ACTION HOLD / Gel de l'action par intégrationLa sortie de l'intégrateur peut être gelée. Pour cela, il faut mettre à 1 cette entrée.
I_ITL_ON	FALSE	INITIALIZATION OF THE INTEGRAL ACTION / Initialisation de l'action par intégrationLa sortie de l'intégrateur peut être forcée à la valeur initiale I_ITL_VAL. Pour cela, il faut mettre à 1 cette entrée.
D_SEL	FALSE	DERIVATIVE ACTION ON / Activation de l'action par dérivationDans 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	T#1s	SAMPLE TIME / Période d'échantillonnageLe temps s'écoulant entre les appels de bloc doit être constant. Il est indiqué par cette entrée.

## Annexe

SP_INT	0	INTERNAL SETPOINT / Consigne interne Cette entrée sert à introduire une consigne.
PV_IN	0	PROCESS VARIABLE IN / Mesure interne Cette entrée permet de paramétrer une valeur de mise en service ou de relier une mesure externe en virgule flottante.
PV_PER	W#16#0000	PROCESS VARIABLE PERIPHERIE / Mesure de périphérie La mesure en format de périphérie est reliée au régulateur à cette entrée.
MAN	0	MANUAL VALUE / Valeur manuelle Cette entrée sert à introduire une valeur manuelle grâce à des fonctions de contrôle-commande.
GAIN	2	PROPORTIONAL GAIN / Coefficient d'action proportionnelle Cette entrée indique le gain du régulateur.
TI	T#20 s	RESET TIME / Temps d'intégration Cette entrée détermine le comportement dans le temps de l'intégrateur.
TD	T#10 s	DERIVATIVE TIME / Temps de dérivation Cette entrée détermine le comportement dans le temps du dérivateur.
TM_LAG	T#2 s	TIME LAG OF THE DERIVATE ACTION / Retard de l'action par dérivation L'algorithme de l'action D contient un retard qui peut être paramétré à cette entrée.
DEADB_W	0	DEAD BAND WIDTH / Largeur de zone morte Le signal d'erreur est conduit par une zone morte. Cette entrée détermine la taille de la zone morte.
LMN_HLM	100	MANIPULATED VALUE HIGH LIMIT / Limite supérieure de la valeur de réglage La valeur de réglage est toujours limitée vers le haut et vers le bas. Cette entrée indique sa limite supérieure.
LMN_LLM	0	MANIPULATED VALUE LOW LIMIT / Limite inférieure de la valeur de réglage La valeur de réglage est toujours limitée vers le haut et vers le bas. Cette entrée indique sa limite inférieure.
PV_FAC	1	PROCESS VARIABLE FACTOR / Facteur de normalisation Cette entrée est multipliée par la mesure. Elle sert à adapter la plage de la mesure.
PV_OFF	0	PROCESS VARIABLE OFFSET / Décalage de normalisation Cette entrée est ajoutée à la mesure. Elle sert à adapter la plage de la mesure.
LMN_FAC	1	MANIPULATED VALUE FACTOR / Facteur de valeur de réglage Cette entrée est multipliée par la valeur de réglage. Elle sert à adapter la plage de la valeur de réglage.
LMN_OFF	0	MANIPULATED VALUE OFFSET / Décalage de valeur de réglage Cette entrée est ajoutée à la valeur de réglage. Elle sert à adapter la plage de la valeur de réglage.
I_ITLVAL	0	INITIALIZATION VALUE OF THE INTEGRAL ACTION / Valeur d'initialisation pour l'action par intégration La sortie de l'intégrateur peut être forcée par l'entrée I_ITL_ON. La valeur d'initialisation est à cette entrée.
DISV	0	DISTURBANCE VARIABLE / Grandeur perturbatrice La grandeur perturbatrice de compensation est reliée à cette entrée.

1. Paramètres dans les branches de consigne et de mesure avec même unité.
2. Paramètres dans la branche de valeur de réglage avec même unité.

