الجمـــهوريـة الجزائريـــة الديمقراطية الشعبيـة
La république algérienne démocratique et populaire
وزارة التـــعليم العـــــالي و البحث العلـــمي
Ministère de l'Enseignement Supérieur et les Recherches Scientifiques

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

معهد الصيانة و الأمن الصناعي
Institut de maintenance et sécurité industrielle

Département Maintenance en Instrumentation

Mémoire de fin d'étude

Pour l'obtention de diplôme Master

Filière : Génie Industriel

Spécialité : Ingénierie de la maintenance en instrumentation

Thème

Commande et supervision d'un système hydraulique à base API SIEMENS

Préparé par :

LAHCIENE Elhoucine et **HADJIMI Yasser**

Devant le jury composé de :

Nom et prénom	Grade	Etablissement	Qualité
Mr. BENARBIA Taha	MCB	IMSI-Univ .D'Oran 2	**Encadreur**
Mr.MEKKI Ibrahime El Khalil	MCB	IMSI-Univ .D'Oran 2	**Président**
Mr. BENFEKIR Abderrahim	MCB	IMSI-Univ .D'Oran 2	**Examinateur**

2022 / 2023

Dédicas

Louange au bon Dieu, le seul, l'unique et le tout puissant»

À ma très chère mère et père

À mes oncles

À mes frères

À mes sœurs

À toute la famille ***LAHCIENE***

À mes proches grands et petits

À tous mes amis

À tous ceux qui ont participé de près ou de loin dans la réalisation de ce travail

LAHCIENE Elhoucine

Dédicas

Louange au Bon Dieu, le seul, l'unique et le tout puissant»

À ma très chère mère et père

À mes oncles

À mes frères

À mes sœurs

À toute la famille ***HADJIMI***

À mes proches grands et petits

À tous mes amis

À tous ceux qui ont participé de près ou de loin dans la réalisation de ce travail

HADJIMI Yasser

Remerciements

En premier lieu, nous tenons à remercier notre **ALLAH**, notre créateur pour nous donner la chance d'étudier et la force pour accomplir ce travail.

Nous adressons nos vifs remerciements à nos parents qui nous aident.

Nous tenons à remercier en premier lieu et très chaleureusement notre encadrant **:** Dr. **BENARBIA Taha** pour avoir accepté de diriger notre travail, pour ses précieux Conseils, pour son esprit et pour sa disponibilité.

Nos remerciements s'adressent également aux membres de jury qui nous font l'honneur de juger notre travail.

Nous tenons à remercier tous ceux qui nous ont aidés de près ou de loin, directement ou indirectement à la réalisation de ce travail.

Enfin, nous voudrons exprimer notre reconnaissance à tous les amis et collègues qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.

Résume :

La contribution principale de ce travail de mémoire est de synthétiser un régulateur PID pour contrôler le débit et le niveau d'un système hydraulique. De plus, Nous avons proposé une solution de conception de système de commande et de supervision intégrant la commande PID avec une commande TOR pour un système hydraulique comprenant un réservoir et une chaudière, à base d'un automate programmable Siemens S7-300 et utilisant les logiciels STEP7 et WINCC.

En seconde contribution ce a travail résumé une étude générale sur les différents types de systèmes dynamiques ainsi que les méthodologies utilisées pour leurs commandes, ainsi que les outils mathématiques nécessaires pour leurs modélisations. En outre, une présentation générale des automates SIMATIC a été abordée, incluant une description de leur utilité et de leur fonctionnement, ainsi que quelques exemples de modèles d'automates.

Abstract :

The main contribution of this thesis work is to synthesize a PID controller to control the flow rate and level of a hydraulic system. Additionally, we proposed a solution for the design of a control and supervision system that integrates PID control with TOR control for a hydraulic system consisting of a reservoir and a boiler, based on a Siemens S7-300 programmable logic controller and using STEP7 and WINCC software. In a second contribution,

this work summarizes a general study on different types of dynamic systems and the methodologies used for their control, as well as the mathematical tools required for their modeling. Furthermore, a general presentation of SIMATIC programmable logic controllers was addressed, including a description of their usefulness and operation, along with some examples of PLC models

ملخص :

المساهمة الرئيسية لهذا العمل هي تخليص وتجميع متحكم PID للتحكم في تدفق ومستوى نظام هيدروليكي، بالإضافة إلى ذلك قمنا بتقديم حلاً لتصميم نظام تحكم ومراقبة يدمج التحكم PID مع التحكم TOR لنظام هيدروليكي يتضمن خزانًا وسخان الماء ، وذلك باستخدام وحدة تحكم برمجية قابلة للبرمجة من نوع Siemens S7- 300 واستخدام برامج STEP7 وWINCC.

كمساهمة ثانوية يلخص هذا العمل دراسة عامة حول مختلف أنواع الأنظمة الديناميكية والمنهجيات المستخدمة للتحكم فيها ، بالإضافة إلى الأدوات الرياضية اللازمة لنمذجتها. علاوة على ذلك تمت مناقشة عرض عام لوحدات التحكم SIMATIC ، بما في ذلك وصف فائدتها وعملها، بالإضافة إلى بعض أمثلة نماذج الوحدات المبرمجة.

SOMMAIR

Chapitre I : Etude général sur les déférentes classes des Systèmes

Chapitre II : Généralités sue les automates programmable industriel

Chapitre III :Régulation de débit et niveau d'un système hydraulique

Liste des figures

Liste des tableaux

Symboles et Abréviations

Gs : gain statique d'un procédé autorégulant.
k: gain dynamique d'un procédé intégrateur.
Q: débit.
e : constante de temps .
τ: temps mort ou retard pur .
n: ordre d'un système.
W : signal de consigne .
X : signal de mesure .
ε : signal d'écart mesure/ consigne.
Y: signal réglant (de sortie) d'un régulateur .
H(p) : fonction de transfert isomorphe du procédé .
C(p) : fonction de transfert du régulateur.
F(p) : fonction de transfert en chaîne fermée.
A (p): = C(p) .H(p) : fonction de transfert en chaîne ouverte .
H(jω) : fonction de transfert isochrone du procédé .
ω:pulsation (rad· s- 1 ou rad· min- 1).
QE : débit d'entrée
QS : débit de sortie
θ_{ve} : température *de*La vapeur entre dans l'échangeur.
θ_v :température *de*sortie .
U_e : La tension d'entrée .
U_s: La tension sortie.
R : résistance électrique.
L :bobine d'inductance.
LQ :la commande linéaire quadratique.
LQG : La commande linéaire quadratique gaussienne.
SED : Systèmes à Evénements Discrets.
RdP : Réseaux de Petri.
SDH :un système dynamique hybride .
PO: Partie Opérative.
PC: Partie Commande.
API: Automates Programmable Industriel.
BF : Boucle Fermée.
BO : Boucle Ouverte.
Kp: Coefficient de Gain Proportionnel.
P : Proportionnel.
PI : Proportionnel Intégral.
PID : Proportionnel Intégral Dérivé.
PLC : Programmable Logic Controllers.
TOR : Tout Ou Rien.
Ti : Temps d'intégration.
Td : Temps dérivation
DB : Blocs de données globaux.
FB : Bloc de fonction.
FC : Fonction.
HMI : Humann Machine Interface.
LIST : Le langage de liste d'instructions.
LOG : Le langage a base de logigramme.
MPI : Protocole de communication (Multi Point Interface).

OB : Bloc d'organisation.
PN/IE : Profin et /Industriel Ethernet.
LD: Ladder Diagram.
IL: Instruction List.
ST: Structurede texte.
RAM: Random Access Memory.
ROM: Read Only Memory.
S7: Step7.
GRAFCET: Graf de Commande Etapes-Transition.
SIMATIC: Siemens Automatic.
SM : Gamme des modules E/S des automates de Siemens.
SP : Sept Point.
WinCC : Windows Control Center.
AI : Analogique Input.
AO : Analogique Output.
COUNT : Schéma à Contact.
CPU Central Processing Unit.
DE : entrée digitale.
DI : Digital Input.
DO : Digital Output.
E/S : Entrée/Sortie.
FM : Function Module.
HW Config : HardWare Configuration.
IM : Coupleurs d'extension.
LOG: ports logiques.
OP : Operator Pupiter.
PC : Personnel Computer.
PG :la console de programmation sur le terrain.
PROFIBUS : Process Field Bus.
PS: Gamme des alimentations stabilisées de Siemens.
SFB : System Fonctionnel Bloc.
SFC : System Fonction.
SIMATIC : Siemens Automatic.

Introduction Générale

Automatisme est l'une des branches les plus avancées de l'ingénierie de nos jours. Son principal objectif est de doter les systèmes d'une certaine autonomie, leur permettant de prendre les décisions appropriées à différents stades de leur évolution. Cette science est considérée comme étant le pilier central de toutes les sciences de l'ingénierie, étant en interaction constante avec d'autres disciplines telles que l'informatique, l'électricité et la mécanique, ce qui lui a permis de faire des avancées majeures en profitant des progrès de chacune de ces disciplines. L'automatisation des systèmes est indispensable dans l'industrie moderne, car elle permet de réduire les coûts de main-d'œuvre, d'éviter les travaux dangereux et pénibles, d'assurer une meilleure qualité du produit, de réaliser des opérations impossibles à contrôler manuellement, de commander à distance pour améliorer les performances du système de production et d'améliorer la sécurité de l'installation industrielle et du personnel.

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

Dans ce contexte , nous présentons trois chapitres décrivant les éléments clés de notre projet comme suit:

Chapitre 01 : Consiste à présenter une étude globale des différents type des système en automatique. Dans ce contexte, la recherche portera sur les techniques de contrôle avancées employées dans les systèmes continus. D'ailleurs, a abordé–la modélisation des systèmes à événements discrets et les technique d'aide à l'analyse de ce type de systèmes. En outre, l'étude explorera la modélisation des systèmes hybrides, c'est-à-dire des systèmes présentant des comportements à la fois continus et discrets.

Chapiter 02 : ce chapitre a été consacré de présenter un état de l'art sur les API, en particulier sur les éléments et modules spécifiques de l'API SIEMENS S7. Les API SIEMENS S7 sont largement utilisés dans l'industrie pour leur fiabilité, leur facilité d'utilisation et leur flexibilité.

Chapiter 03 : Nous avons discuté dans ce chapitre la modélisation du système hydraulique, en mettant en évidence les éléments clés tels que les réservoirs, les vannes et les pompes. La recherche se concentrera également sur la programmation de régulation PID sous step7. concentrera sur la présentation de WINCC flexible, un outil de visualisation et de supervision qui permet de surveiller et de contrôler les systèmes automatisés en temps réel. La recherche explorera également la simulation de notre programme sous WinCC

Chapitre I

Etude général sur les déférentes classes des Systèmes en automatique

I .Introduction

Au XXe siècle, l'automatisation a connu un développement massif qui s'est articulé globalement autour de deux types de systèmes : les systèmes à événements discrets et les systèmes continus. Ces systèmes étaient mis en œuvre selon des méthodologies spécifiques et par des personnels aux origines diverses. Les systèmes à événements discrets, qui reposent sur une logique binaire de deux états (ouvert/fermé, marche/arrêt, sorti/rentré, etc.), sont traditionnellement gérés à l'aide de l'algèbre de Boole, développée par les logiciens anglais du XIXe siècle, et des méthodes états-transitions associées à des représentations graphiques telles que les graphes d'état, les réseaux de Petri ou encore les Grafcet. Les systèmes continus, quant à eux, sont composés d'éléments dont la mesure peut varier de façon continue (température, vitesse, niveau, etc.) et nécessitent des outils mathématiques tels que les équations différentielles, les transformations de Laplace ou de Fourier, et les méthodes d'état matricielles. Cette distinction entre les deux catégories de systèmes n'est pas absolue, car de nombreux ensembles industriels comprennent des éléments appartenant aux deux types de systèmes. Par exemple, les distributeurs et électrovannes sont généralement considérés comme faisant partie du monde des systèmes à événements discrets, mais des versions proportionnelles de ces mêmes équipements sont également utilisées dans le monde des systèmes continus.

I .1. Notion de System

Un système peut être défini comme un ensemble d'éléments exerçant collectivement une fonction déterminée. Il est soumis aux lois de la physique. Le système communique avec l'extérieur par l'intermédiaire de grandeurs, fonctions du temps, appelées signaux, entrées/sorties et soumis à des perturbations.**[01]**

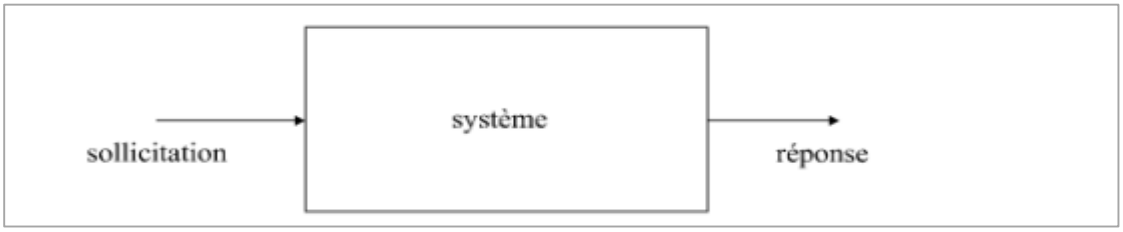


Figure 1.1 Modèle général d'un système.

Les systèmes peuvent être classés en plusieurs catégories**:**

1.2.Systèmes continus

Les systèmes continus sont des systèmes physiques, tels que les systèmes mécaniques, électromécaniques ou thermodynamiques, qui peuvent être modélisés par des équations différentielles continues. Ces systèmes ont des variables d'état continues qui varient dans le temps et qui peuvent être décrites mathématiquement par des équations différentielles ordinaires ou des équations différentielles partielles. . Les systèmes continus peuvent être analysés à l'aide de la théorie des systèmes dynamiques, qui étudie la façon dont les variables d'état évoluent dans le temps en fonction des entrées et des conditions initiales.

Les méthodes d'analyse des systèmes dynamiques peuvent être utilisées pour comprendre le comportement des systèmes continus.**[02]**

I .2.1.Exemples Systèmes continus

a/ Premier exemple : moteur à courant continu

Un moteur à courant continu (**MCC**), dont le schéma de principe est donné à la **Figure 1.2** , est un dispositif électromécanique qui convertit une énergie électrique d'entrée en énergie mécanique. L'énergie électrique est apportée par un convertisseur de puissance qui alimente le bobinage disposé sur l'induit mobile (**rotor**). Ce bobinage est placé dans un champ magnétique, permanent ou non, produit par l'inducteur. On supposera pour simplifier que cette excitation est séparée et constante, comme c'est le cas, notamment lorsque l'inducteur est constitué d'aimants. Le courant circulant dans les spires de l'induit du moteur, des forces électriques lui sont appliquées et, grâce à un dispositif adapté(**balais** et **collecteur**), les forces s'additionnent pour participer à la rotation. On peut ainsi considérer le moteur comme un système dont l'entrée est la tension d'induit et la sortie une grandeur liée à la position angulaire du rotor. On choisit tout d'abord la vitesse de rotation du rotor comme grandeur de sortie.

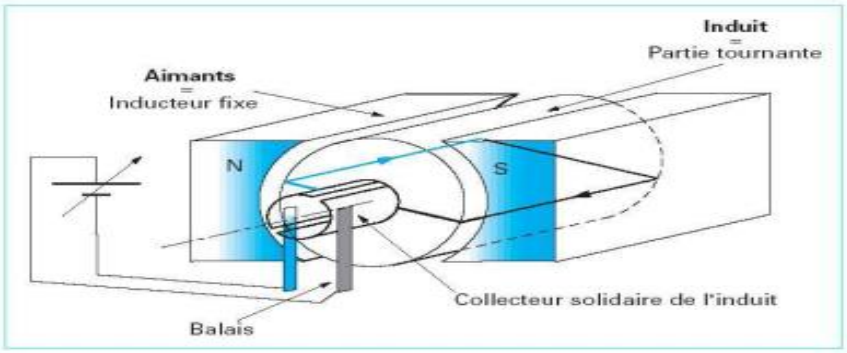


Figure 1.2: Principe de fonctionnement d'un moteur à courant continu

b/ Deuxième exemple: régulateur de niveau

On considère le système de régulation de niveau de liquide [**Oga10**, **WJK02**] représenté à la**Figure 1.3**. Un réservoir est alimenté en liquide et le niveau réglé par l'intermédiaire de deux vannes placées d'une part dans la canalisation d'entrée et d'autre part à la sortie du réservoir. On supposera que la vanne de sortie est laissée dans une position donnée et que la seule action sur le système résulte du réglage de la vanne d'entrée.

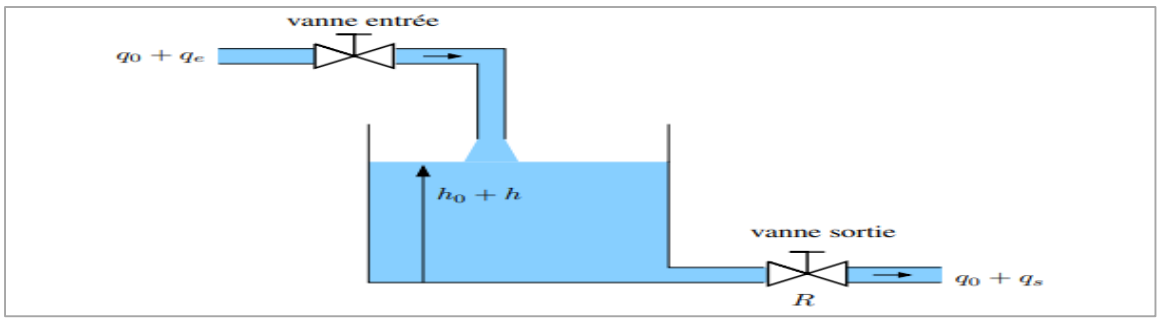


Figure 1.3: Schéma d'un régulateur de niveau de liquide

c/ Troisième exemple: échangeur thermique

On considère le cas d'un échangeur thermique **[FPEN14].** Le système échange de l'énergie thermique entre de la vapeur circulant dans l'enceinte de l'échangeur et de l'eau circulant dans une tuyauterie qui serpente dans l'enceinte de l'échangeur, comme représenté sur la**Figure 1.4**

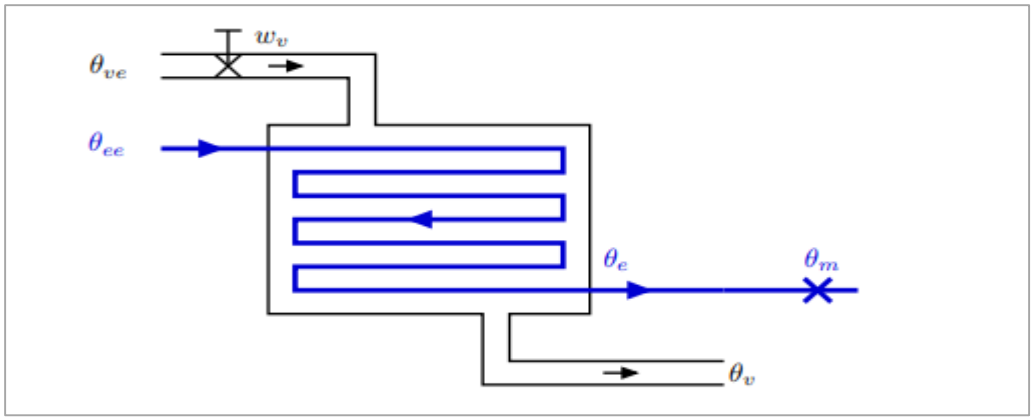


Figure 1.4: Schéma d'un échangeur thermique.

La vapeur entre dans l'échangeur à la température θ_{ve}et en sort à la température θ_v. L'eau, quant à elle, entre à la température ambiante θ_{ee}et ressort à la température θ_e. Le débit de vapeur est réglé par une vanne placée dans le tuyau d'arrivée de vapeur.**[03]**

I.3.Les outillés et les techniques utilisés pour modéliser les systèmes continus

I .3.1. Modélisation par les équations différentielle

a/ Système du premier ordre

Exemple : Le réservoir.

Un débit Q_e alimente un réservoir la **Figure 1.5**. Une pompe volumétrique soutire un débit constant Q_sOn cherche la relation entre H(t) et Q_e (t) . La section transversale S du réservoir est de 0,25m^2 .
Les conditions initiales sont : **H = 1m, Q_{e0} = Q_{s0}= 0,02 $m^3.s^{-1}$**. À **t = 0**, on augmente le débit Q_e d'une variation q_e**= 0,001 $m^3.s^{-1}$**.

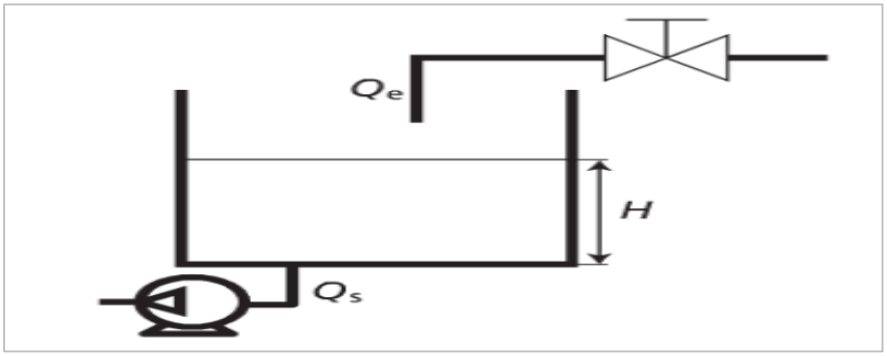


Figure 1.5: Niveau dans un réservoir

On exprime la variation de volume $d_v$de liquide dans le réservoir en un temps dt, considéré comme très petit, en fonction des débits Q_e , $Q_s$et dt.

dv = (Q_e - Q_s) dt

Comme la variation de volume est **dv = Sdh**, on obtient **:**
L'équation différentielle décrivant le système étudié est donc **:**

$$\frac{dh}{dt} = \frac{1}{S}(Q_e - Q_s) = \frac{1}{S}(Q_{e0} + q_e - Q_{s0}) = \frac{1}{S}q_e \quad \text{Soit} \quad \frac{dh}{dt} = \frac{0.001}{0.025} = 0.004\ m.s^{-1} \qquad (\mathbf{I.1})$$

Le coefficient S est constant (il ne dépend pas du temps) et l'ordre le plus élevé de la dérivée sur la sortie h (t) du système est 1. L'équation différentielle de ce système est donc linéaire et d'ordre 1 : c'est ici l'équation la plus simple que l'on puisse trouver. Si l'on intègre cette équation différentielle on obtient:

$$H(t) = H + \frac{q_e}{S}t \ \ soit \ \ H(t) = 1 + 0.004t \qquad (\mathbf{I.2})$$

La réponse H(t) obtenue, montrée à la **Figure 1.6** dépend bien sûr des conditions initiales et de la manière dont varie l'entrée, ici Q_e (t).

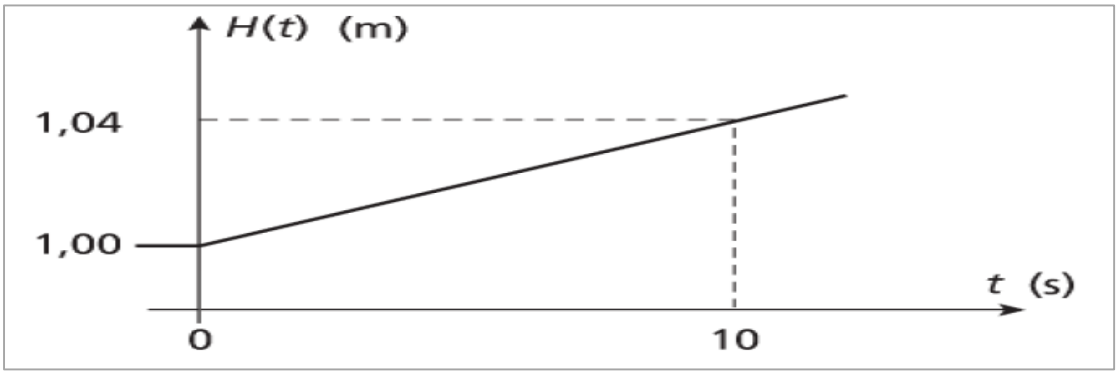


Figure 1.6: Évolution de H(t) après variation de Q_e (t)

b/ système du deuxième ordre

Exemples: du Circuit électrique RLC

Le montage électrique présenté ici est un circuit très classique d'étude. Il est composé d'une résistance électrique *R,* d'une bobine d'inductance *L,* d'un condensateur de capacité C (**Figure 1.7**). La tension d'entrée $U_e(t)$est une tension continue et le signal de sortie est la tension $U_s(t)$. Les conditions initiales sont: $U_e(0) = U_s(0)$= 2 V et$I(0)$= 50mA. Les variations sont donc **:**

$U_e(0) = U_s(0)$= 0et $(0) = 0$. On cherche à déterminer la variation de tension u_s *(t)* en fonction de la variation tension d'entrée $U_e(t)$.