## Annexe

### Paramètres de sortie

Le tableau suivant présente les paramètres de sortie du bloc SFB41/FB41 "CONT\_C".

Paramètre	Type de données	Valeurs admises	Par défaut	Description
LMN	REAL		0	MANIPULATED VALUE / Valeur de réglage Cette sortie donne en virgule flottante la valeur de réglage agissant réellement.
LMN_PER	WORD		W#16#0000	MANIPULATED VALUE PERIPHERY / Valeur de réglage de périphérie La valeur de réglage en format de périphérie est reliée avec le régulateur à cette sortie.
QLMN_HLM	BOOL		FALSE	HIGH LIMIT OF MANIPULATED VALUE REACHED / Limite supérieure de la valeur de réglage atteinte La valeur de réglage est toujours limitée vers le haut et vers le bas. Cette sortie signale le dépassement de sa limite supérieure.
QLMN_LLM	BOOL		FALSE	LOW LIMIT OF MANIPULATED VALUE REACHED / Limite inférieure de la valeur de réglage atteinte La valeur de réglage est toujours limitée vers le haut et vers le bas. Cette sortie signale le dépassement de sa limite inférieure.
LMN_P	REAL		0	PROPORTIONALITY COMPONENT / Composant PC Cette sortie contient le composant proportionnel de la grandeur réglante.
LMN_I	REAL		0	INTEGRAL COMPONENT / Composant IC Cette sortie contient le composant intégral de la grandeur réglante.
LMN_D	REAL		0	DERIVATIVE COMPONENT / Composant DC Cette sortie contient le composant dérivé de la grandeur réglante.
PV	REAL		0	PROCESS VARIABLE / Mesure Cette sortie donne la mesure agissant réellement.
ER	REAL		0	ERROR SIGNAL / Signal d'erreur Cette sortie donne le signal d'erreur agissant réellement.

## Annexe

Below are some common P&ID symbols used with the instrument abbreviations discussed above for developing P&ID drawings

<p>○ LOCALLY MOUNTED INSTRUMENT</p> <p>◐ BOARD MOUNTED INSTRUMENT</p> <hr/> <p>○ PC PRESSURE CONTROLLER</p> <p>○ PI PRESSURE INDICATOR</p> <p>○ PR PRESSURE RECORDER</p> <p>○ PIC PRESSURE INDICATING CONTROLLER</p> <p>○ PRC PRESSURE RECORDING CONTROLLER</p> <p>○ PSV PRESSURE SAFETY VALVE</p> <p>○ RV RELIEF VALVE</p> <hr/> <p>○ LA LEVEL ALARM</p> <p>○ LAH LEVEL ALARM HIGH</p> <p>○ LAL LEVEL ALARM LOW</p> <p>○ LC LEVEL CONTROLLER</p> <p>○ LG LEVEL GLASS</p> <p>○ LI LEVEL INDICATOR</p> <p>○ LIC LEVEL INDICATING CONTROLLER</p> <p>○ LRC LEVEL RECORDING CONTROLLER</p>	<p>○ FA FLOW ALARM</p> <p>○ FE FLOW ELEMENT</p> <p>○ FI FLOW INDICATOR</p> <p>○ FR FLOW RECORDER</p> <p>○ FRC FLOW RECORDING CONTROLLER</p> <hr/> <p>○ TA TEMPERATURE ALARM</p> <p>○ TI TEMPERATURE INDICATOR</p> <p>○ TR TEMPERATURE RECORDER</p> <p>○ TRC TEMPERATURE RECORDING CONTROLLER</p> <p>○ TW TEMPERATURE WELL</p> <hr/> <p>⊗ GATE VALVE</p> <p>⊗ GLOBE VALVE</p> <p>⊗ CHECK VALVE</p> <p>⊗ CONTROL VALVE</p> <p>⊗ PLUG VALVE</p> <p>⊗ BALL VALVE</p> <p>⊗ BUTTERFLY VALVE</p>	<p>⊠ USD UNIT SHUT DOWN</p> <p>○ ZSC V-230 POSITION/ UNIT SWITCH CLOSED</p> <p>○ SDY V-230 SHUT DOWN VALVE RELAY</p> <p>○ SDV V-230 SHUT DOWN VALVE</p> <p>⊠ ZIO F-250 POSITION/ LIMIT INDICATOR OPEN</p> <p>○ TY F-250 TEMPERATURE RELAY</p> <hr/> <p>⊗ SPECTACLE BLIND OPEN</p> <p>⊗ SPECTACLE BLIND CLOSED</p> <p>⊗ ORIFICE FLANGES</p> <p>⊠ SP PIPING SPECIALITY ITEM</p> <hr/> <p>##### INSTRUMENT AIR LINE</p> <p>----- INSTRUMENT ELECTRICAL</p> <p>-x-x-x-x-x- INSTRUMENT CAPILLARY TUBING</p> <p>———— PIPE</p> <hr/> <p>○ T TRANSMITTER (OR) ⊗</p> <p>○ HCV HAND CONTROL VALVE</p>
--	---	---