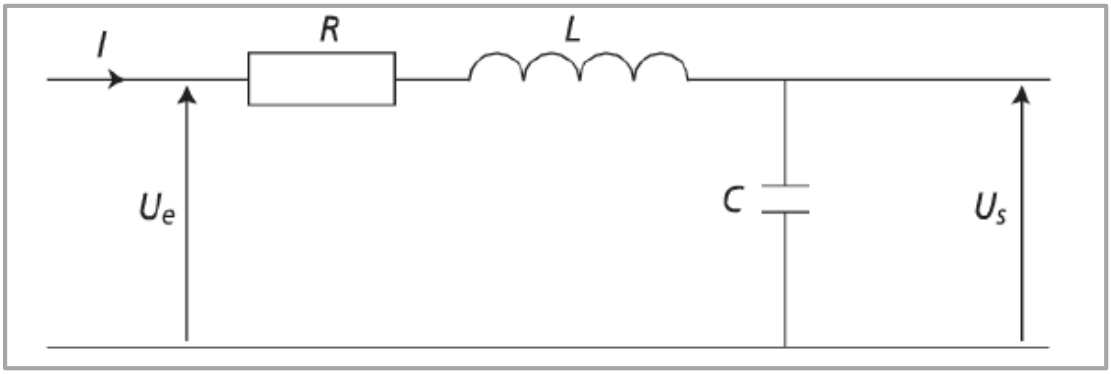


Figure 1.7: Circuit résistance - inductance - capacité : circuit RLC.

La loi des mailles permet d'écrire:$\boldsymbol{U_e(t) = U_R(t) + U_L(t) + U_c(t)}$. **(I.3)**

Soit $\boldsymbol{u_e(t) = Ri(t) + l\frac{di}{dt}(t) + u_c(t)}$ avec $\boldsymbol{i(t) = C\frac{du_c}{dt}(t) = C\frac{du_s}{dt}(t)}$. **(I.4)**

On obtient alors l'équation différentielle de ce système :**[04]**

$\boldsymbol{u_e(t) = LC\frac{du^2_s}{dt^2}(t) = RC\frac{du_s}{dt}(t) + u_s(t)}$

Les tensions et le courant sont stabilisés et constants avant $t = 0$: les conditions initiales pour les variations sont donc nulles, et la fonction de transfert de ce système peut s'écrire

$$\boldsymbol{F(p) = \frac{U_s(p)}{U_e(p)} = \frac{1}{LCp^2 + RCp + 1} = \frac{1}{\frac{1}{\omega 0^2}p^2 + \frac{2\lambda}{\omega} + 1}} \quad \textbf{(I.5)}$$

Le gain statique est $G_s = 1$; cela signifie, qu'en régime permanent, la variationde la tension de sortie sera la même que celle de l'entrée. Le gain statique est ici sansunité puisqu'il correspond à un rapport de deux tensions. En identifiant terme à terme on a:

$$\boldsymbol{\omega_0 = \frac{1}{\sqrt{LC}} \text{ et } \lambda = \frac{R}{2}\sqrt{\frac{C}{L}} = \frac{R}{2L\omega_0}}$$

c/ Système plus complexe :

Nous avons vu qu'un système global est constitué de plusieurs composants. Le diagramme fonctionnel du système global conduit à combiner les différentes équations de chaque composant et on aboutit alors à une nouvelle équation différentielle d'ordre supérieur mettant en jeu l'entrée et la sortie du système. D'une manière générale, un système linéaire continu invariant peut être modélisé par une équation différentielle d'ordre n qui s'écrit sous la forme :

$$\boldsymbol{a_n\frac{d^{n-1}s(t)}{dt^n} + a_{n-1}\frac{d^{n-1}s(t)}{dt^{n-1}} + .. + a_1\frac{ds(t)}{dt} + a_0 s(t) = b_m\frac{d^m e(t)}{dt^m} + b_{m-1}\frac{d^{m-1}e(t)}{dt^{m-1}} + .. + b_1\frac{de(t)}{dt} + b_0 e(t)} \quad \textbf{(I.6)}$$

Pour poursuivre l'analyse des SLCI, et définir une modélisation performante du système global, deux remarques sont nécessaires:

- On veut obtenir la réponse du système pour une entrée quelconque afin d'analyser les performances globales du système.
- On doit pouvoir prendre en compte dans le modèle global la modification de modélisation d'un composant.**[03]**

I.3.2.modélisation par Transformée de Laplace et Fonctions de transfert

La transformée de Laplace est une technique mathématique utilisée pour convertir une équation différentielle en une équation algébrique. Cela facilite la résolution de l'équation, car les équations algébriques sont souvent plus faciles à manipuler que les équations différentielles.

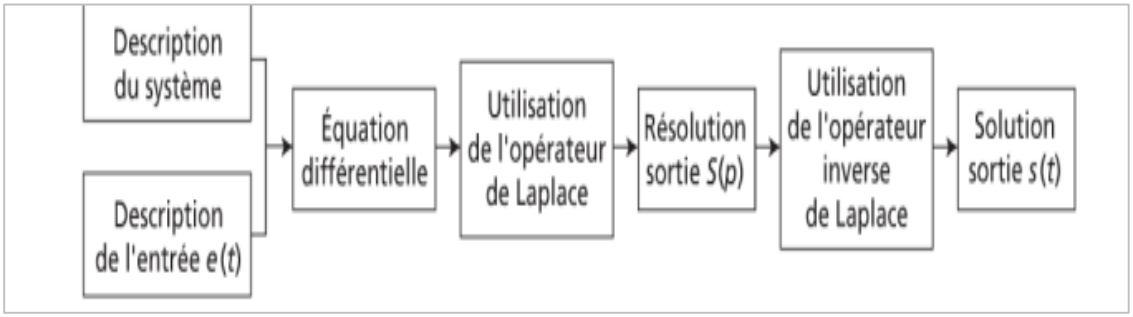


Figure 1.8: Méthode de résolution par la transformée de Laplace.

Exemple

Reprenons l'exemple d'étude du réservoir décrit par la **Figure 1.5**. Cherchons à représenter le système par sa fonction de transfert. On détermine la transformée de Laplace de l'équation différentielle du système soit :

$$\mathcal{L}\left[\frac{dh}{dt}(t) = \frac{1}{S}q_e(t)\right] = \left[pH(p) = \frac{1}{S}Q_e(p)\right] \qquad \textbf{(I.7)}$$

La fonction de transfert est donc :

$$\frac{H(p)}{\mathrm{Q_e(p)}} = \frac{1}{Sp} \; Soit \; \frac{H(p)}{Q_e(p)} = \frac{1}{0.25\,p} = \frac{4}{p} \qquad (\mathbf{I.8})$$

Si on cherche la variation de h(t) lorsque le débit d'entrée varie brusquement de $q_e(t)$, il suffit d'exprimer la transformée de $q_e(t)$et de déterminer h(t) à l'aide de la transformée inverse ,
On a donc:

$$\mathcal{L}[q_e(t)] = \mathcal{L}[0.001\,u(t)] = Q_e(p) = \frac{0.001}{p} \qquad (\mathbf{I.9})$$

La variation du p niveau est alors :

$$H(p) = \frac{4}{p}Q_e(p) = \frac{0.004}{p^2} \qquad (\mathbf{I.10})$$

La transformée inverse $\mathcal{L}^{-1}[H(p)]$

donne **h(t) = 0,004 t**. Comme on sait que le niveau est H_0 **= 1**m à **t = 0**.on obtient **H(t) = 0,004 t + 1.**

I.4. Les commandes avancées des systèmes continus

Le contrôle des systèmes continus implique l'utilisation des commandes pour réguler le comportement du système et atteindre des objectifs spécifiques

I.4.1. Commande PID

PID est un nom communément donné au contrôle à trois termes. L'acronyme PID fait référence aux premières lettres des noms des termes individuels qui composent le régulateur à trois termes standard. Ceux-ci sont P pour le terme proportionnel, I pour le terme intégral et D pour le terme dérivé dans le régulateur. Les régulateurs à trois termes ou PID sont probablement les régulateurs industriels les plus largement utilisés. Même les systèmes de contrôle industriels complexes peuvent comprendre un réseau de contrôle dont le bloc de construction principal est un module de contrôle PID. Le régulateur PID à trois termes a une longue histoire d'utilisation et a survécu aux changements technologiques de l'ère analogique à l'ère des systèmes de contrôle par ordinateur numérique de manière assez satisfaisante. C'était le premier (et le seul) régulateur à être produit en masse pour le marché à grande échelle qui existait dans les industries de transformation.**[05].**

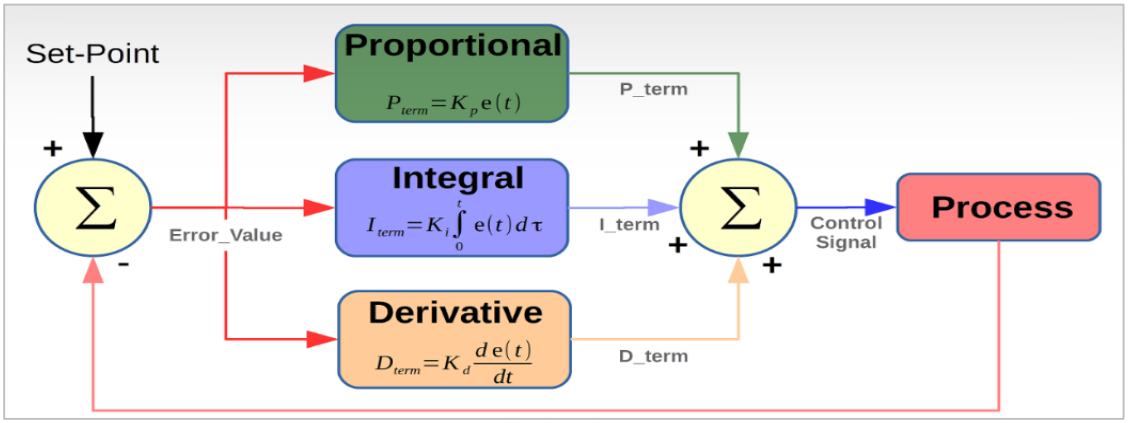


Figure 1.9: Asservissement avec régulateur PID

I.4.2. Commande prédictive

La commande prédictive traditionnelle implique la conception d'un système de contrôle qui assure, en boucle fermée, des performances les plus proches de celles désirées pour garantir la sureté de fonctionnement. Elle permet l'incorporation du système de contraintes dans le problème d'optimisation. La stratégie de la commande prédictive génère donc des signaux de commande à partir de la résolution du problème d'optimisation. Ce dernier doit être optimisé en ligne et s'articule sur le modèle du processus ainsi que les mesures réelles, qui sont intégrées à chaque pas d'échantillonnage dans le problème d'optimisation afin d'assurer une structure de contrôle bouclée.

Les éléments suivants représentent les concepts de base de la stratégie de commande Prédictive :

- Un modèle du système permet la prédiction des signaux qui agissent sur les performances du système en respectant les contraintes sur ces signaux.
- Connaître la trajectoire à suivre sur l'horizon de prédiction.
- Un critère d'optimisation ; généralement quadratique et qui se porte sur l'erreur entre les sorties prédites et désirées, qui sera pondéré par le signal de contrôle.

➢ Un solveur permet de trouver une solution optimale au problème d'optimisation en respectant les contraintes, et qui sera implémenté en temps réel. A mesure que le temps avance, le problème d'optimisation sera actualisé avec l'état du système (les nouvelles mesures).

➢ En fixant un horizon fini de commande Nc et un horizon de prédiction Np (avec $N_p < N_c$) et en mettant l'état présent comme état initial, et après optimisation de la fonction du coût sur l'intervalle de prédiction Np, tout en respectant les contraintes entrées sorties du système, on obtient une séquence de N commandes dont juste le premier élément est appliqué. Ainsi, à la prochaine période d'échantillonnage on répète la même procédure.**[05]**

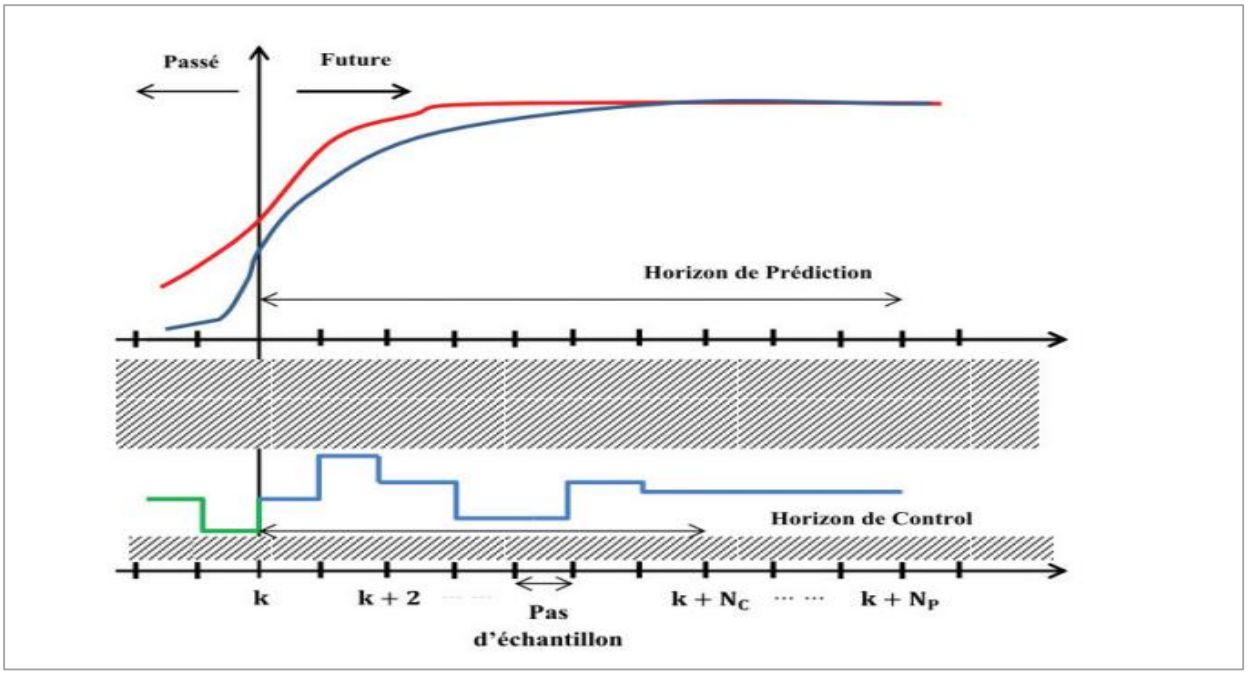


Figure 1.10: Principe de la commande prédictive

I.4.3. Commande adaptative

la commande adaptative regroupe un ensemble de concepts et de techniques permettant d'ajuster automatiquement en temps réel les régulateurs mis en œuvre dans la commande d'un processus lorsque les paramètres de ce processus sont difficiles à déterminer ou varient avec le temps.
On distingue deux approches principales de commande adaptative :
- La commande adaptative directe (**Figure b**) dans laquelle les paramètres du régulateur sont ajustés directement et en temps réel à partir de la comparaison entre les performances réelles observées et les performances désirées.
- La commande adaptative indirecte (**Figure a**) qui suppose une estimation des paramètres du processus par une procédure d'identification récursive, ce type de commande adaptative tient compte des caractéristiques d'évolution du système afin d'auto-ajuster le régulateur.
- Les schémas de commande adaptative peuvent fonctionner en régime d'auto ajustement ou en régime adaptatif. Dans le premier cas, l'opération d'adaptation est déclenchée lors d'une dégradation des performances et elle s'arrête une fois que les performances souhaitées sont a atteintes. C'est le cas de ce que nous proposons dans cette thèse, puisque nous supposons que le système commandé peut fonctionner à différents niveaux de performances prédéfinis, chaque niveau de performances correspond à une consommation propre, et à selon les besoin, nous demanderons au système de passer d'un niveau de performances à l'autre. Le processus de

contrôle s'arrête quand le niveau de performances demandé est atteint. Dans le deuxième cas, l'algorithme d'adaptation est actif en permanence.**[05]**

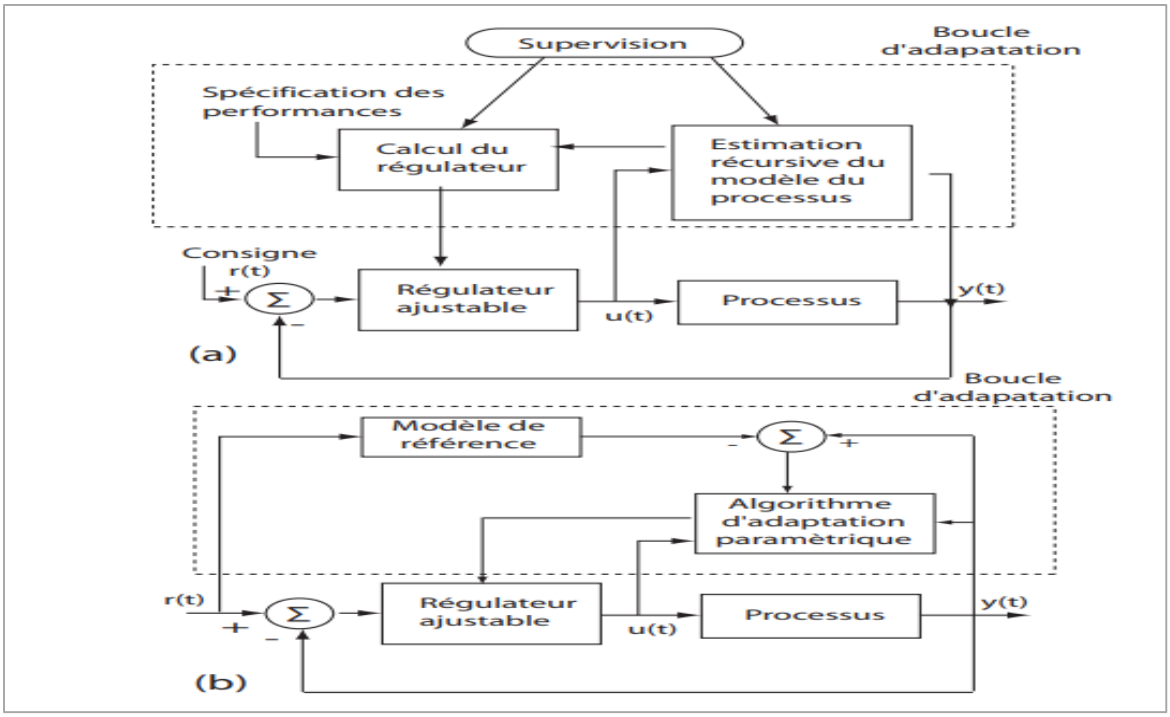


Figure 1.11: Commande adaptative en boucle fermée (schéma de principe):（a) Commande adaptative indirecte (b) Commande adaptative directe.

I.4.4. La commande optimale

La commande optimale est une stratégie de contrôle qui consiste à trouver la meilleure façon de réguler un système en utilisant des méthodes mathématiques et de modélisation. L'objectif est de minimiser un critère de performance spécifié tout en satisfaisant des contraintes opérationnelles.
L'optimalité peut être définie de plusieurs façons. Fondamentalement, c'est le critère de performance soumis à l'optimisation qui définit la performance réalisable et l'optimalité. Il existe différentes méthodes d'optimisation et de critères de performance. Par exemple, la commande linéaire quadratique (LQ) vise à minimiser une combinaison d'erreurs de commande pondérées et d'actions de commande où les poids sont des paramètres de conception de la commande. En conséquence, le contrôleur optimal fournit des performances qui minimisent le critère de performance donné pour les poids choisis ,La commande linéaire quadratique gaussienne (LQG) est une extension stochastique de la commande LQ déterministe. La commande LQG fournit un contrôle optimal de manière similaire à la commande LQ, mais elle fonctionne mieux que la commande LQ lorsqu'elle est confrontée à des perturbations de processus et du bruit de mesure.**[02]**
En LQG, la nature des perturbations et du bruit est capturée en utilisant les variances des perturbations et du bruit. La commande optimale est applicable aux processus multi variables et interactifs où la performance optimale peut être facilement définie et atteinte en termes de critères de performance. Les inconvénients évidents de la commande optimale sont la sensibilité du modèle de processus et la maintenance délicate lorsqu'elle est confrontée à des changements d'instrumentation, de dispositif de commande ou d'autres matériels de processus.

Cependant comme la commande optimale est plus ou moins cachée dans presque toutes les méthodes de contrôle avancées, ces inconvénients s'appliquent généralement à tous les types de contrôle avancé. Dans sa forme la plus générale, la commande optimale permet de prendre en compte explicitement les objectifs ultimes de la commande (optimisation économique), les contraintes, les non linéarités et l'aspect dynamique, ce qui est particulièrement intéressant pour les procédés discontinus. La formulation du problème est assez intuitive, ce qui facilite l'acceptation et la formulation des objectifs par des non spécialistes. Les principales limites de la commande optimale sont la difficulté de la mise en œuvre, l'applicabilité aux procédés lents uniquement et la nécessité de disposer d'un modèle dynamique du procédé suffisamment précis et rapide à simuler, pour rendre l'optimisation en temps réel efficace.**[06]**

I.5. Systèmes à évènements discrets

Un système à évènements discrets est un système dont l'espace d'état est discret. Les transitions d'état sont provoquées par l'occurrence d'événements. L'ensemble des événements est également discret. Un événement est validé si l'ensemble des conditions nécessaires à son occurrence sont vérifiées. Pour prendre en compte le temps, des temporisations peuvent être associées aux événements, une temporisation est un laps de temps qui doit être écoulé avant qu'un événement validé puisse survenir. Plusieurs outils mathématiques ont été proposés pour modéliser les systèmes à événements discrets on cite par exemple : les file d'attente, algèbre dioïdes, les réseaux de Petri et les Grafcet pour synthèse des commande TOR mais sont limitées.**[07].**
Un exemple courant de système à événements discrets est une chaîne de montage de fabrication automatisée, où chaque étape du processus est déclenchée par un événement discret, tel que l'achèvement d'une tâche ou l'arrivée d'un nouveau composant.

I.5.1. Exemple de système à événements discrets

I.5.2. Système de communication

Les clients d'un système de communication sont appelés messages, paquets ou appels (comme dans un réseau téléphonique). Un message est généralement généré par un utilisateur situé à un point de «source » et destiné à atteindre un autre utilisateur situé à un point de « destination ». Souvent, la connexion entre la source et la destination n'est pas directe, mais passe par un ou plusieurs points intermédiaires. Pour que les messages puissent voyager de la source à la destination, ils doivent accéder à divers serveurs, qui dans ce cas sont des équipements de commutation tels que des commutateurs téléphoniques simples ou des processeurs informatiques assez sophistiqués, ainsi que des supports de communication tels que des fils ou des ondes radio.**[07]**

I .5.3. caractéristique de systèmes de communication

Une caractéristique importante des systèmes de communication est la nécessité de mécanismes de contrôle qui garantissent que l'accès aux serveurs est accordé de manière équitable et efficace, et que les objectifs du processus de communication sont atteints (c'est-à-dire qu'un message est effectivement transmis avec succès à sa destination). Ces mécanismes de contrôle, parfois appelés protocoles, peuvent être assez complexes. Ainsi, leur conception et le processus de vérification de leur bon fonctionnement posent des problèmes difficiles.**[07]**

a/ Exemple 1

Un exemple de systèmes de transport en tant que système à événements discrets pourrait être un système de feux de circulation. Le système fonctionne sur la base d'événements discrets, tels que l'arrivée de véhicules à une intersection, et déclenche des actions telles que le changement des feux de signalisation pour permettre un mouvement sûr et efficace du trafic.

b/ Exemple 2

Systèmes de fabrication

Les clients d'un processus de fabrication sont des pièces de production. Les pièces concurrencent pour accéder à différents serveurs, qui dans une usine typique sont des machines effectuant des opérations spécifiques et des dispositifs de manipulation de matériaux tels que des robots et des convoyeurs. Lorsque les pièces ne sont pas en cours de traitement, elles sont stockées dans des tampons jusqu'à ce que le serveur requis pour la prochaine opération soit disponible. En raison de limitations physiques réelles, les tampons dans un système de fabrication ont généralement des capacités finies. Un exemple simple est présenté dans la **Figure 1.12**. Les pièces passent par deux machines, où la capacité du tampon devant la deuxième machine est limitée à deux. Par conséquent, il est possible qu'une pièce qui a fini son service à la machine 1 constate que la machine 2 est occupée et que les deux emplacements du tampon sont occupés. Dans ce cas, la pièce doit rester à la machine 1 même si elle n'a plus besoin de service là-bas ; de plus, les autres pièces en attente d'accéder à la machine 1 sont obligées de rester en file d'attente. Cette situation est appelée blocage L'ensemble d'événements pour cet exemple est le suivant**:**
A: est une arrivée depuis le monde extérieur vers la première machine.
C1 : est l'achèvement d'un service à la première machine.
D2 : est un départ de la deuxième machine.**[08]**

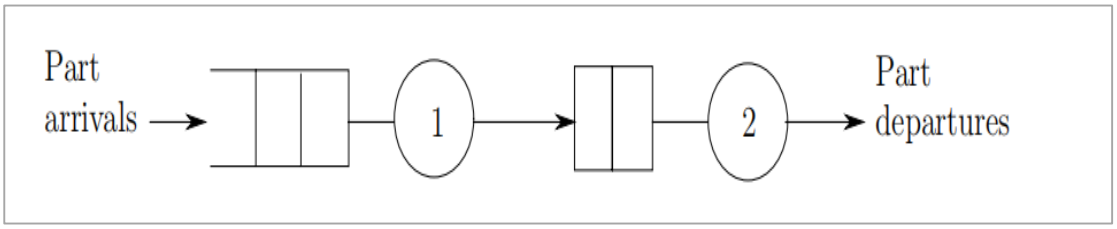


Figure 1.12 :Modèle de file d'attente d'un système de fabrication.

b/ Exemple 3

Systèmes de circulation routière

Dans un environnement de circulation routière, les véhicules utilisent des serveurs tels que les feux de circulation, les guichets de péage et l'espace physique sur les routes. À titre d'exemple, considérons comment une intersection en T simple (Fig.) peut être considérée comme un système d'événements discrets. Il existe quatre types de véhicules :
(**1,2**) les véhicules venant du point 1 et tournant à droite vers 2,
(**1,3**) les véhicules venant du 1 et tournant à gauche vers 3,
(**2,3**) les véhicules allant tout droit de 2 à 3.
(**3,2**) les véhicules allant tout droit de 3 à 2

Le feu de circulation est réglé de sorte qu'il passe soit au rouge pour les véhicules (**1,2**) et (**1,3**) (vert pour les véhicules (**2,3**) et (**3,2**)), soit il passe au vert pour les véhicules (**1,2**) et (**1,3**) (rouge pour les véhicules (**2,3**) et (**3,2**)).
l'ensemble d'événements :**E =** {**a12,a13,a23,a32,d12,d13,d23,d32,g,r**}
a12, **a13**, **a23**, **a32** correspondent à l'arrivée d'un véhicule pour chacun des quatre types, **d12**, **d13**, **d23**, **d32** correspondent à un départ de véhicule après avoir traversé l'intersection pour chaque type,
G indique que le feu passe au vert pour les véhicules (**1,2**) et (**1,3**), R indique que le feu passe au rouge pour les véhicules (**1,2**) et (**1,3**).**[08]**

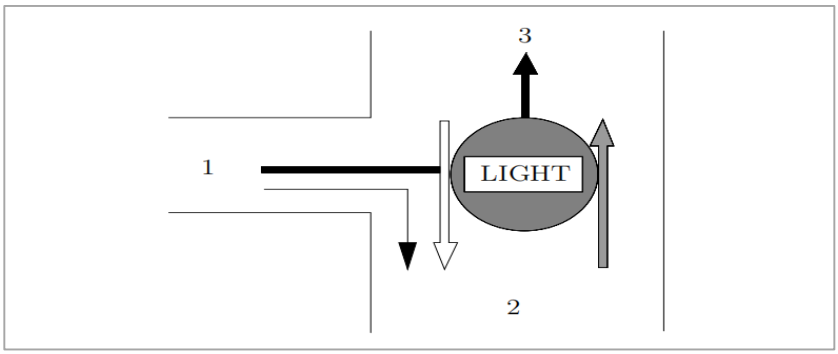


Figure 1.13 : Une intersection simple contrôlée par un feu de circulation

I.6. Modélisation des systèmes à événements discrets

I.6.1. Réseau de Petri

Les Réseaux de Petri (RdP) constituent un formalisme graphique propre à la modélisation des Systèmes à Evénements Discrets (SED) introduit en 1962 par Carl Adam Petri [Petri 1962].Ils sont particulièrement adaptés à l'étude des processus complexes mettant en jeu des propriétés de synchronisme et de partage de ressources. Leur support mathématique a permis en outre d'établir de nombreux résultats analytiques. Pour l'étude des SED dans l'algèbre des diodes, les RdP sont souvent utilisés comme un outil de modélisation intermédiaire.**[05]**

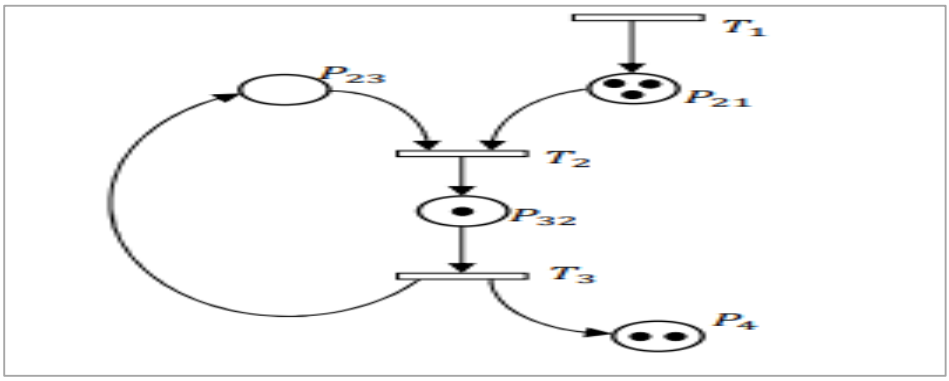


Figure 1.14:Exemple de réseau de Petri

b/ Définition : (marquage)

Chaque place contient un nombre entier positif ou nul de marquage (jeton). Le marquage M définit l'état du système décrit par le réseau a un instant donné. C'est un vecteur colonne de dimension m (le nombre de places dans le réseau). La $j^{ème}$ composante de ce vecteur représente le nombre de marque dans la$j^{ème}$ place du réseau. On note : M0 marquage initial**.[07]**

c/ Définition : (Réseaux de Petri P-temporisés)

Pour chaque couple de transition $\boldsymbol{T_i}$, Ton note $\boldsymbol{p_{ij}}$ la place qui relie la transitiont_i à $\boldsymbol{t_i}$ .nSicetteplace$\boldsymbol{p_{ij}}_i$ existe, la temporisation(en valeur rationnelle positive) Correspondant tees notée $\boldsymbol{r_{ij}}$etson marquage noté$\boldsymbol{m_{ij}}$.La plus grande temporisation du graphe d'évènement considéré est notée $\boldsymbol{\tau^{max}}$,définie par: $\boldsymbol{\tau^{max}} = \boldsymbol{m\{\tau_{ij}\}\frac{ij}{p_{ij}} \in p}$.

d/ RdP synchronisé

Un ensemble d'événements externes est associé au RdP, ces événements permettent le franchissement de certaines transitions. Un tel RdP est dit synchronisé. Considérons le RdP modélisant la machine décrite ci-dessous. On associe à ce RdP l'ensemble d'événements A, D, S où A désigne l'événement « Arrivée pièce », D l'événement « Démarrage service », S l'événement « Sortie pièce ». La figure représente le système modélisé par un RdP synchronisé.

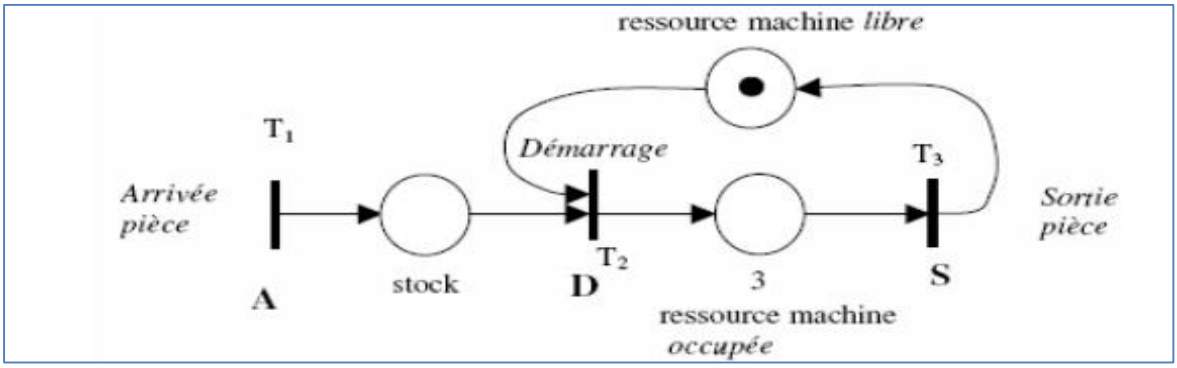


Figure 1.15: Exemple sur un RDP synchronise

Le tir de la transition T1 est lié à l'occurrence de l'événement A.
- A la validation de la transition, matérialisée par la présence d'au moins un jeton dans la place « stock » et d'un jeton dans la place « ressource machine libre » .
- Au démarrage effectif du service (occurrence de l'événement D).
- Le tir de la transition T3 est lié à l'occurrence de l'événement S.**[7]**

I.6.2. Commande par Grafcet

Le GRAFCET (Graphe Fonctionnel de Commande par Etapes et Transitions) ou SFC (Séquentiel Fonction Chart) est un outil graphique qui décrit les différents comportements (cahier des charges) de l'évolution d'un automatisme et établit une correspondance à caractère séquentiel et combinatoire entre :

– Les entrées : c'est-à-dire les transferts d'informations de la Partie Opérative vers la Partie Commande.

– Les sorties : transferts d'informations de la Partie Commande vers la Partie Opérative. C'est un outil graphique puissant, directement exploitable, car c'est aussi un langage pour la plupart des API existants sur le marché. Lorsque le mot GRAFCET (en lettre capitale) est utilisé, il fait référence à l'outil de modélisation. Lorsque le mot grafcet est écrit en minuscule, il fait alors référence à un modèle obtenu à l'aide des règles du GRAFCET. Le GRAFCET comprend **:**

– **Etape initiale :** représente une étape qui est active au début du fonctionnement. Elle se différencie de l'étape en doublant les côtés du carré.

– Transition : la transition est représentée par un trait horizontal.

– Réceptivité : les conditions de réceptivité sont inscrites à droite de la transition.

– Etape : chaque étape est représentée par un carré repéré numériquement

– Action(s) : elles sont décrites littéralement ou symboliquement à l'intérieur d'un ou plusieurs rectangles reliés par un trait à la partie droite de l'étape.

– Liaisons orientées : indique le sens du parcours.**[04]**

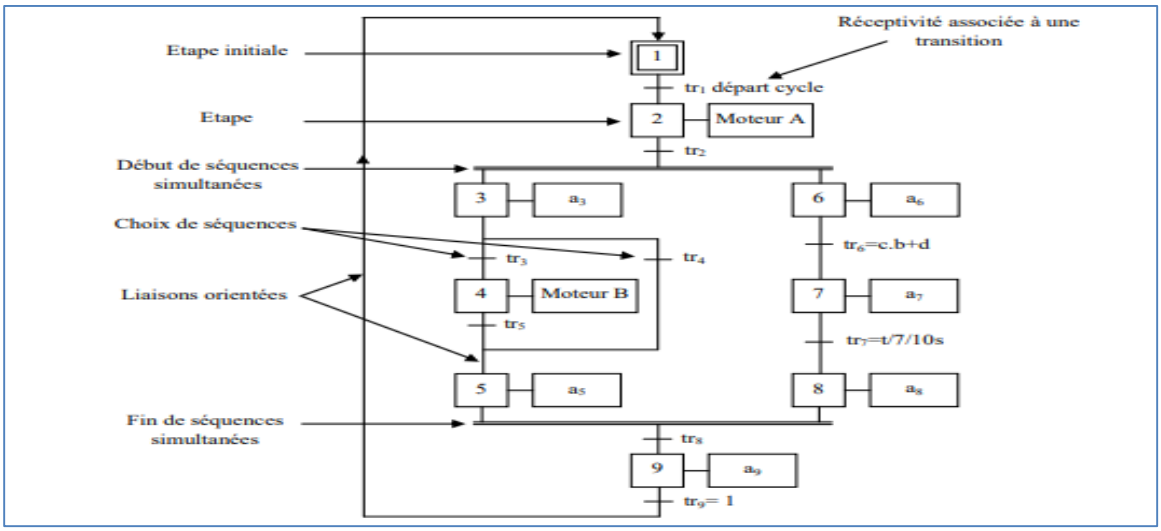


Figure 1.16: Représentation d'un modèle type de GRAFCET.

I.6.2.1. Description du GRAFCET

La description du comportement attendu d'un automatisme peut se représenter par un GRAFCET d'un certain « niveau ». La caractérisation du niveau du GRAFCET nécessite de prendre en compte trois dimensions **:**

-Le point de vue : caractérisant le point de vue selon lequel un observateur s'implique dans le fonctionnement du système pour en donner une description. On distingue trois points de vue :

- Un point de vue système : C'est un graphe qui décrit le fonctionnement global du système. Il traduit le cahier des charges sans préjuger de la technologie adoptée. Il permet de dialoguer avec des personnes non spécialistes (fournisseurs, décideurs ...) Son écriture, en langage clair, permet donc sa compréhension par tout le monde.

-Un point de vue Partie Opérative : Dans ce type on spécifie la technologie de la PO ainsi que le type de ses informations reçues (ordres) et envoyées (comptes-rendus).

-Un point de vue Partie Commande : Ce type est établi la technologie des éléments de dialogue : entre PC et PO, PC et opérateur, PC et autre système. C'est la version qui permet d'établir les équations des schémas de réalisation (électrique, pneumatique).

Exemple : Unité de perçage.

Les perçages sont effectués en même temps après action sur un bouton poussoir Départ Cycle.

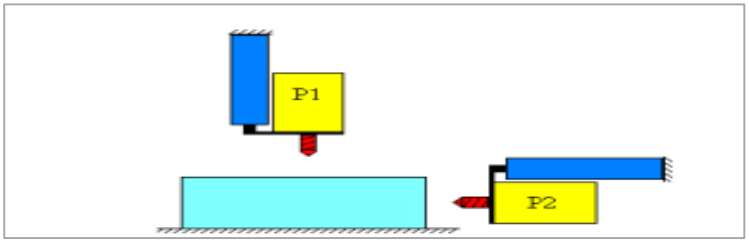
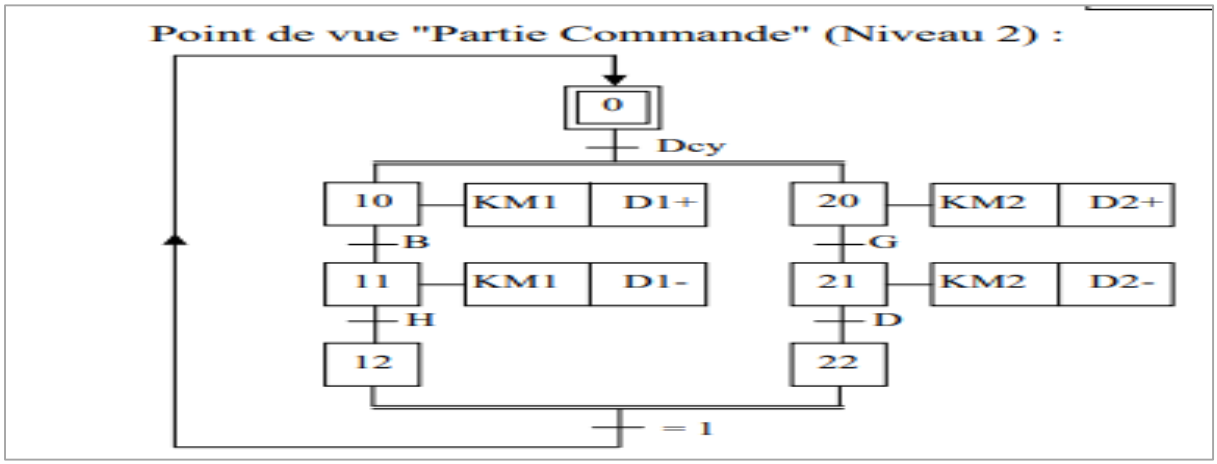
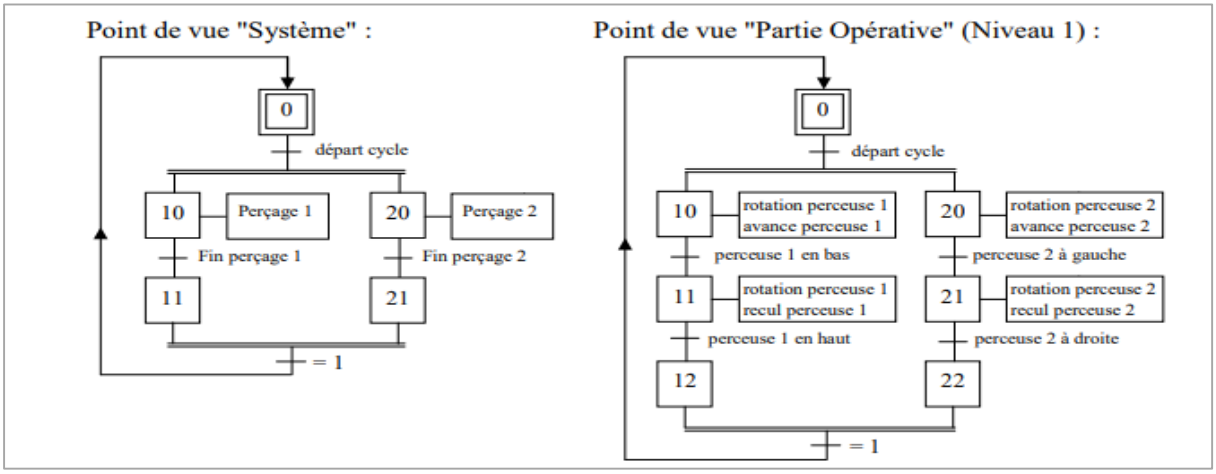


Figure 1.17 : Unité de perçage.





I.6.2.2. Les Inconvénients de GRFCET

- Difficulté de gérer des événements simultanés : Le Grafcet est conçu pour gérer des événements discrets et séquentiels, mais il peut être difficile de gérer des événements simultanés ou parallèles. Cela peut entraîner des conflits ou des problèmes de priorité dans le système. Des techniques telles que la modélisation par réseau de Petri peuvent être utilisées pour modéliser des événements simultanés.

- Difficulté à gérer des systèmes complexes : Le Grafcet peut être limité dans sa capacité à modéliser des systèmes complexes qui comportent de nombreuses interactions et dépendances entre les différents composants. Cela peut rendre la modélisation de systèmes complexes difficile ou impossible à l'aide du Grafcet seul.

- Limitation de la lisibilité du code : Le Grafcet peut devenir difficile à lire et à comprendre si le système modélisé est complexe ou si le code devient trop long et détaillé. Des techniques telles que la Modélisation ou l'utilisation de diagrammes plus grands peuvent aider à améliorer la lisibilité.

- Dépendance à la compréhension de l'utilisateur : Le Grafcet dépend de la compréhension de l'utilisateur pour interpréter correctement le comportement du système. Si l'utilisateur n'a pas une Compréhension précise du système modélisé, cela peut conduire à des erreurs ou à des interprétations incorrectes du comportement attendu du système.**[09]**

I.6.3. Modélisation par des files d'attente

a/ Notations et définitions générales

Généralement une file d'attente est la donnée d'une (ou plusieurs) unités de services où arrivent des clients qui demandent une certaine durée d'utilisation de cette unité (le service demandé par les clients) .Quand les clients peuvent accéder à cette unité de service, ils patientent dans une file d'attente en attendant d'être servis .La file d'attente peut éventuellement n'accepter qu'un nombre fini des clients, dans ce cas les clients trouvant la file pleine à leur arrivée sont rejetés par le système. Un client peut être servi pendant une certaine période puis abandonné par le serveur. Le service résiduel d'un client est la durée du service qui reste à effectuer, quand celui-ci est nul, le client quitte la file d'attente. La charge de la file d'attente est la somme de tous les services résiduels de tous les clients présent **[10]**

I.6.3.1. La Théorie des files d'attente

La Théorie des files d'attente est une technique de la Recherche opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions.

Les files d'attente peuvent être considérées comme un phénomène caractéristique de la vie contemporaine. On les rencontre dans les domaines d'activité les plus divers spécialement dans le trafic routier (carrefour à feux). L'étude mathématique des phénomènes d'attente constitue un champ d'application important des processus stochastiques. On parle de phénomène d'attente chaque fois que certaines unités appelées "clients" se présentent d'une manière aléatoire à des "stations" afin de recevoir un service dont la durée est généralement aléatoire**. [10]**

I.6.3.2. Modèle de Files d'attente:

Tous les exemples de phénomènes d'attente ont des caractéristiques communes que l'on peut résumer ainsi des entités circulent dans un système et utilisent des ressources communes. Le système, les entités ou les ressources peuvent avoir un comportement imprévisible, c'est-à-dire dans le contexte d'une modélisation mathématique, aléatoire. Une file d'attente se compose des éléments suivants **:**

1. La population.
2. Le number de servers.
3. Les tendances quant à l'arrivée et au service.
4. . L'ordre de traitement des clients.
5. Une salle d'attente, c'est-à-dire un lieu ou les clients attendent quand aucun des serveurs n'est disponible pour les servir. Tout système de file d'attente peut être représenté par le schéma suivant :

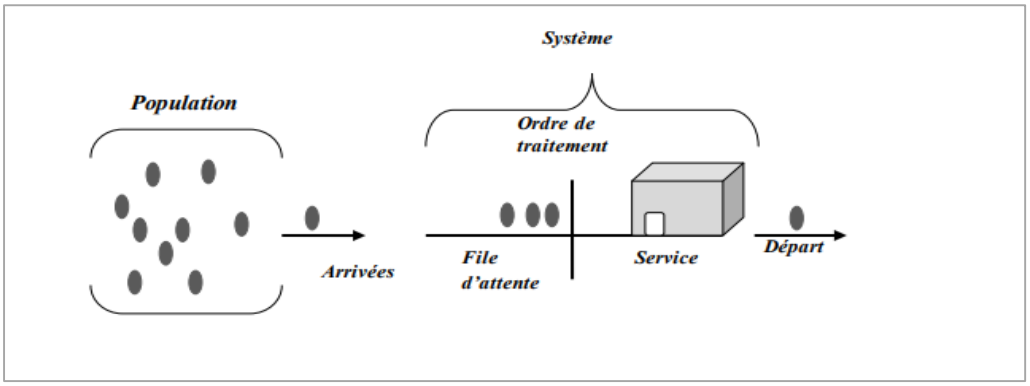


Figure 1.18: Structure générale d'un système de file d'attente.

b/ La population:

Dans la théorie des files d'attente, la population est la source de clients potentiels. Il y a deux situations possibles. Dans le premier cas, la population est infinie, c'est-à-dire que le nombre potentiel de clients est infiniment grand en tout temps. C'est le cas des clients des supermarchés, des banques, des restaurants, des cinémas, des centres d'appels, trafic urbain, etc. De plus, les clients proviennent de toutes les régions possibles. Dans la deuxième situation, la population est finie, ce qui signifie que le nombre de clients potentiels est limité. Un bon exemple est le nombre de machines, d'avions, etc., en réparation dans le centre de maintenance d'une entreprise.

c/ Le nombre de serveurs:

La capacité de service dépend de la capacité de chaque serveur et du nombre de serveurs disponibles. Le terme « serveur » représente ici la ressource et, en général, on suppose qu'un serveur ne traite qu'un client à la fois. Les systèmes de files d'attente fonctionnent avec serveur unique ou serveurs multiples.

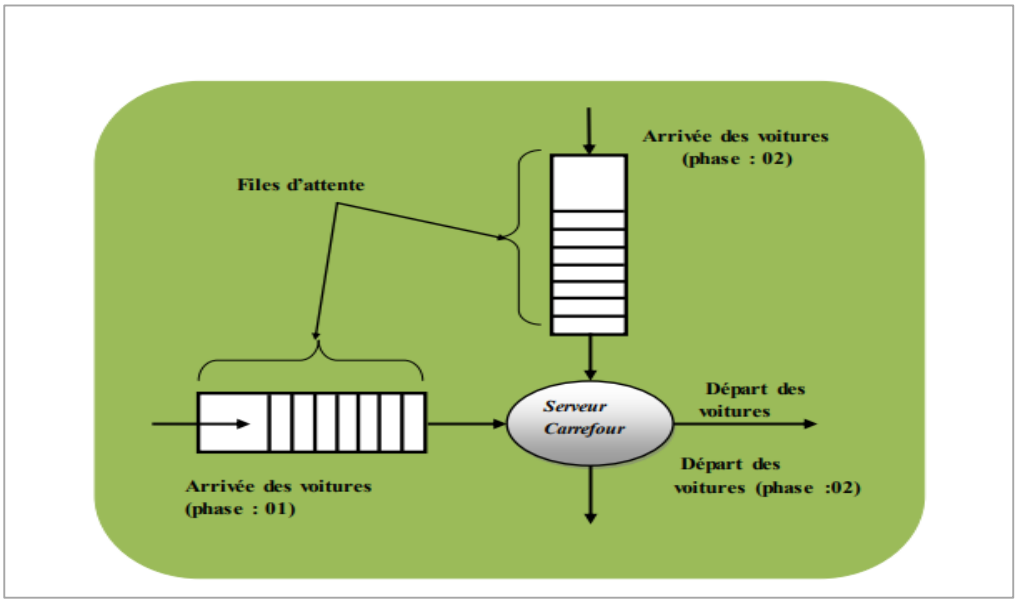


Figure 1.19: modélisation du système par un réseau de files d'attente

I.6.4. Système multi-agents

a/ Définition

Un système multi-agents est essentiellement un système composé d'entités autonomes(appelées agents) qui peuvent interagir les unes avec les autres au sein d'un environnement commun. Cet environnement est un espace ou plus souvent une portion d'espace, continu ou discret, de topologie quelconque : il peut s'agir aussi bien d'un espace euclidien que d'une « grille », d'un graphe (par exemple pour décrire des relations d'accointance) ,il peut éventuellement se réduire à un point si l'on considère que le système n'est pas « spatialisé ». Mais son rôle premier est de servir de support à l'activité ou à la communication des agents. Quant à l'agent, la manière la plus simple de le définir est celle de " **Russell et Norvig** ".**[11].**

I.6.4.1. Le principe de la simulation multi-agent

La simulation multi-agents (MAS) est un type de simulation informatique où plusieurs agents autonomes interagissent entre eux et avec leur environnement. Chaque agent a son propre ensemble de règles, de comportements et de processus de prise de décision, et ils peuvent communiquer et collaborer avec d'autres agents pour atteindre leurs objectifs.

- Les agents: Les agents individuels sont programmés avec des règles de comportement spécifiques et sont capables de percevoir leur environnement et d'interagir avec les autres agents. Les agents peuvent être des personnes, des animaux, des organisations, des machines ou des éléments abstraits.

- L'environnement: L'environnement dans lequel les agents évoluent est modélisé en utilisant des variables et des paramètres qui décrivent les caractéristiques de l'environnement. L'environnement peut être physique, social, économique ou écologique.