The table below contains some of the instrument abbreviations used in conjunction with P&ID symbols in instrumentation diagrams. I have dealt with some of them before but for the purpose of emphasis and completeness let us go through again. The list here is by no means exhaustive but it is a good starting point for beginners to P&IDs:

## Annexe

<b>Instrument Abbreviation</b>	<b>Expansion</b>	<b>Functions Performed</b>
FC	Flow controller	Flow measurement and control
LC	Level controller	Level control
FE	Flow element	Flow sensor
LG	Level gauge	Level measurement
FIC	Flow indicator and controller	Indicating flow as well as controlling flow
LA	Level alarm	Indicating level alarm
FR	Flow recorder	Recording flow
LAH	Level alarm high	Indicating high level
FRC	Flow recorder and controller	Flow recording; controlling flow
LAHH	Level alarm high high	Indicating very high level
FT	Flow transmitter	Transmitting flow signal
LAL	Level alarm low	Indicating low level
FA	Flow alarm	Indicating flow alarm
LI	Level indicator	Level indication
LIC	Level indicator and controller	Indicating level; controlling level
PC	Pressure controller	controlling pressure
TC	Temperature controller	Controlling/regulating temperature
PI	Pressure indicator	Indicating pressure
TI	Temperature indicator	Indicating pressure
PIC	Pressure indicator and controller	Indicating pressure; controlling pressure
TIC	Temperature indicator and controller	Indicating temperature; controlling temperature
PR	Pressure recorder	Recording pressure
TR	Temperature recorder	Recording temperature

## Annexe

PRC	Pressure recorder and controller	Recording pressure; controlling pressure
TRC	Temperature recorder and controller	Recording temperature; controlling temperature
PSV	Pressure safety valve	Relieving excess pressure in case of high pressure situation
TT	Temperature transmitter	Transmitting measured temperature signals
PT	Pressure transmitter	Transmitting measured pressure signals

**ملخص:** هندسة التحكم الصناعي أو هندسة السيطرة هي إحدى فروع الهندسة المبنية على نماذج رياضية لظواهر متعددة وتحليل أداؤها الديناميكي باستخدام نظريات التحكم لخلق متحكمات قادرة على جعل هذه الأنظمة تعمل بطريقة معينة. وللحصول على نظم التحكم، يتم استخدام تطبيقات الدوائر الكهربائية ومعالجات الإشارات الرقمية والمتحكمات الصغيرة، بالإضافة إلى أجهزة الاستشعار والأجهزة المتعلقة بتنفيذ عملية التحكم.

والهدف من عملنا هو تطبيق نظرية التحكم " تقسيم النطاق " للتحكم في صمامين للتنظيم باستعمال جهاز تحكم منطقي قابل للبرمجة الصناعية من نوع S7-300 على نموذج اولي لخزان ترحيل الغاز، قمنا بدراسته وانشائه في مخابر المعهد الوطني للبترول ببومرداس.