- Les interactions: Les agents interagissent les uns avec les autres et avec leur environnement en utilisant des mécanismes de communication, de coopération, de compétition ou de conflit.
Les interactions peuvent être directes ou indirectes, et peuvent impliquer des relations de dépendance ou d'indépendance.

- Les règles de comportement **:** Les agents sont programmés avec des règles de comportement spécifiques qui influencent leur prise de décision et leur comportement dans des situations spécifiques. Les règles de comportement peuvent être simples ou complexes, et peuvent être basées sur des modèles mathématiques, des algorithmes ou des heuristiques.

- Les scénarios: La simulation multi-agent permet de créer différents scénarios en modifiant les paramètres de l'environnement, les règles de comportement des agents ou les interactions entre les agents. Les scénarios peuvent être utilisés pour tester des hypothèses ou pour prédire les résultats de différentes situations.

- **L'évaluation:** La simulation multi-agent permet d'évaluer les résultats de différents scénarios en utilisant des indicateurs de performance spécifiques. Les indicateurs peuvent être quantitatifs ou qualitatifs, et peuvent être utilisés pour mesurer l'efficacité, l'équité, la durabilité ou d'autres aspects du système modélisé **[09]** , **[12].**

I.7.le système hybride

a/ Définition:

Systèmes hybrides: un système hybride combine des éléments de systèmes continus et de systèmes à événements discrets. Les variables d'état d'un système hybride peuvent évoluer de manière continue dans le temps et changer de manière discrète en réponse à des événements. Les systèmes hybrides sont souvent utilisés pour modéliser des situations où des événements discrets interagissent avec des processus continus.**[02]**

Systèmes dynamiques hybrides: Un système dynamique hybride est un type particulier de système hybride. Il s'agit d'un système hybride dont les évolutions continues et discrètes sont régies par des équations différentielles et des règles de transition bien définies. Les systèmes dynamiques hybrides sont souvent utilisés pour modéliser et analyser des systèmes complexes où les comportements continus et discrets sont étroitement liés, tels que les systèmes de contrôle, les systèmes robotiques et les réseaux de capteurs.**[12]**

Formellement, un système dynamique hybride (**SDH**) est composé d'un 5- tuple :

SDH = (T, (x, q),(x_0, q_0), $U_c \cup U_d$, Φ):

- T∈ R est l'intervalle de temps
- (x, q) ∈ X × Q représente l'état complet du système hybride.
- (x_0, q_0) est l'ensemble des états initiaux.
- $U_c \cup U_d$représente l'ensemble des commandes continues etdiscrètes.

I.7.1. Structure Générique D'un Système Dynamique Hybride

D'une façon générale, un système dynamique hybride SDH est composé d'un système dynamique continu, d'un système à événement discrets et d'une interface qui gère les interactions entre les deux évolutions (continue et discrète).

a/ La Partie Discrète :

La partie discrète est associée à un système à événement discret SED dont l'évolution est représentée par un ensemble fini d'états. La transition d'un état discret à un autre état discret successeur est réalisée grâce à l'occurrence d'événements. Ces événements sont de deux types : événements contrôlés et événements autonomes.

b/ La Partie Continue:

L'évolution de la partie continue est caractérisée sur un espace de temps continus et peut être représentée de différentes manières (équations différentielles ordinaires, algèbre-différentielles, les fonctions de transfert, les bonds graphs, etc.). Le modèle de comportement continus est exprimé sous la forme d'état

$$\begin{cases} \dot{x}(t) = f\left(x(t), z(t)\right) \\ \mathrm{h(t)} = \left(\mathrm{x(t)}, \mathrm{z(t)}\right) \end{cases} \qquad (\mathbf{I.11})$$

Où

x(t) est le vecteur d'état.
y(t) est le vecteur des sorties.
z(t) est un vecteur regroupant les entrées connues ou inconnues, les perturbations, etc. f et h sont des fonctions vectorielles linéaires ou non linéaires.**[10]**

c/ L'interface:

L'interface traduit l'interaction entre la partie continue et la partie discrète du SDH. La représentation formelle de l'interface est plus complexe et dépend des approches de modélisation considérées.

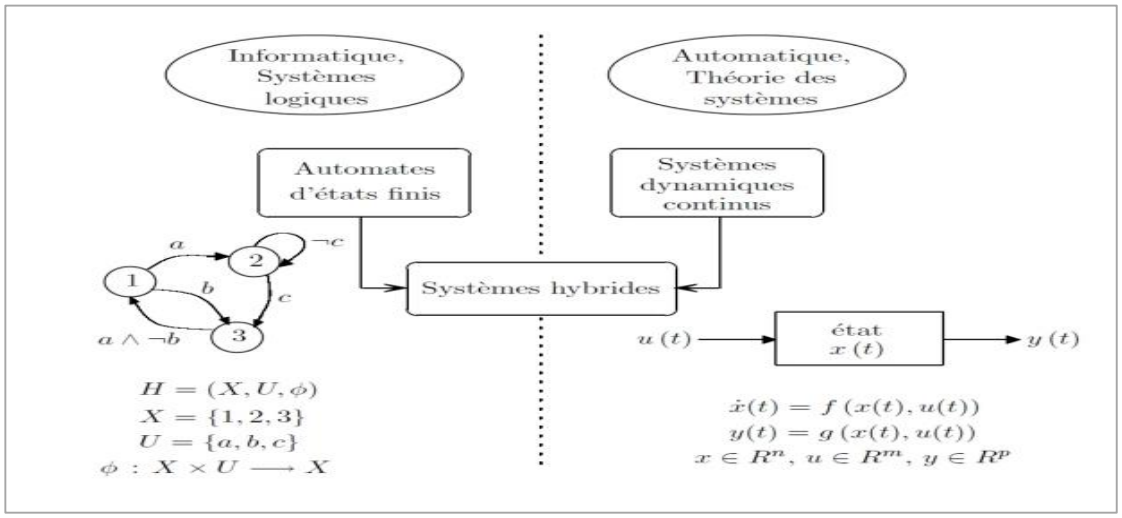


Figure 1.20: Structure du système dynamique hybride.

I.7.2. Exemples de systèmes dynamiques hybrides

a/ Circuit électrique:

Le système illustré par la **Figure 1.21** est composé d'une source de courant, d'une diode D, d'une résistance R_1, d'un interrupteur S_W, et d'une self L. Quand l'interrupteur est fermé, le courant $I_L$traversant la self génère un flux magnétique ϕ_0. Lors de l'ouverture de l'interrupteur le flux se décharge très rapidement**.** Ce changement d'état discret de l'interrupteur crée une discontinuité du flux qui passe de ϕ_0à0. Cette variation du flux entraîne l'apparition d'une force électromotrice (f.e.m) selon la loi : f.e.m = dϕ/dt donc d'un courant induit I_iLe rapport dϕ/dt est important, la différence de potentiel V_L (qui est la f.e.m) au borne de la self est donc importante.La diode est alors passante car le courant $I_L$généré par la self est supérieur au courant de seuil Is de la diode . Quand le courant $I_L$généré par l'inductance devientinférieur à I_s,la diode passe alertât "Bloqué".L'évolution du système met en évidence une dynamique continue entrecoupée par des commutations dues à l'état discret de l'interrupteur qui peut être "Fermé" Ou "Ouvert" et à l'état discret de la diode qui peut être "Passant" ou "Bloqué". Le changement d'état discret de l'interrupteur crée une discontinuité du flux dans l'inductance qui passe instantanément de ϕ_0à 0.**[13]**

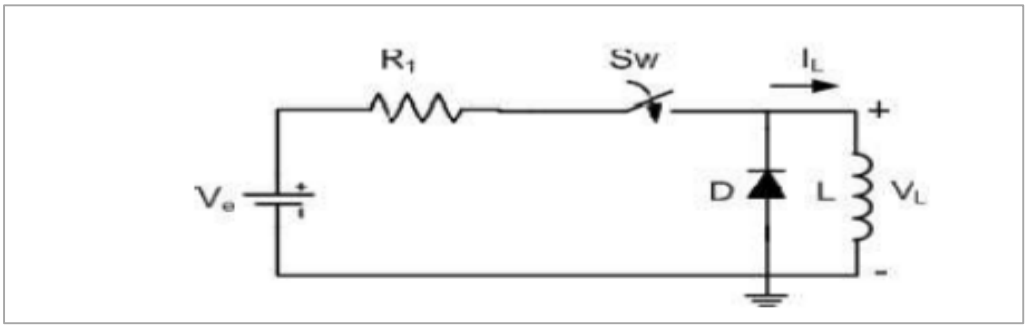


Figure 1.21: Circuit électrique intrinsèquement hybride.

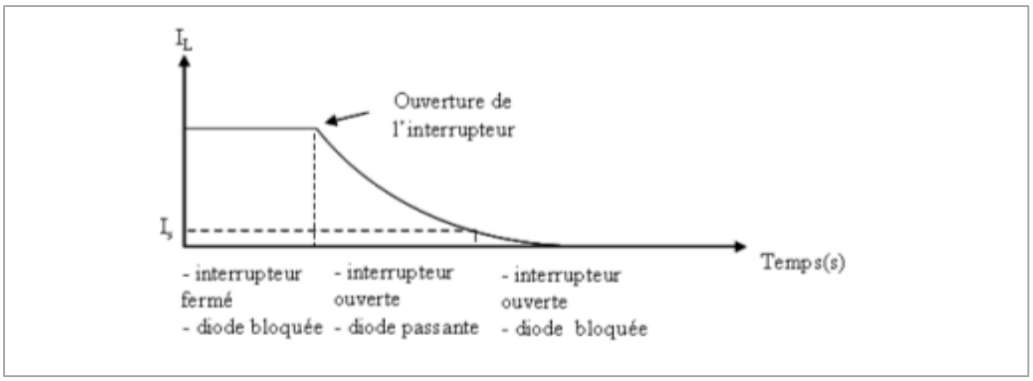
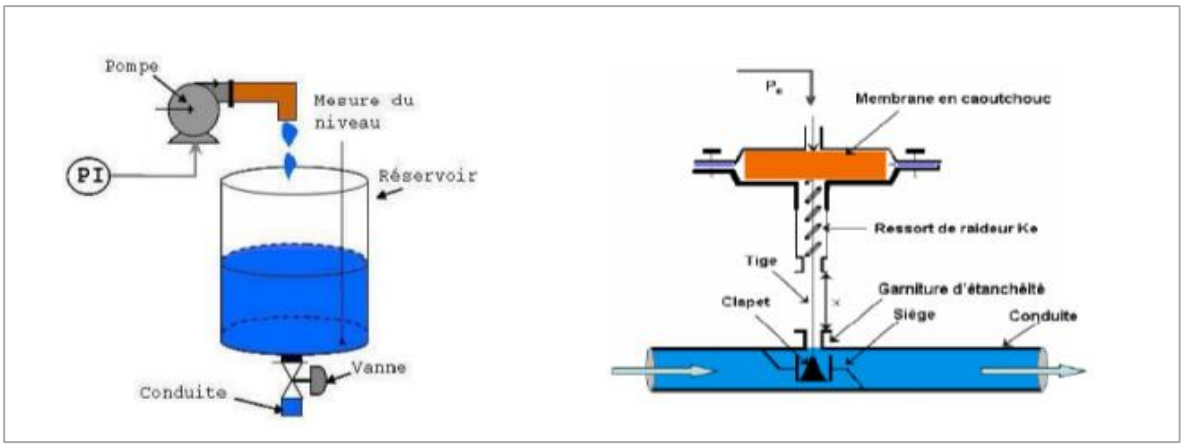


Figure 1.22: Figure :Evolution du courant dans la self.

b/ Système hydraulique:

Le système est constitué d'un réservoir de section S muni d'une conduite C. Une vanne pneumatique V située sur C et commandée en Tout ou Rien (**TOR**),permet de prélever du liquide pour utilisation. Une pompe P permet d'alimenter le réservoir et une commande automatique (**régulateur PI**) permet de maintenir un niveau de liquide constant dans le réservoir Le niveau de liquide h(t) est une variable dont l'évolution est continue. Sa valeur dépend des débits entrants (sortie de la pompe) et sortants (évacuation au travers de la vanne V). Ces deux débits ont eux aussi des évolutions continues : le débit d'entrée est régulé (régulateur PI) et la valeur du débit de sortie est fonction de l'ouverture (restriction) de la vanne pneumatique. L'ouverture de la vanne est en toute rigueur une fonction continue dans le temps. Son évolution est donnée par la fonction ϕe, ϕe (x), où x est la position de la tige. Lors d'une commande d'ouverture (passage de x = 0% à x = 100%), l'évolution continue dex, donc ϕe, est très rapide et la dynamique peut donc être négligée. Ainsi un état discret associé à la vanne peut être considéré. Cet état prend deux valeurs ou modalités correspondant respectivement à x = 0% (vanne fermée) et x = 100% (vanne ouverte). Ceci suffit en première approximation à caractériser le comportement de la vanne dans le système global. L'introduction de cet état discret met en évidence deux modes de fonctionnement du système et permet de simplifier la modélisation complète du système.**[09]**



Système hydraulique. **Exemple de vanne pneumatique.**

c/ Système de production

Le système illustré par la **Figure 1.23** est composé d'un ensemble de ressources ou stocks (S_1, S_2 ,$S_{2'}$,...) et d'un groupe de machines (M_1 , M_2 , M_3 ...). Chaque groupe est composé d'un nombrede machines identiques. Les stocks (S_1, S_2, $S_{2'}$,...) sont utilisés pour emmagasiner les pièces(matière première ou pièces en cours de traitement) jusqu'au moment où une machine en avales disponible pour commencer un nouveau traitement. Ce système de production peut être vu comme étant un système hybride ayant une évolution continue, représentant les flux de pièces dans le système et une évolution discrète liée à l'état des ressources. Le niveau des pièces dans les stocks peut être modélisé par une équation différentielle linéaire $\dot{x}(t) = A.x(t) + B.u(t)$ **(I.12)**
Où x=[x_1 ...; x_i ...; x_n], t représente le niveau de pièces dans les stocks (la composante xireprésente le niveau de pièces dans le stock Si) et u étant le débit des pièces en entrée.L'évolution discrète du système de production est décrite par l'occurrence des événements associés à l'intervention de l'opérateur pour démarrer ou arrêter le flux d'entrée et aux états des ressources. L'occurrence de l'un de ces événements entraîne le changement de l'état discret: flux d'entrée démarré, flux d'entrée arrêté, machine1 en marche ou en arrêt, etc.**[10]**

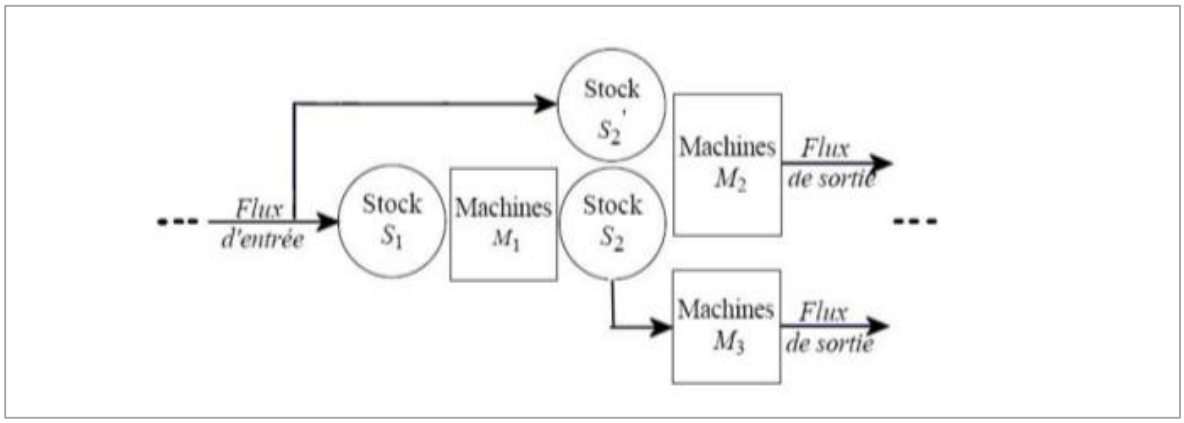


Figure 1.25: Architecture générique d'un système de production.

I.8.Les outils de modélisation

1.8.1. Réseau de Pétri hybride

Définition :

Un Rdp hybride (Hybride Pétri Nets (PNH)) est une structure PNH = (P, T, b, E, Pré, Post, Σ, Tempo, V, M_0) tel que :

* P = {P1, P2, ... ,P_n } est un ensemble de n places, P = $P^C \cup P^D$ avec.

P^C= {P1, P2, ... ,P_{nC}} est l'ensemble fini de places continues (ou C-places) .

P^D= {P_{nC+1}, ... ,P_n} est l'ensemble fini de places discrètes (ou D-places) .

* T = {T1, T2, ... ,T_m} est un ensemble de m transitions, T = $T^C \cup T^D$avec .

- T^C= {T1, T2, ... ,T_{mC}' } est l'ensemble fini de transitions continues (ou C transitions).

- T^D= {T_{mC+1}, ... ,T_m} est l'ensemble fini de transitions continues (ou Ctransitions).

* b:P∪T→ {D,C} est une application qui désigne les nœuds discrets, h(x) = D, et les nœuds continus, h(x) = C .

* E est un ensemble fini d'événements .

* Σ : T^D→ E est une fonction qui associe à chaque transition discrète un événement de E .

* Pré et Post désignent respectivement les applications d'incidence avant et arrière, ces applications doivent satisfaire la condition suivante :

∀(Pi, Tj) ∈P^C× P^D : Pré(Pi, Tj) = Post(P_i, T_j).

- Tempo : → Q^+ est une application qui associe à chaque D-transition la durée de sa temporisation.
- V: T^C→ R^+est une application qui associe à chaque C-transition sa vitesse maximale de franchissement.
- M_0 est le marquage initial, des D- places contiennent un marquage entier positif et les C-place .

La condition sur les applications d'incidence avant et arrière est repérée sur le RdP par des boucles reliant les D- places aux C-transitions, elle signifie qu'une marque discrète ne peut pas être fluidifiée par une transition continue. Le modèle RdP hybride ainsi défini permet donc la modélisation des conditions logiques influant sur le comportement du système, mais il permet aussi la modélisation de transformation de marques continues en marques discrètes et vice-versa (formation et éclatement de lots).**[13]**

Nous numérotons les nœuds du RdP de telle sorte que les nœuds continus aient les indices les plus petits ,cela fait que la matrice d'incidence a la forme suivant:

$$w = \begin{bmatrix} w_C & w_{cD} \\ 0 & w_D \end{bmatrix}$$

Les RdP élémentaires constituent une classe particulière de RdP hybrides ou il n'ya pas de transformation de marquage, du discret vers le continu ou du continu vers le discret. Dans ce modèle le RdP T-temporisé contrôle le comportement du RdP continu C via des boucles connectant certaines D-places à certaines C-transitions, ce qui signifie que ces dernières ne sont validées et par conséquent ne peuvent être franchies que si les D-places sont marquées. Le RdP continu C à son tour peut influencer le comportement du RdP T-temporisé une Transition T_j peut avoir comme condition de franchissement le marquage d'une C-place $P_i$qui atteint un seuil S.

Graphiquement ceci est représenté de deux manières soit par une boucle(un arc de P_i vers Tj et un arc de $T_j$vers P_i) dont le poids est S si ce seuil est un seuilsupérieur c'est-à-dire si le marquage de $P_i$ne peut être supérieur à S.Dans le cas contraire si le marquage de $P_i$ne doit pas être inférieur à S, un arc inhibiteur est utilisé pour relier T_jà P_i. Et dans les deux cas le franchissement de $T_j$ne modifie pas le marquage de P_i**.[11] ,[12] , [14]**

Exemple :

En exemple, on modélise un système de fabrication produisant des pièces par lots de cinq par un RdP hybride. A la fin de la production de 2 lots de 5 pièces, on entame un nouveau cycle de production. Ce système est illustré par la **Figure 1.25.**

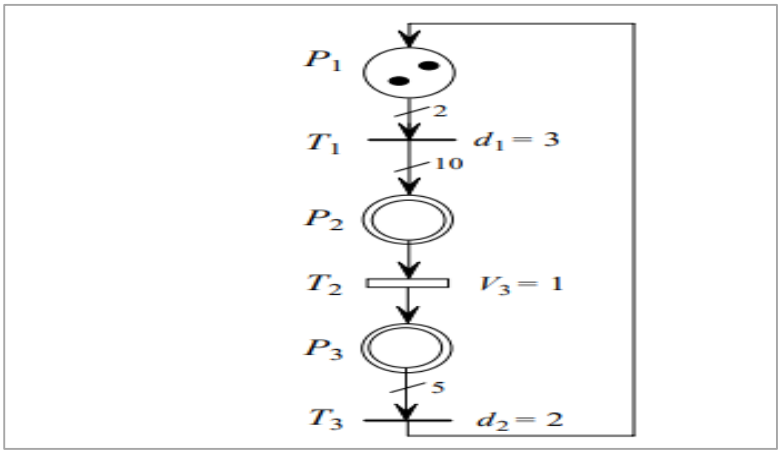


Figure 1.23:Modèle de RdP hybride d'un système de fabrication par lots

Le marquage de la place P1 (D-place) est associé au nombre de lots à l'entrée du système de fabrication. Les valeurs d1 et d2 représentent les temps de chargement et de déchargement des pièces (D-transitions). La transition T3 modélise une machine dont la vitesse de production est V3 (C-transition). Les stocks d'entrée et de sortie de la machine sont respectivement définis par les places P2 et P3 (C-places). Le franchissement continu de la transition T3 correspond à une production continue à la vitesse V3 quand la place P2 n'est pas vide. Lorsque P2 est marqué. le franchissement d'une quantité V3dt de T3 correspond au retrait de V3dt marques à P2 et à l'ajout de la même quantité à P3.

I.8.2. Automates Hybrides

Un automate hybride est un outil de représentation qui permet de tenir compte explicitement des deux évolutions continue et discrète du SDH. Il apparait comme l'association d'un automate à états finis pilotant un ensemble d'équations dynamiques continues. Les équations modélisant le comportement continu à un instant donné dépendent de l'état de l'automate mais ce dernier peut évoluer en fonction de la valeur des grandeurs Continues **.[13]**

Un automate hybride est un graphe composé de sommets (ou places) et d'arcs orientés modélisant les transitions discrètes qui relient les sommets. Tout arc orienté doit avoir un sommet destination.

Exemple :

$$\boldsymbol{Mode\ 1 :\ x'(t) = a_1 x(t) + b_1} \qquad (\mathbf{I.13})$$

$$\boldsymbol{Mode\ 2 :\ x'(t) = a_2 x(t) + b_2} \qquad (\mathbf{I.14})$$

$$\boldsymbol{a_1, a_2, b_1, b_2 \in \mathcal{R}}$$

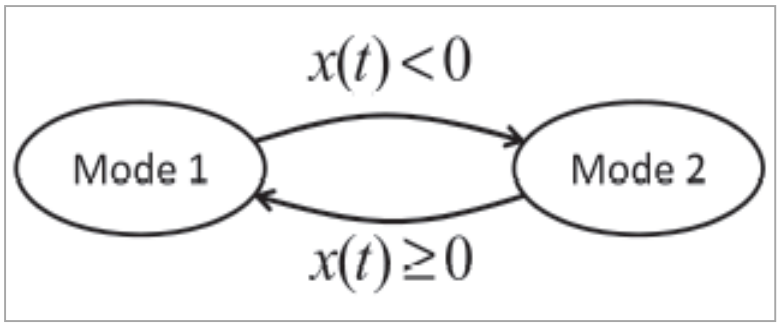


Figure 1.24: Automate hybride.

En effet, cette modélisation permet de représenter à la fois l'évolution de l'état discret (entre les modes 1 et 2) et la dynamique de l'état continu x(t) associé à chaque mode.

I.9.Conclusion

Le chapitre I présente une étude générale sur les déférents types de systèmes industriels. Il explique que les systèmes peuvent être classés en plusieurs catégories, notamment les systèmes continus et à évènement discret. Chacun de ces systèmes a ses propres avantages et inconvénients, et le choix du type de système dépendra de la nature du problème à résoudre.

Les théories des systèmes continus sont utiles pour modéliser des phénomènes physiques tels que les mouvements de véhicules ou les circuits électriques. Les théories des systèmes à événements discrets sont utiles pour modéliser des systèmes de traitement de données tels que les réseaux informatiques et les systèmes de production. Les systèmes hybrides et leur théorie sont utiles pour modéliser des systèmes complexes tels que les systèmes de contrôle de la circulation aérienne ou les systèmes de production qui impliquent à la fois des processus continus et des événements discrets.

Chapitre II

Généralités sue les automates programmable industriel

II.1. Introduction

Dans le domaine de l'industrie, l'automatisme est utilisé pour piloter les moyens de production. L'objectif des équipements d'automatisme est de produire tout en assurant l'intégrité de la chaine de production et la sécurité des personnes. Les plateformes d'implémentation sont souvent composées d'Automates Programmables Industriels (API) notamment pour leur facilité d'intégration et pour leur robustesse de fonctionnement. L'utilisation de ces API nécessite des méthodes de programmation basées sur la standardisation des langages de programmation.**[15]**

Ce chapitre consiste à décrire d'une manière globale l'API, son rôle et son principe de fonctionnement.

II.1.2. Système automatisé

Définition

Un système automatisé ou automatique est un système réalisant des opérations et pour lequel L'homme n'intervient que dans la programmation du système et dans son réglage. Il s'appelle aussi un système technique command able. On dit qu'un système est command able si en faisant varier uniquement les entrées, on peut faire subir des modifications au système, afin qu'il atteigne un objectif fixé en un temps fini. **[15]**

II.1.2.3. Les objectifs D'un système automatisé

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée. Ces éléments sont exprimables en termes d'objectifs :

- La recherche des coûts plus bas pour le produit par la réduction des frais de main d'œuvre, d'économie d'énergie, d'économie de la matière, etc.
- La recherche d'une meilleure qualité du produit en limitant le facteur humain et multipliant les contrôles automatisés.
- L'amélioration de la flexibilité de la production.
- La suppression des travaux dangereux ou pénibles et l'amélioration des conditions de travail.
- La réalisation d'opérations impossibles à contrôler manuellement, par exemple des assemblages miniatures, des opérations très rapides, des coordinations complexes.**[16]**

II.1.2.4. Structure d'un système automatisé

Un Système Automatisé est composé d'une **Partie Commande** et d'une **Partie Opérative** pour faire fonctionner ce système, l'Opérateur (personne qui va faire fonctionner le système) va donner des consignes à la Partie Commande. Celle-ci va traduire ces consignes en ordres qui vont être exécutés par la Partie Opérative. Une fois les ordres accomplis, la Partie Opérative va le signaler à la Partie Commande (elle fait un compte-rendu) qui va à son tour le signaler à l'Opérateur. Ce dernier pourra donc dire que le travail a bien été réalisé.**[15]**

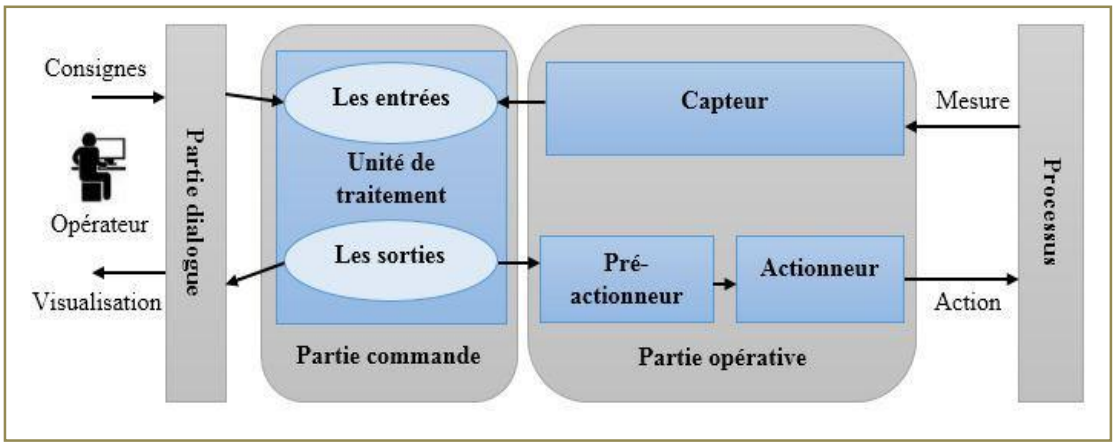


Figure II.1 :différents modules de la machine.

II 1.2.5. Partie opérative

Que l'on appelle également partie puissance, c'est la partie visible du système (corps) qui permet de transformer la matière d'œuvre entrante. Elle est composée d'éléments mécaniques, d'actionneurs (vérins, moteurs), de pré-actionneurs (distributeurs et contacteurs) et des éléments de détection (capteurs, détecteurs).

- **Les actionneurs**

Est un élément de la Partie Opérative qui reçoit une énergie « transportable » pour la transformer en énergie « utilisable » par le système. Ils exécutent les ordres reçus en agissent sur le système ou son environnement. Un actionneur est un système dont la matière d'œuvre est l'énergie et dont la fonction est de transformer l'énergie.
Ces actionneurs appartiennent à trois technologies **:**
a) Actionneurs pneumatiques (vérins, moteurs).
b) Actionneur hydraulique (vérins).
c) Actionneurs électriques (moteurs électriques).

- **Pré-actionneur**

Le Pré-actionneur est le constituant qui autorise le passage de l'énergie du milieu extérieur vers l'actionneur. Le Pré-actionneur distribue l'énergie nécessaire à l'actionneur en fonction des ordres reçus.
Le pré-actionneur peut être :
a) Contacteurs pour moteurs électriques
b) Variateurs de vitesse pour moteurs électriques.
c) Distributeurs pour vérins pneumatiques ou hydrauliques.

- **Les capteurs**

Les Capteurs permettent de prélever sur la partie opérative, l'état de la matière d'œuvre et son évolution, il est capable de détecter un phénomène physique dans son environnement (déplacement, présence, chaleur, lumière, pression...) puis transforme l'‘information physique en une information codée compréhensible par la partie commande.

Ce qui mène à que les capteurs transforment la variation des grandeurs physiques liées au fonctionnement de l'automatisme en signaux électriques. **[16]**

II.1.2.6. Partie commande

Elle est considérée comme le « cerveau » du système. La partie commande remplace l'opérateur, le savoir faire de l'opérateur est traduit sous la forme d'un programme. Elle élabore des ordres destinés à la partie opérative en fonction de :

a) Programme hébergé.
b) Informations acquises par les capteurs.
c) Consignes données par l'utilisateur.

II .1.2.7. partie interface

Composé des pupitres de commande et de signalisation, il permet à l'opérateur de communiquer avec système et donner des consignes (marche, arrêt, départ cycle …).

II.1.3. Les avantages et les inconvénients de l'automatisation des systèmes

II.1.3.1. Les avantages

L'automatisation des systèmes a pour avantages :

- améliorer les conditions de travail (effectuer des tâches pénibles, dangereuses et répétitives).
- sécurité.
- précision.
- réduire les coûts de fabrications (produits plus compétitifs).
- augmenter la productivité (réduire le temps de travail nécessaire à la production, donc augmenter les cadences de travail).
- flexibilité (une machine peut s'adapter à plusieurs productions).**[17]**

II.1.3.2. Les inconvénients

- investissement pour l'achat de machines (le coût élevé du matériel).
- coût de maintenance.
- consommation d'énergie.
- formation d'un personnel plus qualifié (technicien de maintenance, de contrôle...).**[17]**

II.2. Automate programmable industriel (API)

II.2.2. Définition d'un API

L'automate programmable industriel (API), ou en anglais 'Programmable Logique Controller' (PLC), est une machine électronique programmable destinée à piloter dans une ambiance industrielle et en temps réel des procédés logiques séquentiels. Autrement dit, un Utilisateur (censé être un automaticien) l'utilise pour le contrôle et essentiellement la commande d'un procédé industriel en assurant l'adaptation nécessaire entre tout ce qui est de grande puissance par rapport à ce qui est de faible puissance côté commande. Son objectif principal est de rendre tout le mécanisme de type "laisser-faire-seul" : le système contrôle ses sorties, décide et agit sur

ses entrées afin de maintenir le fonctionnement comme prévu par l'utilisateur. **[18]**

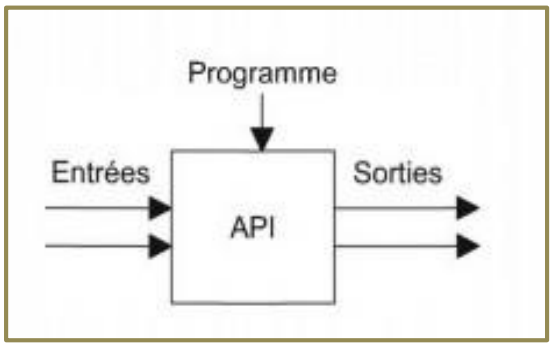


Figure II.2 : Un automate programmable industriel (API).

II.2.3. Domaines d'emploi des automates

Les API s'adressent à des applications que l'on trouve dans la plupart des secteurs Industriels. Ces machines fonctionnent dans les principaux secteurs suivants **[18] :**

- Métallurgie et sidérurgie.
- Mécanique et automobile.
- Industries chimiques.
- Industries pétrolières.
- Industries agricoles et alimentaires

II.2.4. Communication avec un API

L'automate doit pouvoir se connecter et dialoguer avec d'autres matériels et les agents d'exploitation .L'API ne se limite pas à communiquer avec le processus qu'il pilote via ses modules d'E/S. Parmi les autres types de relations susceptibles d'être assurées, on cite :

- La communication avec un opérateur par un pupitre ou un terminal industriel.
- L'affichage local de valeurs numériques ou de message.
- Les échanges d'informations avec d'autre API ou système de commande.
- Les échanges d'informations avec des capteurs et actionneur intelligents.
- Les échanges d'informations avec un processeur maître, ou avec des esclaves, dansle carde d'un réseau **[17].**

II.2.5.Le choix d'un automate

Le choix d'un API est en fonction de la partie commande à programmer. Il est imperatif de tenir compte de plusieurs critères :

- Le type des entrées/sorties nécessaire.
- Le nombre d'entrées/sorties nécessaire.
- La qualité du service après-vente.
- Les capacités de traitement du processeur (vitesse, données, opérations, temps)

- Les compétences/expériences de l'équipe d'automaticiens en mise en œuvre et en programmation de la gamme d'automate**. [19]**

II.3.Architecture des automates programmables industriels

II.3.1. Fonctionnement de l'API :

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

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

Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique). On appelle scrutation, l'ensemble des quatre opérations réalisées par l'automate et le temps de scrutation est le temps mis par l'automate pour traiter la même partie de programme. Ce temps est de l'ordre de la dizaine de millisecondes pour les applications standards.**[20]**

II.3.2.Architecture interne d'un API

En général, un API est constituée de composants fonctionnels de base suivante :

- Une unité de traitement,
- La mémoire,
- Une unité d'alimentation,
- Des interfaces d'entrées-sorties,
- Une interface de communication

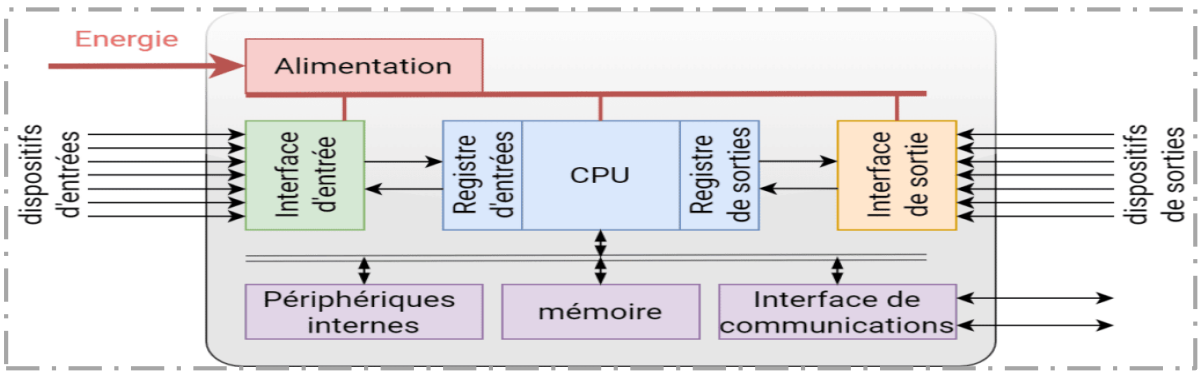


Figure II.3: Structure interne d'un Automate Programmable Industriel.

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple, l'unité centrale est le regroupement du processeur et de lamémoirecentrale.Ellecommandel'interprétationetl'exécutiondesinstructionsprogramme

II.3.3.Description des éléments d'un API

✓ La mémoire

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système, qui sont le terminal de programmation (PC ou console) et le processeur. Elle reçoit également des informations en provenance des capteurs (Figure **II.4)[21].**

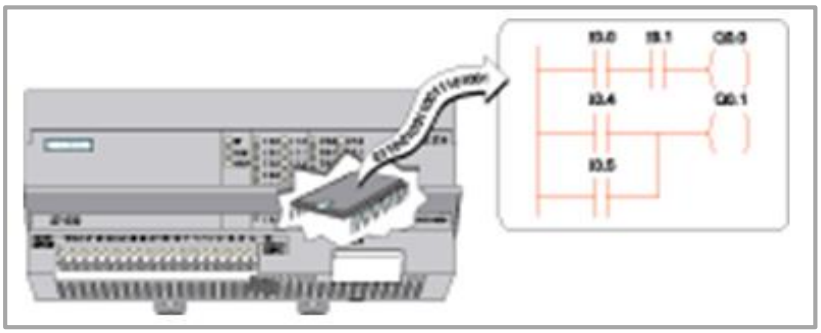


Figure II.4 : La mémoire d'un API

Il existe dans les automates deux types de mémoires qui remplissent des fonctions différentes :

- La mémoire Langage où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte)
- La mémoire Travail utilisable en lecture-écriture pendant le fonctionnement c'est la RAM(mémoire vive).Elle s'efface automatiquement à l'arrêt de l'automate (nécessite une batterie de sauvegarde) , Réparation des zones mémoires:

  - Table image des entrées.
  - Table image des sorties.
  - Mémoire des bits internes.
  - Mémoire programme d'application.

✓ Le processeur

Le processeur, appelé aussi unité de traitement ou unité arithmétique et logique, a double vocation d'assurer le contrôle de l'ensemble de la machine et effectuer les traitements demander par l'instruction des programme. Il lit permanence et à grande vitesse les états.

Logiques des signaux en provenance des capteurs périphériques en fonction du programme stocké dans la mémoire, et il transmet des ordres de sortie vers les actionneurs**.**

✓ Les interfaces et les cartes d'Entrées/Sorties

L'interface d'entrée comporte des adresses d'entrée. L'interface de sortie comporte de la même façon des adresses de sortie. Le nombre de ces entrées et sorties varie suivant le type d'automate. Les cartes d'E/S ont une modularité de 8, 16 ou 32 voies. Les tensions disponibles sont normalisées (24, 48, 110 ou 230V continu ou alternatif ...) (**Figure II.5**).**[21].**

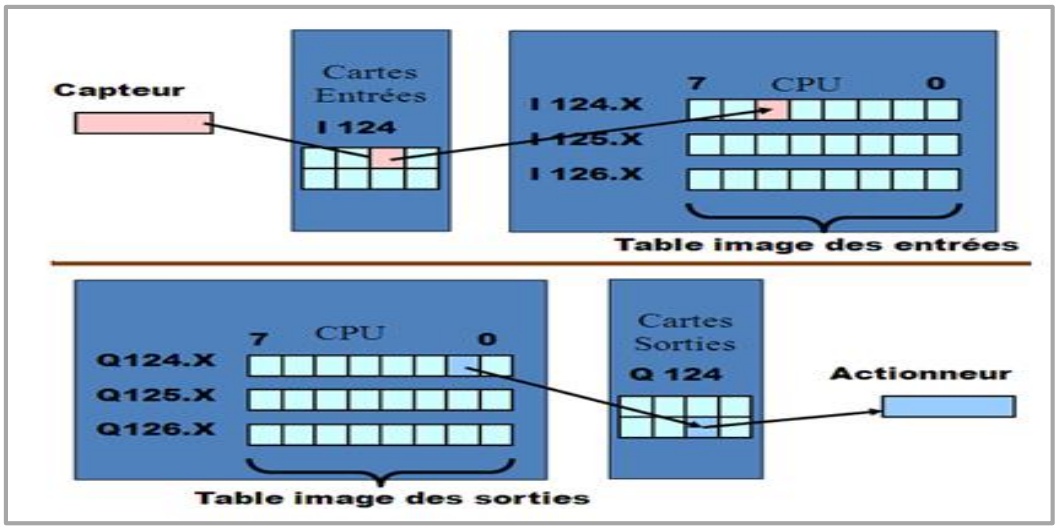


Figure II.5 : Interfaces des E/S d'un API

a) Cartes d'entrées

Elles sont destinées à recevoir l'information en provenance des capteurs et adapter le signal en le mettant en forme, en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative.

b) Cartes de sorties

Elles sont destinées à commander les pré-actionneurs et éléments des signalisations du système et adapter les niveaux de tensions de l'unité de commande à celle de la partie opérative du système en garantissant une isolation galvanique entre ces dernières.

- ✓ **Une alimentation électrique**

Elle a pour le rôle de transforme la tension du réseau en tension stable pour le bon fonctionnement de l'unité centrale, des modules d'entrées/sorties et des mémoires notamment face aux microcoupures de réseau électrique qui constitue de la source d'énergie, un onduleur est nécessaire pour évite le risque de coupure non tolérées, la tension d'alimentation peuvent être de 5V, 12V ou 24V **[22].**

- ✓ **Liaisons de communication**

L'interface de communication est utilisée pour recevoir et transmettre des données sur des réseaux de communication qui relient l'API a d'autres API distants. File est impliquée dans des opérations telles que la vérification d'un périphérique, L'acquisition de données, la synchronisation entres des applications et la gestion de la connexion (**Figure II.6**) **[21].**

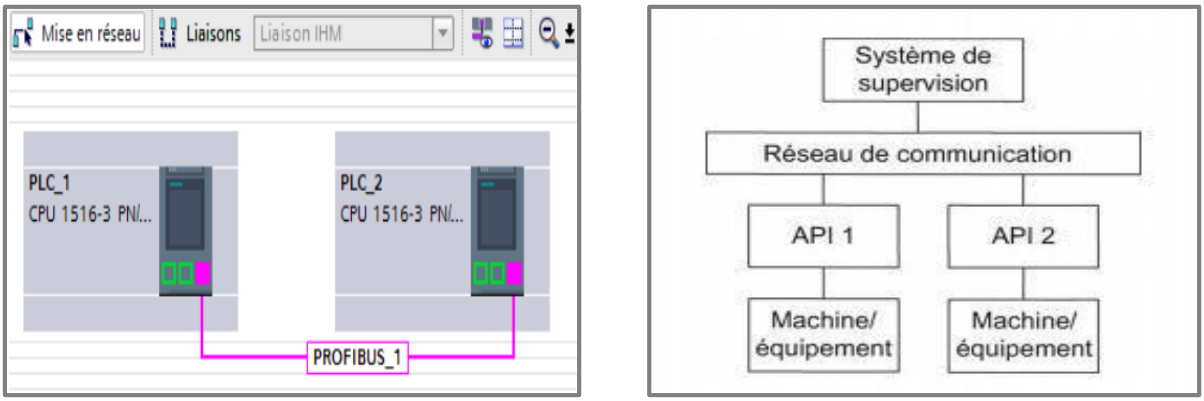


Figure II.6: Modèle de base des communications.

II.3.4. Modules d'entrées et sorties TOR (Tout Ou Rien)

- **Modules d'entrées TOR (Tout Ou Rien)**

L'automate reçoit ses informations sur le processus via les capteurs de signaux reliés aux entrées. Les modules d'entrée TOR permettent de recevoir les signaux des différents capteurs logiques qui peuvent être des détecteurs qui reconnaitront si la pièce d'usinage se trouve à une position donnée (détecteurs des niveaux -haut et bas-) ou de simples commutateur ou interrupteur qui peuvent être fermés ou ouverts. Ce qui fait que l'information délivrée par ces capteurs et qui sera traitée par la CPU ne peut prendre que deux valeurs 0 ou 1.**[17]**

- **Modules de sorties TOR (Tout Ou Rien)**

Ces modules permettent de délivrer des signaux qui permettent à l'automate d'agir sur les pré-actionneurs du système à commander tels que (Vanne Electromagnétique, Electrovannes, contacteur, pompes et Voyants…)

II.3.5. Modules d'entrées et de sorties Analogiques

- **Modules d'entrées Analogiques**

Les Modules des entrées analogiques permettent de traiter des signaux issus par exemple de capteur de niveau, de pression, de température....etc. elles permettent de prendre en compte des signaux de tension ou de courant évoluant de façon continue. Ces interfaces convertissent le signal analogique en signal numérique et isolent ces signaux des bus internes de l'automate.**[23]**

- **Modules de sorties Analogiques**

Les Modules des sorties analogiques permettent de raccorder par exemple un régulateur de pression ou une vanne de chauffage. Ces interfaces isolent les signaux numériques issus des bus internes de l'automate et convertissent ces informations numériques en signaux analogiques de type tension ou courant compatible avec des régulateur de pression, vanne de chauffage....etc.

II.3.6. Nature Des Informations Traitées Par L'automate

Les informations peuvent être de type :

- **Tout ou rien (T.O.R.)** : l'information ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type d'information délivrée par un détecteur, un bouton poussoir ...**[24]**

- **Analogique :** l'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température ...)

- **Numérique :** l'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

II.3.7.Les différents modèles de l'API SIEMENS S7

La gamme automate de Siemens contient 5 types d'automates :

SIMATIC S7-200 : Solution séquentielle simple, performante en terme de temps réel et de communication.

SIMATIC S7-300 : Solution séquentielle complexe. Il permet de réaliser la majorité des applications d'automatisme intégrant des architectures décentralisées**.**

SIMATIC S7-400 : Solution séquentielle complexe, hautes performance en terme de communication et de mémoire.

SIMATIC S7-1200 : Solution séquentielle simple, mais précise.

SIMATIC S7-1500 : C'est la dernière gamme d'automates Siemens. Il se programme sous TIA Portal et dispose d'un petit écran de façade permettant de faire quelques configurations basiques.

Figure II.7: Différents modèles de L'A.P. I SIEMENS S7

Nous choisissons l'automate programmable S7-300 comme un API pour commander les déférentes opérations de la soudeuse.**[25]**

II.4.Présentations de l'automate S7-300

II.4.1. Introduction

L'automate S7-300 est un mini automate modulaire pour des applications d'entrées et de milieu de gamme fabriqué par SIEMENS, on peut le composer en fonction de nos besoins à partir d'un vaste éventail de modules. SIMATIC S7 désigne un produit de la société SIEMENS sont des appareils fabriqués en série, conçus indépendamment d'une tache précise. Tous les éléments logiques, fonction de mémoire, temporisation, compteur …etc, nécessaire à l'automatisation sont prévus par le fabricant et sont intégrés dans l'automate. Ils se distinguent principalement par le nombre des :

- ✓ Entrées et sorties.
- ✓ Compteurs.
- ✓ Temporisation.
- ✓ Mémentos. **[26]**

Figure II.8 : Automate S7-300

II.4.2. Constitution de l'Automate S7-300 :

L'automate S7-300 possède :

- Des CPU de différents niveaux de performances
- Des Modules de signaux pour Entrées/Sorties « TOR » et analogique, ainsi que des Modules de fonction pour les différentes fonctions technologiques
- Une possibilité de mise à niveau par MPI
- Une largeur réduite des Modules, permettant un gain de place au montage
- Une structure compacte, lui permettant le placement aux milieux exigus.

II.4.3. Caractéristiques de l'automate S7-300 :

- Gamme diversifiée de la CPU.
- Gamme complète du module.
- Possibilité d'exécution jusqu'à 32 modules.
- Possibilité de mise en réseaux avec MPI, PROFIBUS ou PROFINET.
- Liberté de montage au différent emplacement.

II.4.4. Modularité du S7-300

Parmi les caractéristiques essentielles du S7-300, le fait qu'il est disposé d'une vaste gamme de Modules.

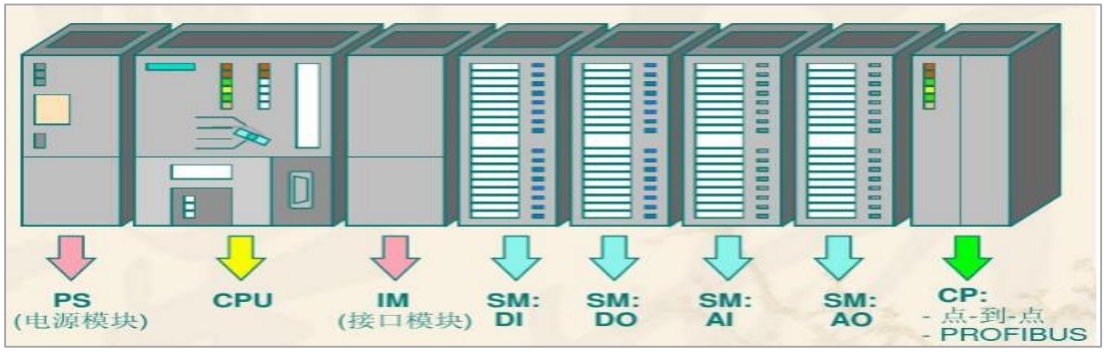


Figure II.9 : Disposition des modules de l'API S7-300.

a/ Module d'alimentation (PS) :

Le S7-300 peut être alimenté avec une tension de 24 VDC, cette dernière est assurée via le module d'alimentation par conversion de la tension Secteur 120V/230 VAC.

Désignation	CS	Tension à la sortie	Tension à la l'entrée
PS307	2A	DC24v	AC120V/230v
PS307	5A	DC24v	AC120V/230v
PS307	10A	DC24v	AC120V/230v

Tab II. 1: modules d'alimentation.

Figure II.10 : Différentes module d'alimentation SIMATIC S7-300.

b/ Unité centrale (CPU) :

C'est une carte électronique construite autour d'un ou plusieurs processeurs et mémoire. La CPU possède un système d'exploitation, une unité d'exécution et des interfaces de communication (MPI, PROFIBUS, PROFINET). Essentiellement la CPU lit l'état des signaux d'entrée et exécute le programme utilisateur séquentiellement.

Figure II.11 : CPU SIMATIC S-300

c/ Interfaces MPI :

Chaque CPU est équipée d'une interface MPI (Multi Point Interface) pour la connexion de la console de programmation (PG) ou un appareil par exemple adaptateur PC.