**كلمات دلالية:** هندسة التحكم الصناعي، هندسة السيطرة، نماذج رياضية، نظريات التحكم، تقسيم النطاق، صمام التنظيم، جهاز تحكم منطقي قابل للبرمجة الصناعية، خزان ترحيل الغاز.

**Résumé :** Régulation industrielle ou asservissement est une branche de l'ingénierie basée sur la modélisation mathématique pour divers phénomènes et d'analyser leurs performances en utilisant les théories des boucles de régulation pour créer des contrôleurs capables de rendre ces systèmes fonctionnent d'une certaine manière. Pour les systèmes de commande, des applications de circuit, des processeurs de signaux numériques et des microcontrôleurs sont utilisés, ainsi que des capteurs et des dispositifs de contrôle.

Le but de notre travail est l'application de la théorie du « Split-range » pour contrôler deux vannes régulatrices par l'utilisation d'un Automate Programmable Industriel de type S7-300 sur notre prototype d'un dégazeur qui nous avons étudié et mise en place dans les laboratoires de l'Institut national du pétrole Boumerdes.

**Mot clés :** régulation industrielle, asservissement, modélisation, boucles de régulation, split-range, Automate Programmable Industriel, dégazeur.

**Abstract:** Industrial regulation or Control engineering is a branch of engineering based on mathematical modeling for various phenomena and to analyze their performance using the theories of control loops to create controllers capable of making these systems work in a certain way. For control systems, circuit applications, digital signal processors and microcontrollers are used, as well as sensors and control devices.

The purpose of our work is the application of the "Split-range" theory to control two control valves by the use of a Programmable Logic Controller type S7-300 on our prototype of a degasser which we have studied and set up in the laboratories of the Algerian Petroleum Institute Boumerdes.

**Keywords:** industrial regulation, control engineering, mathematical modeling, control loops, split-range, Programmable Logic Controller, degasser.



# Glossaire

## Glossaire

---

**ACK** : Acquiescement des Alarmes.

**API** : Automate Programmable Industriel, en anglais PLC (Programmable Logic Controller).

**Blocs** : il structure le programme utilisateur en section indépendantes.

**C** : Consigne.

**CIS** : Complexe Industriel Sud.

**D** : Action dérivée.

**e** : Entrer.

**EBM** : Ecole de Boumerdes.

**Gr** : Gain Statique.

**GRAFCET** : (Grphe Fonctionnel de Commande des Étapes et Transitions) est un outil de représentation d'un système automatisé servant à expliquer le fonctionnement.

**I** : Action intégrale.

**IAP** : Institut Algérien du Pétrole.

**Kp** : Coefficient de l'action proportionnelle.

**LED** : (Light-Emitting Diode) Est un dispositif Optoélectronique capable d'émettre de la lumière lorsqu'il est parcouru par u courant électrique.

**M** : Mesure.

**Mnémoniques** : C'est une suite de caractères alphanumériques qui aide à se souvenir.

**MPI** : (Multi Point Interface) Est un réseau de communication de niveau cellule qui permet le raccordement des stations intelligentes (CPU, PC ...etc.). La communication entre les différentes stations est assurée par l'affectation des adresses M.P.I.

**Out** : Sortie.

**P** : Action proportionnelle.

**PIC** : Pressure Indicator Controller.

**PSI** : est une unité de mesure de contrainte et de pression anglo-saxonne (1 Bar  $\approx$  14,5 PSI).

**PV** : Process Variable (mesure)

**S** : Sortie.

**SIMATIC** : Est le nom d'un système d'automatisation (Logiciel/Matériel), qui a été développé par l'entreprise allemande siemens.

**SP** : Set Point (Consigne).

**Td** : Coefficient de l'action dérivée.

**Ti** : Coefficient de l'action intégrale.

---

## Glossaire

---

**TMI** : Transport, Maintenance et Instrumentation.

**UC** : Unité Centrale.

**UFR** : Unité de Formation et de Recherche.

$\Delta$  : période.

---