Figure II.12 : Câble de programmation pour automate S7-300.

d/ Carte mémoire :

La CPU peut être équipée d'une micro carte mémoire, permettant la sauvegarde du programme en cas de coupure du courant**.**

Figure II.13 : Micro carte mémoire SIMATIC

II.4.5. Signalisation des états

Sur la face avant de la CPU, on trouve des LEDS permettant à l'automate de signaler certains états, tel que :
- **RUN** (LED verte).
- **STOP** (LED jaune).
- **SF** (Erreur matérielle ou logicielle, LED rouge).
- **BF** (Erreur de bus, LED rouge).
- **MAINT** (Requête de maintenance, LED jaune).

II .4.6. Sélecteur de mode de fonctionnement

Le sélecteur de mode permet de régler le mode de fonctionnement de la CPU, tel que :
- **RUN** (La CPU traite le programme utilisateur).
- **STOP** (La CPU ne traite aucun programme utilisateur).
- **MRES** (position dans laquelle en effacement général de la CPU peut être effectué).

L'automate S7 est constitué d'une alimentation (Modules PS), d'une CPU ainsi que des modules d'entrées / sortie, Siemens fournit des **:**

II.4.7. Module de fonction (FM)

Ces modules réduisent la charge de traitement de la CPU en assurant des tâches lourdes de calculs. On peut citer les modules suivants : Module de commande d'axe pour servomoteur, Module de positionnement pour moteur pas à pas, Module de régulation, Module de comptage…**[25]**

II.4.8. Module de communication (CP)

Le module de communication Assurant une communication par transmission série, comme ils peuvent aussi établir des liaisons point à point avec d'autres automates SIMATIC S7-300 ou bien d'autres constructeurs...

I.4.10.Coupleur (IM)

Les coupleurs sont des cartes électroniques qui assurent la communication entre les E/S (périphéries ou autre) et l'unité centrale. L'échange de l'information entre la CPU et les modules d'E/S s'effectue par l'intermédiaire d'un câble PROFIBUS.

II.4.11.Le châssis (rack)

Les châssis constituent des éléments mécaniques de base du SIMATIC S7-300 (Assemblage mécanique des modules).

II.4.12.Fonctionnalités

Le S7-300 est une plate-forme d'automatisation universelle pour des applications avec des architectures centralisées et décentralisées, orientée sécurité, motion control ou avec interface

Ethernet/PROFI net intégrée. Le S7-300 peut également s'intégrer dans des solutions compactes avec HMI ou dans des têtes de station pour traitement intelligent décentralisé.

II.4.13. Câblage de l'automate S7-300

- **Câblage des entrées**

Le principe de raccordement consiste à envoyer un signal électrique vers l'entrée choisie sur l'automate dès que l'information est présente.

Un API peut être à logique positive ou négative :

- **Logique positive :** le commun interne des entrées est relié à **24 V.**
- **Logique négative :** le commun interne des entrées est relié à **0 V.**

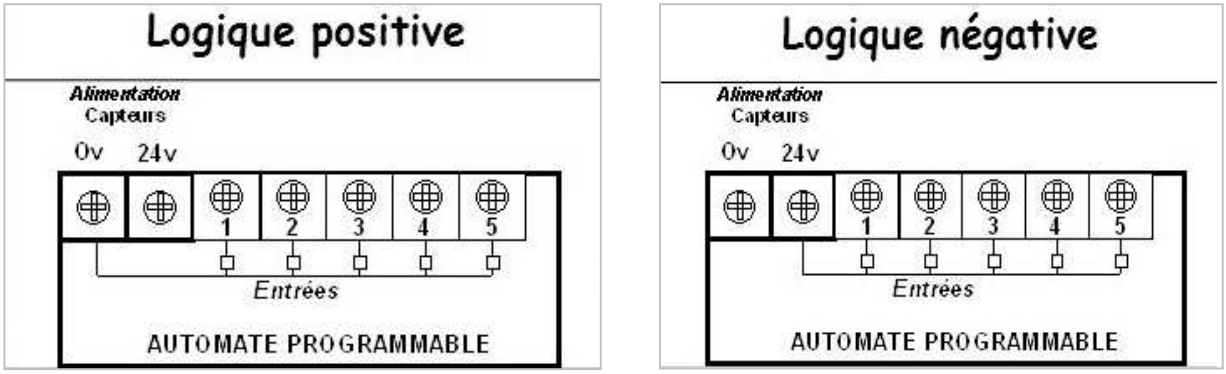


Figure II.14 : Câblage des entrées.

- **Câblage des sorties**

Chaque sortie de l'automate est constituée d'un relais dont la fermeture de contact est commandée par la consigne opérative élaborée par le programme qui permet l'alimentation de la bobine de précautionner en établissant un circuit électrique avec l'alimentation extérieure **[18].**

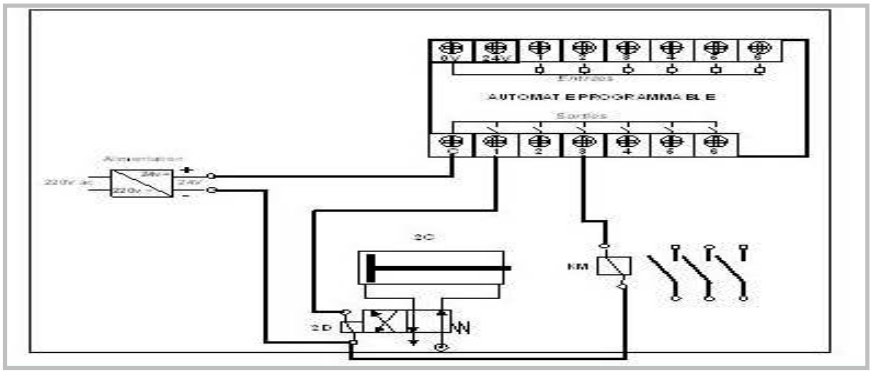


Figure II.15: Câblage des sorties.

II.5. Programmation de l'automate SIEMENS S7-300

II.5.1. Définitions du logiciel STEP7

Step7 est le logiciel de base qui permet la configuration et la programmation des systèmes d'automatisation SIMATIC. Il s'exécute sous un environnement Windows, à partir d'une console de programmation ou d'un PC. Il existe plusieurs versions : STEP 7 micro/ Win pour les applications S7-200 et STEP 7 pour les applications S7-300 et S7-400**.**

La programmation en STEP 7 présente quatre modes de représentation qui peuvent être :

- **Le schéma à contacts (CONT) :** est un langage de programmation graphique. La syntaxe des instructions fait penser aux schémas de circuits électriques. Le langage CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts, les éléments complexes et les bobines .

Fonction	Symbole	
	Européen	Américain
Contact ouvert au ropos	---o o---	⊣⊢
Contact fermé au ropos	---o o---	⊣/⊢
Début de branchement	⊓	⊓
Fin de branchement	⊔	⊔
Affectation	---( )---	---( )

Figure II.16: Usual symbols in LD languages.

- **La liste d'instructions (LIST):** est un langage de programmation textuel proche de la machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes par les quelles la CPU traite le programme .

INSTRUCTIONS BOOLEENNES

LD	Charge un résultat booléen (commence une phrase)
LDN	Charge le complément
LDF	Charge le front montant
LDR	Charge le front descendant
AND	Et
OR	Or
ANDN	Et pas
ORN	Ou pas
ANDF	Et front montant
ORF	Ou front montant
ANDR	Et front descendant
ORR	Ou front descendant
XOR	Ou exclusif

Figure II.17: fonction et instructions d'action de langage IL.

- **ST (Structured Text) :** Ce langage textuel de haut niveau est un langage évolué. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

```
1  IF #start = 1 THEN
2      //comment
3      "Max_nr" := #Array[0];
4      FOR #i := 1 TO 10 DO
5          // Statement section FOR
6          IF #Array[#i] > "Max_nr" THEN
7              "Max_nr" := #Array[#i];
8          END_IF;
9      END_FOR;
10 END_IF;
11
```

Figure II.18 : exemple Langage littéral structuré

4/ Langage Grafcet : Issu du langage GRAFCET, ce langage de haut niveau permet la programmation aisée de tous les procédés séquent**[20].**

II.5.2. L'adressage absolu des modules de signaux

Chaque entrée et sortie du module de signaux possède une adresse absolue déterminée par la configuration matérielle.

a) Adressage des modules TOR :

- L'adressage d'une entrée ou d'une sortie est constitué d'une adresse d'octet et d'une adresse de bit.
- L'adressage d'octet dépend de l'adresse de début de module.
- L'adressage de bit est indiqué sur le module.

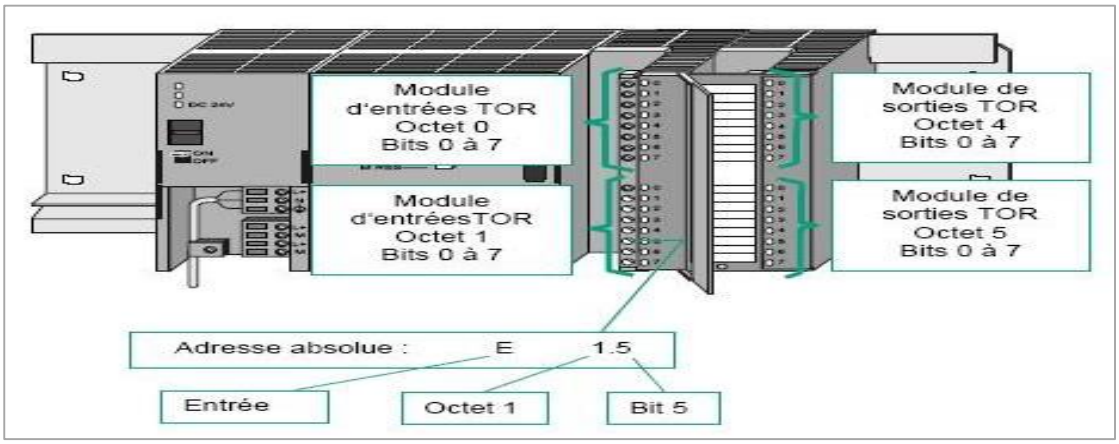


Figure II.19 : Exemple d'adressage absolu d'un module TOR.

b) Adressage des modules analogiques :

L'adresse d'une voie d'entrée ou de sortie analogique est toujours une adresse de mots. L'adresse de voie est basée sur l'adresse initiale des modules. L'adresse initiale de chaque module analogique suivant augmente de 16 bits par emplacement.

II.5.3. Structure du programme

Les automates Siemens sont orientés « programmation structurée ». Cela signifie que le Programme utilisateur peut être découpé en blocs qui sont eux même découpés en segments. Sur les automates Siemens il existe différents types de blocs programmes : les blocs OB, FB,FC et DB. STEP 7 offre les blocs utilisateur suivants pour la programmation structurée **:**

a/ Bloc d'organisation (OB) :

Un OB est appelé cycliquement par le système d'exploitation et constitue donc l'interface entre le programme utilisateur et le système d'exploitation. L'OB contient des instructions d'appels de blocs indiquant à l'unité de commande de l'automate l'ordre dans lequel il doit traiter les blocs.

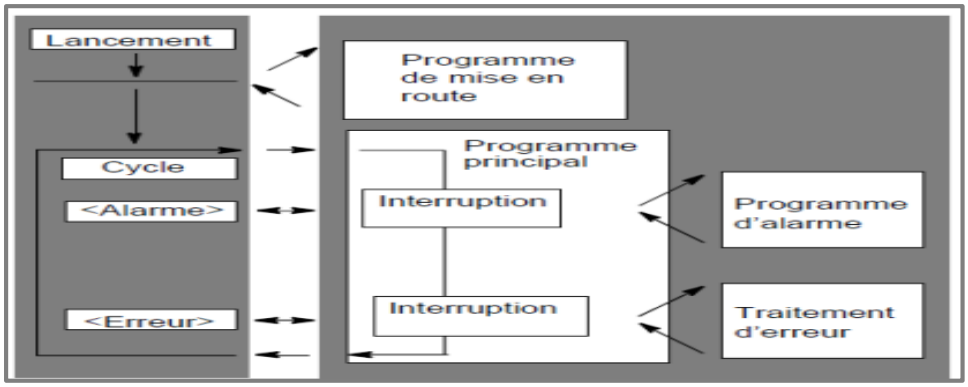


Figure II.20 : Traitement du programme avec possibilité d'interruption.

- **Programme cyclique OB 1**

Lors d'une exécution normale de programme, les traitements se font de façon cyclique. L'exécution du programme contenu dans l'OB 1 est démarrée une fois par cycle (quand il est fini, il recommence). On peut se servir de l'OB 1 pour appeler des blocs de type FC ou FB.

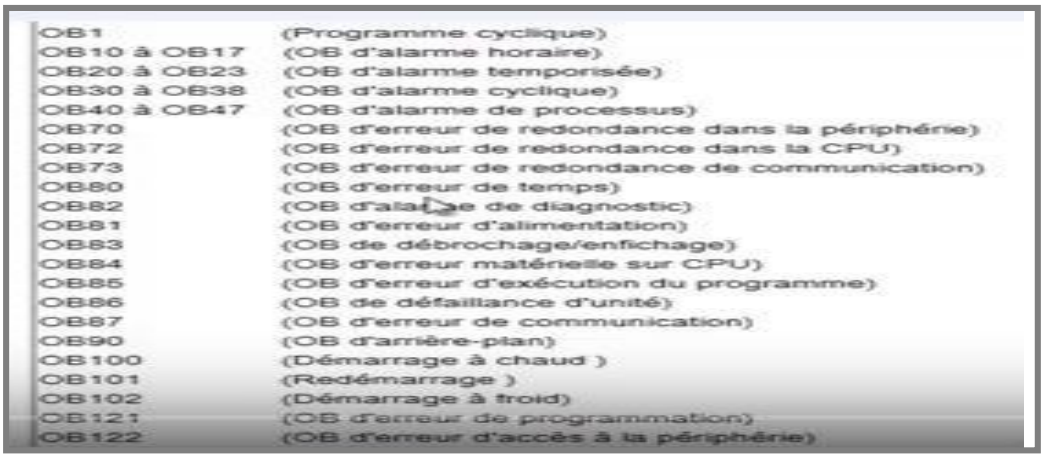


Figure II.21 : Liste des Blocs d'organisation OB disponible sur l'API S7-300

b/ Bloc fonctionnel (FB) :

Le FB dispose d'une zone de mémoire qui lui est affectée en propre. Il est possible d'affecter un bloc de données (DB) au FB à l'appel du bloc. Il est possible d'accéder aux données du DB d'instance via les appels contenus dans le FB. On peut affecter plusieurs DB à un FB. Il est possible d´appeler d'autres FB et FC dans un bloc fonctionnel via des instructions d'appels de blocs.

c/ Fonction (FC) :

Une FC contient des routines pour les fonctions fréquemment utilisées. Elle est sans mémoire et sauvegarde ses variables temporaires dans la pile de données locales. Cependant elle peut faire appel à des blocs de données globaux pour la sauvegarde de ses données.

d/ Bloc de données (DB) :

Les DB sont utilisés pour la mise à disposition d'espace mémoire pour les variables types données. Il existe deux types de blocs de données. Les DB globaux dans lesquels tous les OB, FB et FC peuvent lire les données enregistrées ou écrire des données et les DB d'instance qui sont affectés à un FB donné.

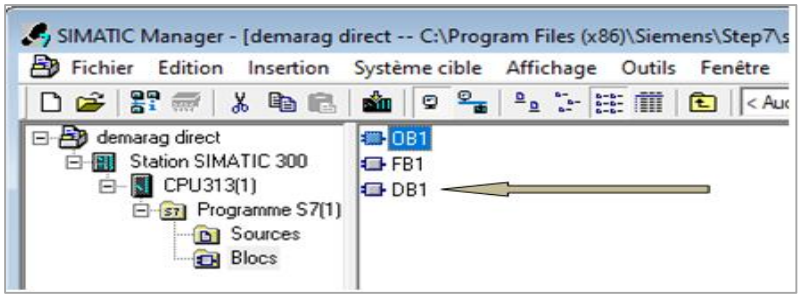


Figure II.22 : Bloc de données DB1.

I.6. Création d'un projet STEP7

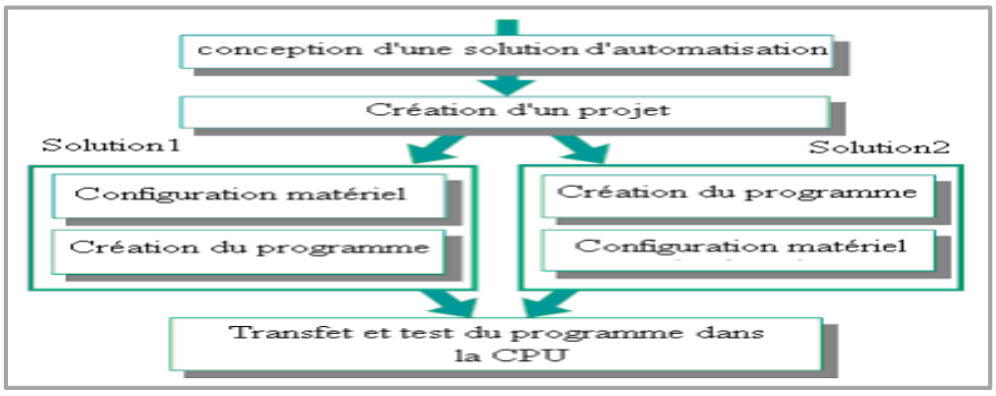


Figure II.23 : Schéma illustrant les deux solutions possibles pour la programmation

Les procédures qui permettent la création de projet sous logiciel STEP7 sont comme suit :

1- double clic sur l'icône SIMATIC Manager; ceci lance l'assistant de STEP7.
2- La fenêtre illustrée en (**Figure II.23**) apparaît, elle permet la création d'un nouveau projet.

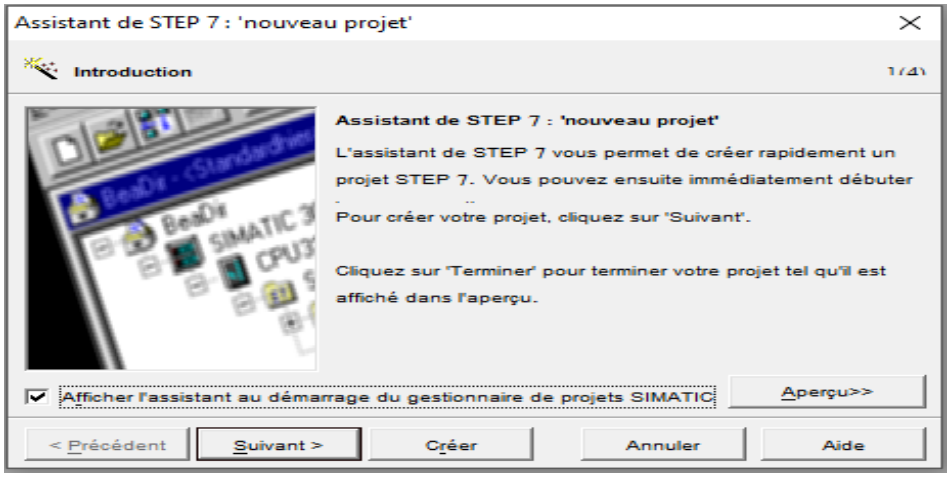


Figure II.24 : Assistant de STEP 7 : nouveau projet.

3- En cliquant sur l'icône suivant, la fenêtre suivante, nous permet de choisir la CPU.

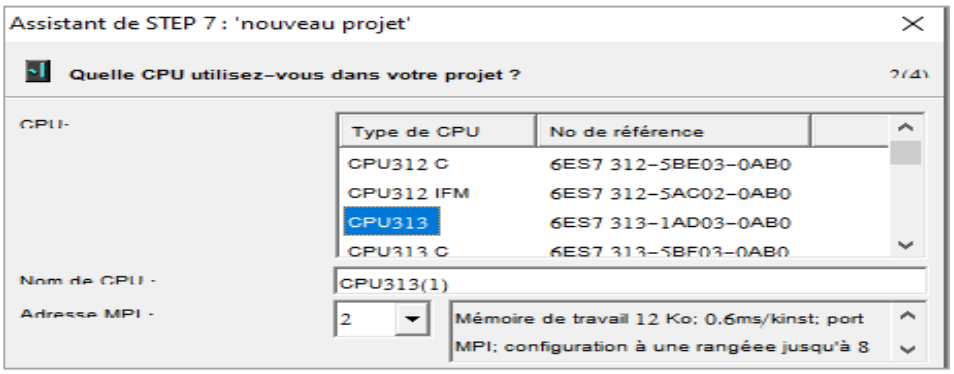


Figure II.25: Fenêtre de choix de la CPU.

4- Après validation de la CPU, la fenêtre qui apparaît permet de choisir les blocs à insérer, et choisir le langage de programmation (LIST, CONT, LOG).

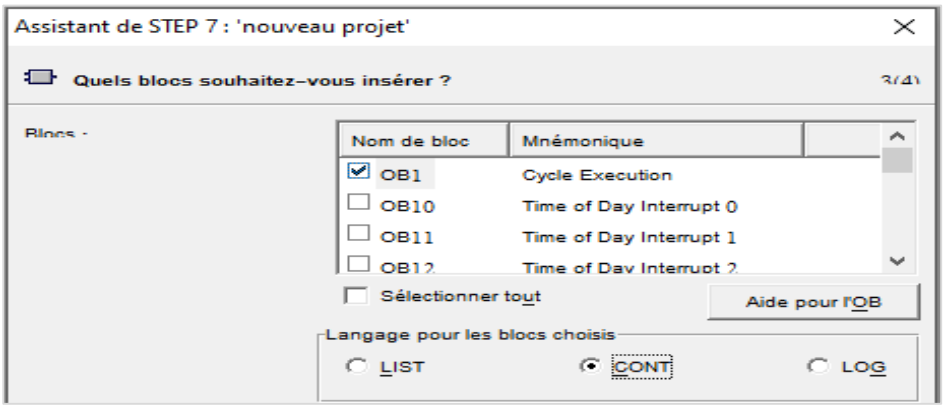


Figure II.26 : Choix des Blocs à utilisés et de langage.

5- En cliquant sur suivant, la création de projet apparaît pour le nommer.

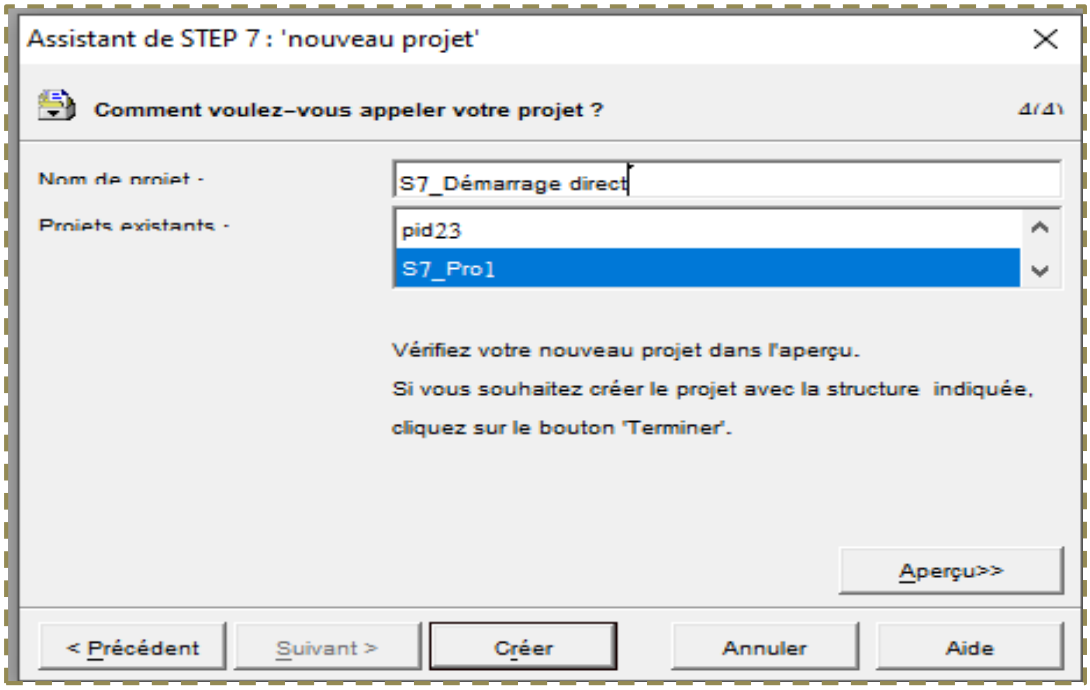


Figure II.27: Nomination du projet.

6-On clique sur créer, la fenêtre suivante apparaît.

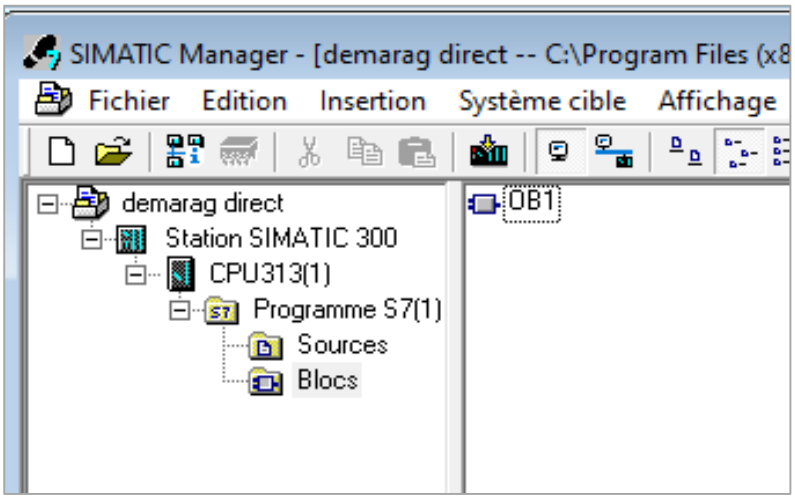


Figure II.28: Répertoire de la station SIMATIC et de la CPU.

7- Configuration matérielle : La configuration matérielle est une étape importante. Elle consiste à disposer les châssis (rack), les modules et les appareils de la périphérie centralisée. Les châssis sont représentés par une table de configuration dans laquelle on peut placer un nombre définis de modules comme dans les châssis réels**.**

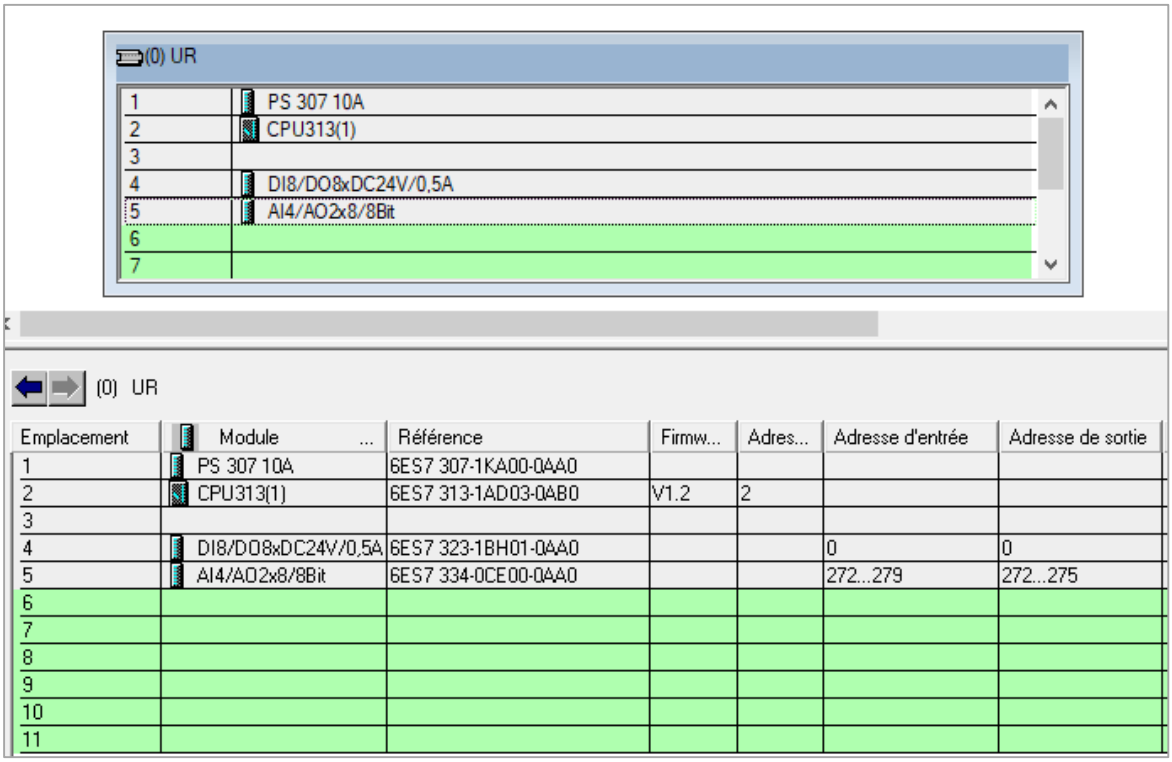


Figure II.29 : Configuration du matériel.

8- Table des Mnémoniques : Une mnémonique est un nom que l'utilisateur définit en respectant les règles de la syntaxe imposée. Il est destiné à rendre le programme lisible et aide donc à gérer facilement le grand nombre de variables couramment rencontrées dans ce genre de programme. Ce nom qu'on a donné à l'adresse pourra être utilisé directement dans le programme une fois les affectations terminées.

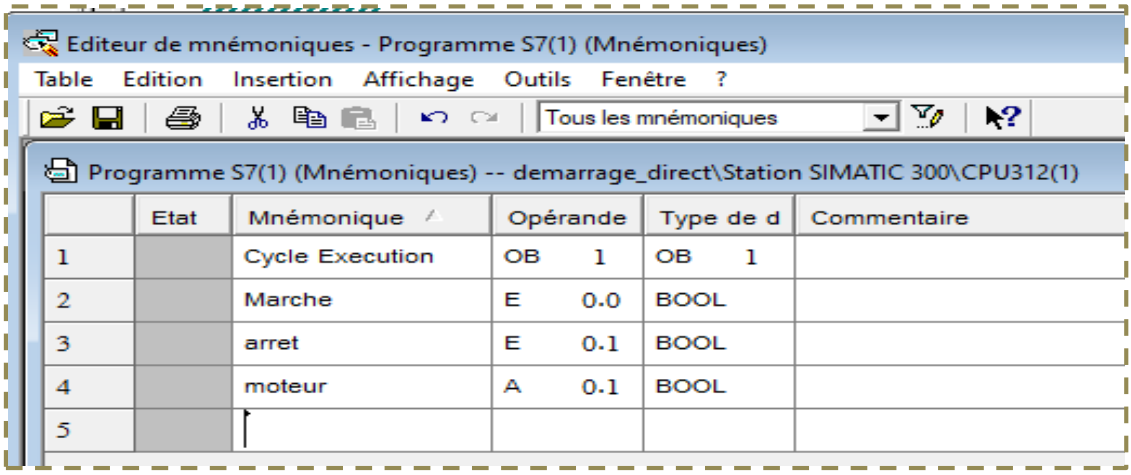


Figure II.30 : La table de mnémoniques (activer et désactiver d'un moteur).

9- création du programme sur le bloc OB1 :

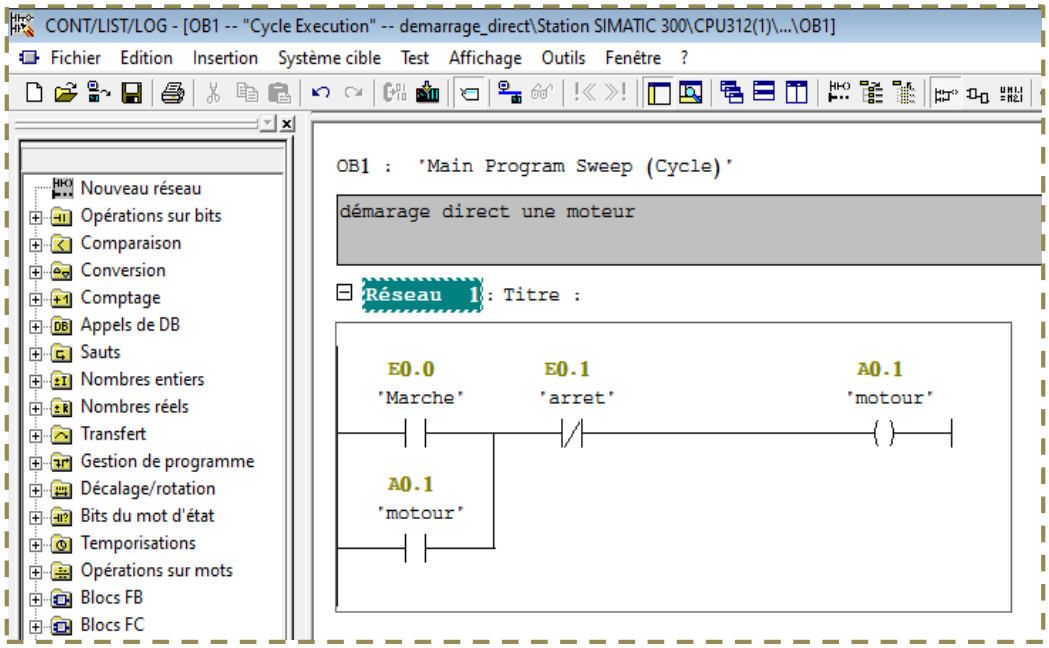


Figure II.31 : Création du programme (activer et désactiver d'un moteur).

II.6.1. Présentation du S7-PLCSIM

L'utilisation du simulateur de modules physiques S7-PLCSIM nous permet d'exécuter et de tester le programme dans un automate de simulation que nous simulons dans un ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7. Le S7-PLCSIM dispose d'une interface simple nous permettant de visualiser et de forcer les déférents paramètres utilisés par le programme (comme activer ou désactiver des entrées.). Tout en exécutant le programme dans l'API de simulation, nous avons également la possibilité de mettre en œuvre les diverses applications du logiciel STEP7 comme, par exemple, le test de bloc afin de visualiser les variables d'entrées et de sorties.

II.6.2. Mise en route du logiciel S7-PLCSIM

Le mode de simulation est disponible à partir du gestionnaire de projet SIMATIC. On peut suivre la procédure suivante pour la mise en route du logiciel S7-PLCSIM.

1- Ouvrir le gestionnaire de projet SIMATIC.
2- Cliquez sur ou sélectionnez la commande Outils > simulation de modules.

Cela lance l'application S7-PLCSIM et ouvre une fenêtre CPU

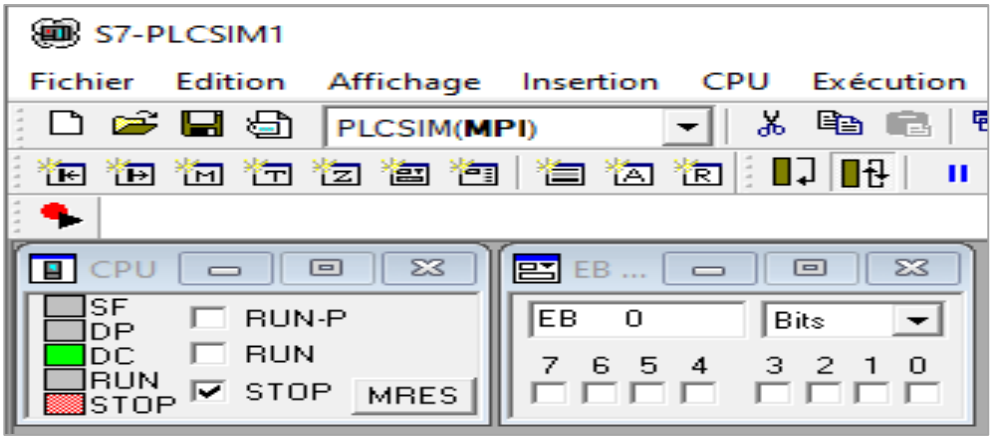


Figure II.32 : Fenêtre du S7-PLCSIM

3- Dans le bloc OB1 cliquez sur ou choisir la commande Système cible (charger) pour charger le dossier blocs dans l'API de simulation.

4- Dans l'application S7-PLCSIM, on crée de nouvelles fenêtres pour visualiser les informations provenant de l'API de simulation :

5- Cliquez sur  ou choisissez la commande Insertion (Entrée) pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des entrées. Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut IB0. Mais on peut modifier l'adresse (IB1, IB2…)

6- Cliquez sur  ou choisissez la commande Insertion (Sortie) pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des sorties. Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut QB0. Mais on peut modifier l'adresse (QB1, QB2…).

7- Cliquez sur  ou choisissez la commande Insertion (Mémento) pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des mémentos. Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut MB0. Mais on peut modifier l'adresse (MB1, MB2…)

8- Cliquez sur  ou choisissez la commande Insertion (Temporisation) pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer les temporisations utilisées par le programme. Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut T0.

9- Choisir le menu CPU dans la fenêtre du s7-PLCSIM et vérifier que la commande à Mettre sous tension est activée.

10- Choisir la commande Exécution (Mode d'exécution) et vérifier que la commande cycle continue est activée.

7- Mettre la CPU de simulation en marche en cliquant sur l'une des cases à cocher RUN ou RUN-P

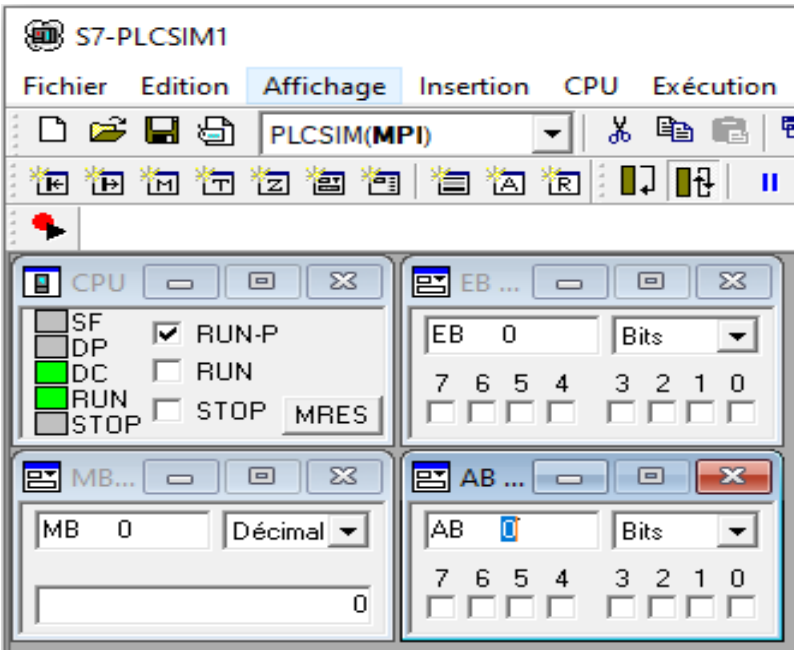


Figure II.33: Simulator S7-PLCSIM.

8- Après le chargement du programme  dans la CPU du simulateur et la mise de cette dernière en mode « RUN » le STEP 7 nous permet de visualiser l'état du programme soit en cliquant sur l'icône  ou on sélectionnant la commande Test > Visualiser.

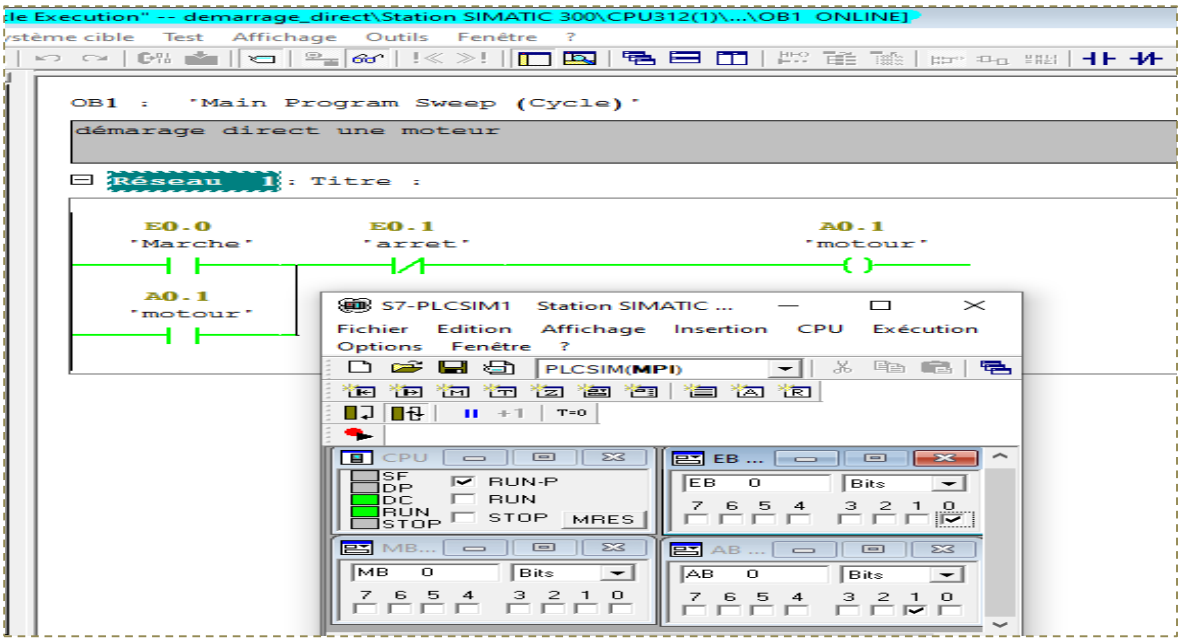


Figure II.34: Visualisation de l'état du programme (activer et désactiver un moteur DC).

II.7. Conclusion

Ce chapitre décrit les Automates Programmables Industriels (API) et leur rôle dans l'automatisation des systèmes industriels. En outre, illustre comment les API ont été introduites pour remplacer les armoires à relais utilisées pour l'automatisation des chaînes de fabrication par des équipements moins coûteux et plus faciles à modifier.

Ce chapitre décrit également les objectifs de l'automatisation d'un système, ainsi que les avantages de l'utilisation d'API dans l'industrie.

Et finalement, il explique comment les méthodes de programmation basées sur la standardisation des langages de programmation sont utilisées avec les API pour assurer l'intégrité de la chaîne de production et la sécurité des personnes.

Chapitre III

Régulation de débit et niveau d'un système hydraulique

III.1. Introduction

Le régulateur standard le plus utilisé dans l'industrie est le régulateur PID (proportionnel, intégral, dérivé), car il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse) d'une régulation d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui ont un comportement voisin de celui de deuxième ordre dans une certaine échelle de temps. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et il est relativement robuste par rapport aux variations des paramètres de procédé, quand on n'est pas trop exigeant pour les performances de la boucle fermée par rapport à celles de la boucle ouverte (par exemple, accélération très importante de la réponse ou augmentation très importante de l'amortissement en boucle fermée. **[27]**

III.1.1. Notion de Boucle Ouverte/Fermée

III.1.2. Système boucle ouverte

On parle de fonctionnement en boucle ouverte quand c'est l'opérateur qui contrôle l'organe de réglage, ce n'est pas une régulation.

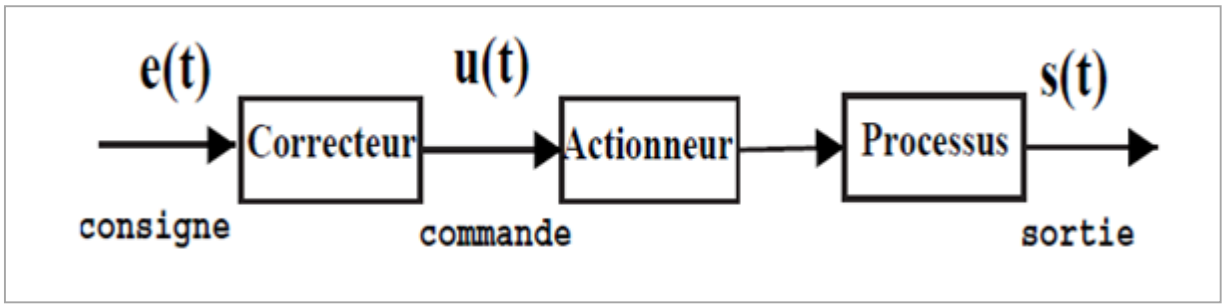


Figure III.1 :Système en BO.

III.1.3. Système boucle fermée

C'est le fonctionnement normal d'une régulation. Le régulateur compare la mesure de la grandeur réglée et la consigne et agit en conséquence pour s'en rapprocher.**[28]**

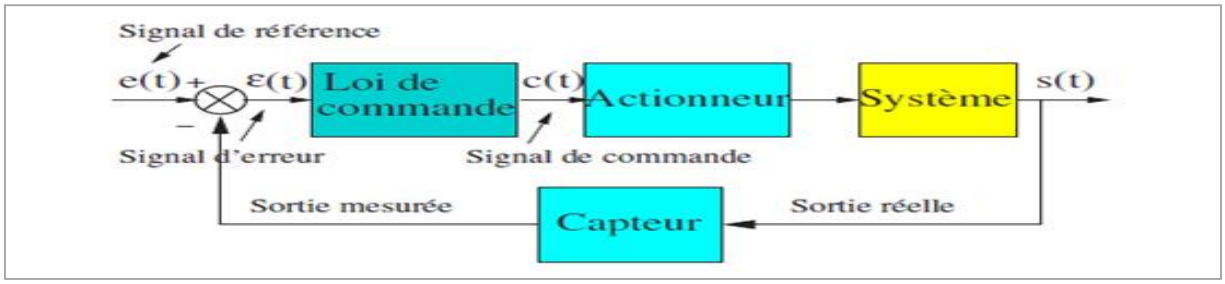


Figure III.2 :Système en BF.

III.1.4. Conception d'un système de commande

La commande d'un processus consiste à détermine la commande appropriée, de manière à assurer aux variables à contrôler (sorties) un comportement défini. L'action de la commande Est une action susceptible de changer l'état du système à commander. Ces commandes sont Délivrées par un organe de commande ; le processus et son organe de commande constituent le système de commande. Le système de commande comprend un élément nécessaire qui est le régulateur qui effectue le calcul de la commande à appliquer au processus à partir de la consigne et de l'état du processus. Lorsqu'il y a un retour d'information de la grandeur observée sur le régulateur, on parle d'un asservissement du système ou d'une régulation du système. **[29]**

a. Régulation :

Dans une régulation, on s'attachera à maintenir constante la grandeur réglée d'un système soumis à des perturbations.**[30]**

b. Asservissement :

Dans un asservissement, la grandeur réglée devra suivre rapidement les variations de la consigne

III.1.5. Constituants d'une chaine de régulation

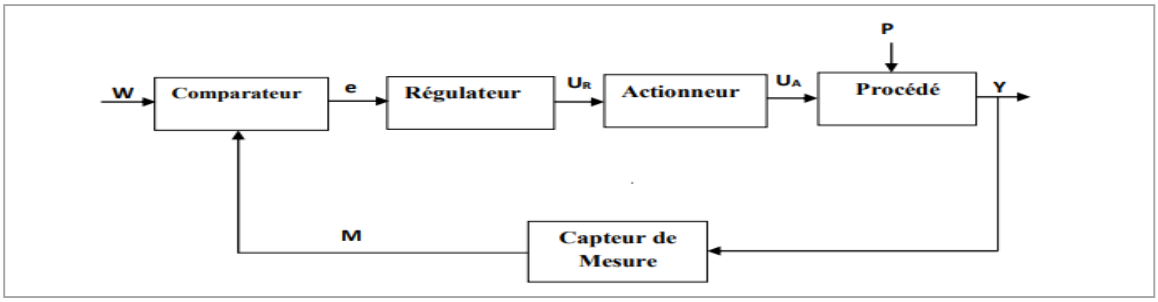


Figure III.3 : Représentation fonctionnelle d'une boucle de régulation.

D'une manière générale, une boucle de régulation peut être représentée de la manière suivante :
Cette organisation fonctionnelle représente la structure de base qu'on trouve dans tous les Systèmes asservis ou régulés. Elle fait intervenir deux chaînes : une chaîne d'action et une chaîne de retour ou d'informations.

- La chaîne d'action englobe tous les organes de puissance (nécessitant un apport extérieur d'énergie) et qui exécute le travail.
- La chaîne de retour ou de mesure : Analyse et mesure le travail effectué et transmet au comparateur une grandeur physique proportionnelle à ce travail. Elle comprend Généralement un capteur qui donne une mesure de la grandeur, qui est ensuite Amplifiée et transformée avant d'être utilisée. **[29]**

a. Consigne « W »

La consigne est la grandeur qui doit commander la sortie, c'est-à-dire la valeur vers laquelle celle-ci doit tendre, pour finalement l'égaler. La consigne n'a pas obligatoirement la même dimension que la sortie, mais doit être en accord avec la dimension de la mesure.**[29]**

b. Sortie « Y »

Elle est le résultat de la régulation, c'est la variable que le système va influer et/ou essayer de garder constante.

c. Retour « M »

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

d. Perturbation « P »

La perturbation est la grandeur qui influe de manière indésirable sur la sortie et qui l'éloigne de la valeur souhaitée (consigne). La seule existence d'une perturbation rend nécessaire la mise en œuvre d'une régulation statique. **[29]**

e. Comparateur

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

f. Régulateur

Le régulateur est l'élément central d'une régulation. Il évalue l'erreur calculée par le comparateur, c'est-à-dire l'écart entre la sortie et la consigne, et en déduit à partir de celle–ci une valeur régulée « UR » ou valeur de correction à transmettre au procédé, afin de corriger la sortie.

La façon (algorithme) avec lequel le régulateur calcule la valeur régulée à partir de l'erreur est la principale activité de la régulation.

g. Actionneur

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

h. Procédé

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

III.2. Types de régulation automatique

III.2.1. Régulation tout ou rien (TOR)

Ce mode d'action est essentiellement discontinu. Sa réalisation impose de se fixer une limite inferieure et une limite supérieur lorsque la mesure atteint la limite inferieure l'actionneur prend une position particulier (ouverture ou fermeture pour une vanne) de façon analogue le fait d'atteindre la limite supérieure place l'actionneur dans la position contraire. Ça oscille donc entre ces deux valeurs extrêmes et sa variation prend une allure en dents de scie. Ce réglage est simple ou le signal de commande ne prend que deux valeurs soit 0 ou 1. **[31]**

III.2.2. Régulation analogique

C'est le type de régulation ou le signal du régulateur et la mesure variant d'une manière continue dans le temps. Le mode d'action analogique le plus simple est l'action proportionnelle est réalisée par un régulateur (p), il convient en générale aux installations ayant une grande inertie.

III.2.3. Régulation numérique

Le principe de la régulation numérique est que le régulateur prend la forme d'un algorithme programmé sur microprocesseur et exécuté en temps réel, i.e. impérativement à chaque période d'échantillonnage. **[32]**

III.2.4. Critère de performance d'une régulation

Les performances d'une régulation peuvent se définir à partir de l'allure du signal de mesure suite a un échelon de consigne. Notons toutefois que les critères de performances classiques peuvent se résumer comme suit **:**

a. Stabilité :cette condition est impérative avec un certain degré de stabilité (marge de sécurité).en générale on impose une marge de gain de 2 a 2.5 L'utilisateur parle en termes de pompage.

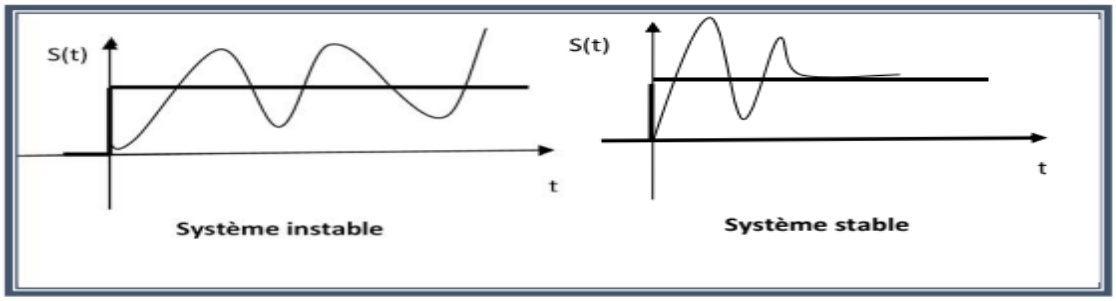


Figure III.4 : Stabilité du système.

b. Précision : l'exploitant demande à ce que le système possède une bonne précision en régime permanent d'où une nécessite de mettre un régulateur **PI** ou d'afficher un gain important dans le cas d'un régulateur **P**.

c. Rapidité : on demande en pratique que le système soit capable rapidement de compenser les perturbations et de suivre la consigne. **[32]**

III.3. Principe de la commande PID

L'intérêt du correcteur PID est d'intégrer les effets positifs des trois correcteurs précédents. La détermination des coefficients Kp ,Ti et Td du correcteur PID permet d'améliorer à la fois la précision (Ki et Kp) ; la stabilité (Td) ; la rapidité (Td,Kp).Le réglage d'un PID est en général assez complexe, des méthodes pratiques de réglages permettent d'obtenir des bons résultats. **[33]**

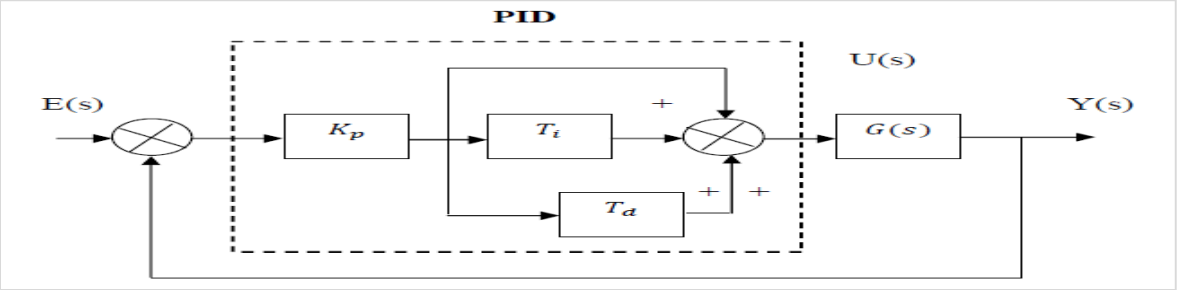


Figure III.5: Structure d'un correcteur PID «standard».

commande du régulateur PID :

$$\boldsymbol{u(t) = K_P * (e(t) + \frac{1}{T_i} * \int_{-\infty}^{t} e(\tau) * T_d * \frac{de}{dt}} \qquad \textbf{(III. 1)}$$

Fonction de transfert du régulateur PID :

$$\boldsymbol{G_c(P) = \frac{U(P)}{E(P)} = K_p * \frac{1+P*T_i+P*T_i*T_d}{P*T_i}} \qquad \textbf{(III. 2)}$$

III.3.1. Le rôle des paramètres PID

P	L'action proportionnelle applique une correction instantanée pour tout écart entre la mesure et la consigne, plus la perturbation est grande, plus la correction apportée est grande. Cette composante seule ne permet pas une grande précision surtout dans les systèmes à faible inertie, comme dans le traitement de l'aire, cette rapidité d'action engendre un phénomène appelé le pompage
I	Cette composante apporte une notion de temps d'intégration à la correction, Cette notion de temps s'exprime généralement en seconde. Cette action est complémentaire à l'action proportionnelle, elle permet de stabiliser dans le temps l'action proportionnelle, plus l'erreur mesurée est constante plus la correction est constante
D	Cette action permet d'anticiper la réponse de la régulation en cas de perturbation rapide ou de modification de consigne ce qui améliore la stabilité du système. On peut donc dire que cette composante permet de compenser tout dépassement excessif de la consigne.

Tab III.1 : Le rôle des paramètres PID [34]

III.3.2. Paramètres d'un régulateur PID

L'idée de base de ce régulateur est de générer une commande u(t) donnée par le régulateur PID, dans sa forme classique est décrite par l'équation **(III.3)**.

$$H_{BF}(p) = \frac{H_{BO}(p)}{1+H_{BO}(p)} \qquad \textbf{(III.3)}$$

Elle est composée de la somme de trois termes **:**

- **Le terme proportionnel** « P » (proportionnel à l'erreur).

Kp: est le gain proportionnel

$$p = K_P\, e(t) \qquad (\textbf{III.4})$$

- **Le terme intégral** « I » (proportionnel à l'intégrale de l'erreur).

Ti : est la constante de temps de l'action intégrale, intégrale, en secondes ou en minutes. Peut être réglé par son inverse (répétitions par seconde ou par minute).

$$I = K_p \frac{1}{T_i} \int_0^t e(t)dt \qquad (\textbf{III.5})$$

- **Le terme dérivatif** « D » (proportionnel à la dérive de l'erreur) .

Td : est la constante de temps de la partie dérivée

$$D = K_p T_d \frac{de(\tau)}{d\tau} \qquad (\textbf{III.6})$$

Les paramètres du régulateur PID sont le gain proportionnel Kp , le temps intégral Ti Et le temps dérivatif Td, les temps étant exprimés en secondes. **[33]**

III.4.1. Méthode de Ziegler-Nichols

Deux méthodes classiques expérimentales de détermination et ajustement rapide des paramètres des régulateurs PID ont été présentées par Ziegler et Nichols en 1942. Ces méthodes sont largement utilisées, soit sous forme originale ou dans une certaine modification. Ils forment souvent la base de procédures de réglage utilisées par les contrôleurs des fabricants et les processus de l'industrie. Les méthodes sont basées sur la détermination de certaines caractéristiques de la dynamique des processus. Les paramètres du régulateur sont alors exprimés en termes de fonctionnalités par des formules simples. Il est surprenant que les méthodes soient si largement référencées parce qu'ils donnent de bons résultats de réglage seulement dans des situations limitées. **[35]**

III.4.2. Méthode de la courbe de réaction (Première méthode)

Cette méthode est basée sur la modélisation de l'information indicielle du processus en boucle ouverte, d'où seulement les processus simple sont utilisés, le principe est d'enregistrer la courbe de réponse du système non régulé à un échelon puis en déduire la valeur des coefficients par

analyse de la réponse (i.e. "lecture graphique"), ainsi mettre le système hors ligne. C'est pour cette raison que cette méthode n'est pas très utilisée dans l'industrie. **[36]**

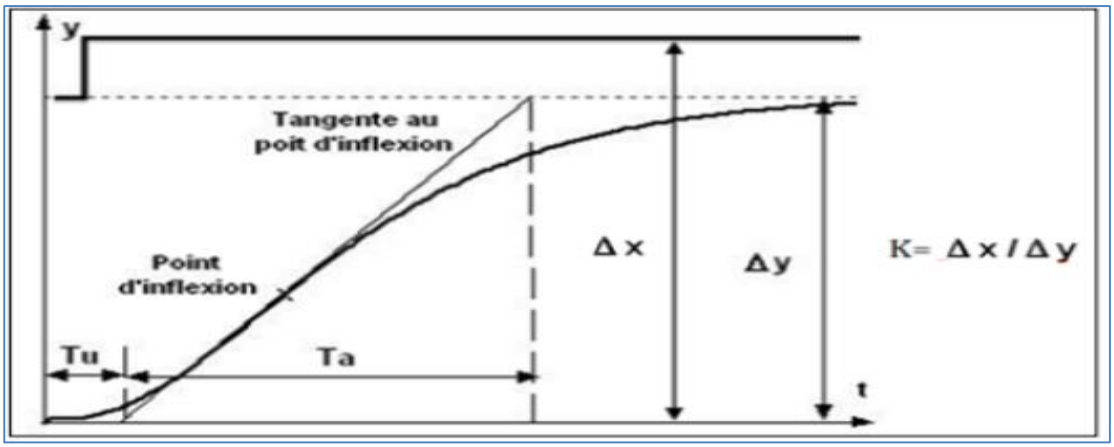


Figure III.6: Réglage de Ziegler-Nichols d'un système stable en boucle ouvert.

Modes de régulation	K_p	T_I	T_D
P	T_a/T_u	∞	-
PI	$0.9 * T_a/T_u$	$T_a/0.3$	-
PID	$1.2 * T_a/T_u$	$2 * T_u$	$0.5 * T_u$

Tab III.2 : Le choix du régulateur en fonction des coefficients de réglage.

III.4.3. Méthode d'oscillation (Seconde méthode)

Cette méthode empirique de Ziegler-Nichols est très répandue dans l'industrie ou chez les automaticiens pour régler les correcteurs de ce type de chaîne. Elle a l'avantage de ne pas nécessiter de modélisation précise du système asservi, mais se contente d'essais expérimentaux, ce qui rend cette méthode très simple, Elle est basée sur la réponse du système en boucle fermée. Elle consiste à effectuer les opérations suivantes :**[36]**

1/ Eliminer l'action dérivée.
2/ Eliminer l'action intégrale.
3/ Partant d'une valeur minimale du gain proportionnel, essayant de l'ajuster afin d'avoir un système oscillatoire.
4/ Calculer le temps d'oscillation Tc (période du signal)
5/ Calculer les paramètres du PID selon le tableau suivant :

Contrôleurs	K_p	T_i	T_d
P	$0.5 * Kpc$	∞	0
PI	$0.45 * Kpc$	$0.85 * Tc$	0
PID	$0.6\ Kpc$	$0.5 * Tc$	$0.12 * Tc$

Tab III.3 : réglage Ziegler Nichols en boucle fermée

III.5. Les blocs de programmations dans Step (FC 105 , 106 , FB 41)

➢ **Le bloc FC 105**

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

- **Si** l'entrée est Bipolaire : La valeur entière d'entrée est supposée être comprise entre 27648.0 et 27648.0, donc : **K1** = -27648.0 et**K2** = +27648.0.
- **Si** l'entrée est unipolaire : La valeur entière d'entrée est supposée être comprise entre 0.0 et 27648.0, donc : **K1** = 0.0 et **K2** = +27648.0.
- S**i** 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.**[37]**

➢ **Paramètres de la FC105 :**

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

➢ **Le bloc FC 106**

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

III.6. Régulateur continue avec le FB 41 « CONT_C »

Le bloc FB 41 « CONT_C » sert à régler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage vous permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système réglé. **[38]**

III.6.1. Programmation de la régulation PID en utilisant le bloc FB 41 intégré

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

- La régulation continue.
- La régulation pas à pas.
- La modulation de largeur d'impulsion**.[29]**

Type de régulation	Bloc de régulation	Type de la sortie de régulation
Continue	FB41 (DB 41)	Analogique
Pas à pas	FB42 (DB 42)	Impulsions
Largeur d'impulsions	FB43 (DB 43)	Analogique

Tab III.4 : la différents types de régulation PID sous Step 7.

Les FB de régulation proposent une régulation purement logicielle, c'est-à-dire qu'un bloc contient toutes les fonctions du régulateur. Les données nécessaires au calcul cyclique sont stockées dans des blocs de données associés OB, les blocs de données d'instance, ce qui permet aux FB de les appeler plusieurs fois. Le bloc FB41 sert à réguler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage du bloc FB 41 nous permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système régulé. Les fonctions les plus importantes sont : **la consigne** et **la mesure**, le bloc FB41 réalise un PID prêt à l'emploi avec une sortie continue et possibilité d'ajuster manuellement la valeur de sortie. **[29]**

III.6.2. Description du bloc FB41 (CONT_C)

En plus des fonctions traitant la consigne et la mesure, le FB réalise un régulateur PID prêt à l'emploi avec sortie continue de la grandeur de réglage et possibilité à la main la valeur de réglage. Selon le type de CPU, il sera mis en œuvre grâce au FB41 (pour les **CPU 313**) ou au FB (pour les CPU sans interface profibus). Il propose les fonctions **partielles suivantes :**

- Branche de consigne.
- Branche de mesure.

Le FB réalise un régulateur PID prêt à l'emploi avec sortie continue de la grandeur de réglage et possibilité d'influencer à la main la valeur de réglage. Il propose les fonctions partielles suivantes**:**

A. Branche de consigne :
La consigne est entrée en format de virgule flottante à l'entrée **SP_INT**.

B. Branche de mesure :
La mesure peut être lue en format de périphérie ou de virgule flottante. La fonction CRP_IN convertit la valeur de périphérie PV_PER en un nombre à virgule flottante compris entre –100 et +100 % selon la formule suivante :

$$\textbf{Sortie de CPR}_{\textbf{IN}} = \textbf{PV}_{\textbf{PER}} * \frac{\mathbf{100}}{\mathbf{27648}} \qquad (\textbf{III. 7})$$

III.7. Modélisation et Commande d'un Système Hydraulique à Réservoirs Multiples par une Approche de Contrôle PID

Notre système hydraulique se compose d'un réservoir d'eau qui est rempli par une **pompe 1** intégrée à une **vanne 1** à contrôle PID. La **pompe 1** et la **vanne 1** sont activées uniquement si **PV(** Le niveau de l'eau dans le réservoir**)** est inférieur à **SP (**consigne" La valeur requise "**)**.

En bas du réservoir, une **vanne 2 (TOR)** est connectée à un chauffe-eau qui s'ouvre si **PV** est supérieur à **5%**. La **pompe 2** fonctionne si la température de l'eau est supérieure à **75 °C** .
La Cuve est remplie par une **pompe 2** jusqu'à **85%** et automatiquement la **vanne 2** est fermée si cette valeur est dépassée.

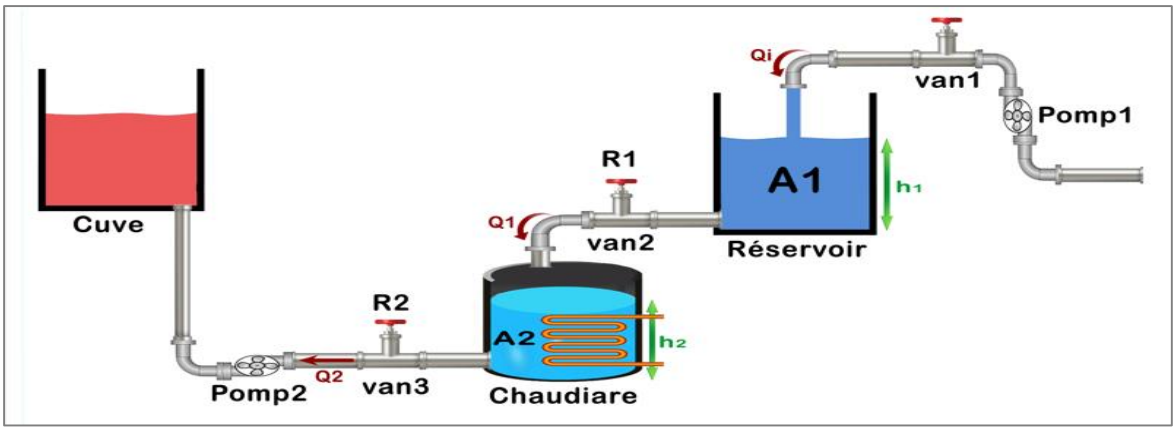


Figure III.7 : Un exemple de système hydraulique.

III.7.1. Modélisation du système hydraulique

Considérons maintenant système hydraulique comprenant d'un réservoiret la chaudière qu'est représenté par la **Figure III.11** Les équations d'état sont obtenues en remarquant que la variation du volume d'eau dans un réservoir est égale à la somme des débits entrants moins la somme des débits sortants. Le système considéré contient " réservoirs / chaudière" et caractérisé par deux états **h1** (le niveau dans le réservoir) et **h2** (le niveau dans la Chaudière) et trois débits : **Qi** c'est le débit d'entrée dans le réservoir, **Q1** est le débit d'entrée dans la chaudière, ou" **Q1** Le flux de sortie du réservoir est le flux d'entrée de la chaudière ",**Q2** est le débit d'entrée dans la Cuve.

III.7.2.Modélisation mathématique à l'écoulement linéaire

Inspirant ce modèle à partir de principe d'évolution de la mécanique de fluide des systèmes hydraulique, via l'utilisant l'équation d'équilibre d'écoulement, pour les réservoirs et la chaudière, on obtient

$$\begin{cases} \frac{dh_1}{dt} = \frac{1}{A_1}\,(Q_i - Q_1) \\ \frac{dh_2}{dt} = \frac{1}{A_2}\,(Q_1 - Q_2) \end{cases} \qquad (\mathbf{III.8})$$

On a Les débits de sortie de réservoir et la chaudière donne par :

$$Q_1 = \frac{H_1}{R_1};\quad Q_2 = \frac{H_2}{R_2} \qquad (\mathbf{III.9})$$

Où :

h1: le niveau du liquide dans le réservoir.

A1 : la section du réservoir $\mathbf{m^2}$.

$\boldsymbol{R1}$: Est la résistance du débit de sortie du réservoir($\boldsymbol{m^3/s}$)

Alors, le système hydraulique est régi par l'équation différentielle linéaire suivante **:**

$$\begin{cases} \frac{dh_1}{dt} = \frac{1}{A_1}\left(Q_i - \frac{H_1}{R_1}\right) \\ \frac{dh_2}{dt} = \frac{1}{A_2}\left(\frac{H_1}{R_1} - \frac{H_2}{R_2}\right) \end{cases} \quad (\mathbf{III.10})$$

a. La fonction de transfert de réservoir :

$$\text{On a : } \frac{dh_1}{dt} = \frac{Q_i}{A_1} - \frac{H_1}{A_1 R_1} \quad (\mathbf{III.11})$$

Avec la transformation de Laplace :

$$sH_1(s) = \frac{1}{A_1}Q_i(s) - \frac{1}{A_1 R_1}H_1(s) \quad (\mathbf{III.12})$$

Alors La fonction de transfert est : $\frac{H_1(s)}{Q_i(s)} = \frac{R_1}{A_1 R_1 s + 1}$ $(\mathbf{III.13})$

b. La fonction de transfert de Chaudière:

On a :

$$\frac{dh_2}{dt} = \frac{H_1}{A_2 R_1} - \frac{H_2}{A_2 R_2} \quad (\mathbf{III.14})$$

Avec la transformation de Laplace :

$$sH_2(s) = \frac{1}{A_2 R_1}H_1(s) - \frac{1}{A_2 R_2}H_2(s) \quad (\mathbf{III.15})$$

Alors La fonction de transfert est :

$$\frac{H_2(s)}{H_1(s)} = \frac{R_2}{R_1\left(A_2 R_2 s + 1\right)} \quad (\mathbf{III.16})$$

Où :

h2: Le niveau de l'eau dans La chaudière
A2: la section de la Chaudière m^2.
$\boldsymbol{R2}$: Est la résistance du débit de sortie de la Chaudière $(\boldsymbol{m^3/s})$

c. La fonction de transfert de system global :

La fonction de transfert de l'hydraulique est calculée en réarrangeant chaque équation.

$$\frac{H_2(s)}{Q_i(s)} = \frac{H_1(s)}{Q_i(s)}\frac{H_2(s)}{H_1(s)} = \frac{R_2}{(\tau_1 s+1)(\tau_2 s+1)} \quad (\mathbf{III.17})$$

Avec $\tau_1 = A_1 R_1$; $\tau_2 = A_2 R_2$

III.7.3.Application de la commande système hydraulique par un PID

Dans cette application, nous fournirons le contrôle **PID** sur le système hydraulique en utilisant du Méthode Ziegler-Nichols. Ce system hydraulique est représenté par la fonction de transfert suivante :

$$\frac{H_2(s)}{Q_i(s)} = \frac{R_2}{(\tau_1 s+1)(\tau_2 s+1)} \qquad (\textbf{III. 18})$$

En suppose :

$$\mathbf{R_1 = 2\ (\frac{m^3}{s})} \quad ; \quad \mathbf{R_2 = 1\ (\frac{m^3}{s})} \ ; \ \boldsymbol{A_1 = 2m^2} \quad ; \quad \boldsymbol{A_2 = 1m^2}$$

Alors la fonction de transfert globale du système de hydraulique est représentée par :

$$\frac{\boldsymbol{H_2(s)}}{\boldsymbol{Q_i(s)}} = \frac{\mathbf{1}}{\boldsymbol{(4s+1)(s+1)}} \qquad (\textbf{III. 19})$$

III.7.4.Calcule des paramètres de régulateur PID

* obtenir expérimentalement la réponse de le system à une entrée de pas unitaire.

* La réponse peut avoir l'apparence d'une courbe si le system ne contient ni intégrateur ni pôles complexes conjugués dominants.

* La courbe peut être caractérisée par deux constantes : le temps de retard L et la constante de temps T.

*La fonction de transfert T(s) peut être approximée par un système du deuxième ordre avec un retard de transport comme suit:

$$\boldsymbol{T(s) = \frac{1}{4s^2+5s+1}} \qquad (\textbf{III. 20})$$

```
% System to be controlled

numerator = 1;
denominator = [4, 5, 1];
G = tf(numerator, denominator);
step(G);
title('Step Response of the open-Loop System');
xlabel('Time');
ylabel('Amplitude');
```

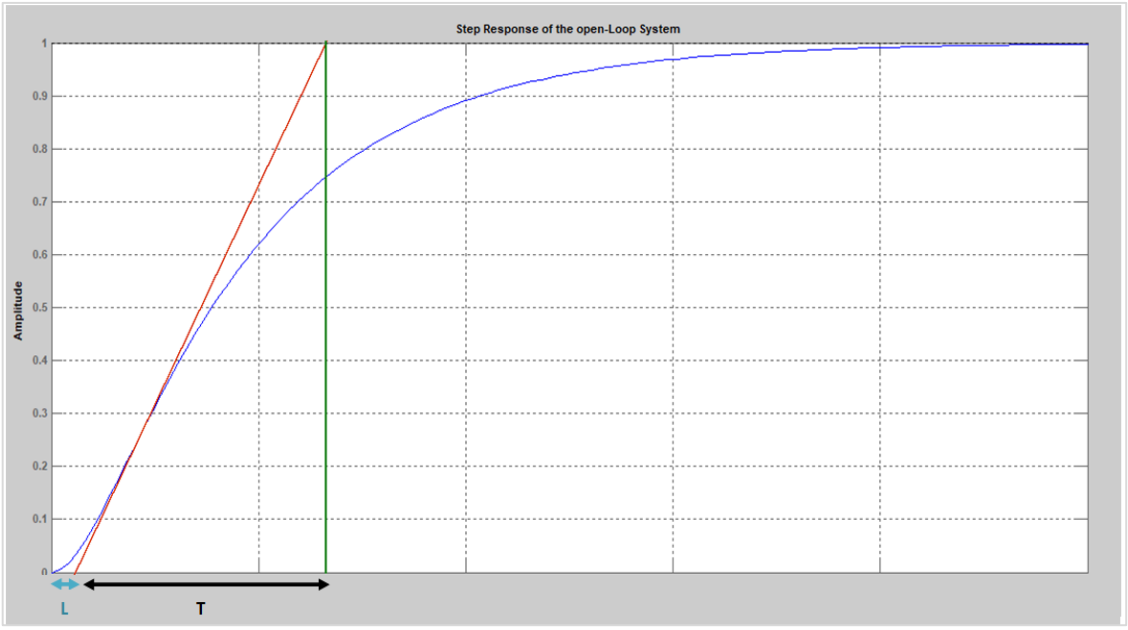


Figure III.8: Step Response of the open-Loop System.

* Nous avons besoin du temps de retard (**L**) et de la constante de temps (**T**) pour la première méthode.

* Dessinez une ligne tangente au point d'inflexion.

* La ligne tangente est utilisée pour déterminer le temps de retard et la constante de temps.

* Le graphe montre : le temps de retard est d'environ **1** seconde et la constante de temps est de **1.2** seconde. Ainsi : **L = 1** seconde et **T = 1.2** seconde.

Déterminez les paramètres du PID, **kp**, **Ti** et **Td**, selon les formules indiquées dans le tableau de Ziegler et Nichols.À partir du tableau de Ziegler et Nichols, nous avons pour le régulateur PID.

$$\boldsymbol{K_p} = 1.2\,\frac{T}{L} = 1.2\,\frac{1.2}{1} = \mathbf{1.2} \qquad (\mathbf{III.21})$$

$$\boldsymbol{T_i} = 2L = 2 * 1 = \mathbf{2}\ \boldsymbol{sec} \qquad (\mathbf{III.22})$$

$$\boldsymbol{T_d} = 0.5\,L = 0.5 * 1 = \mathbf{0.5}\ \boldsymbol{sec} \qquad (\mathbf{III.23})$$

```
% Define transfer function
num = [1];
den = [4 5 1];
sys = tf(num, den);

% Design PID controller
Kp = 1.2;
Ti = 2;
Td = 0.5;
C = pid(Kp, Kp/Ti, Kp*Td);

% Connect PID controller to plant
sys_cl = feedback(C*sys, 1);

% Plot step response
step(sys_cl)
```

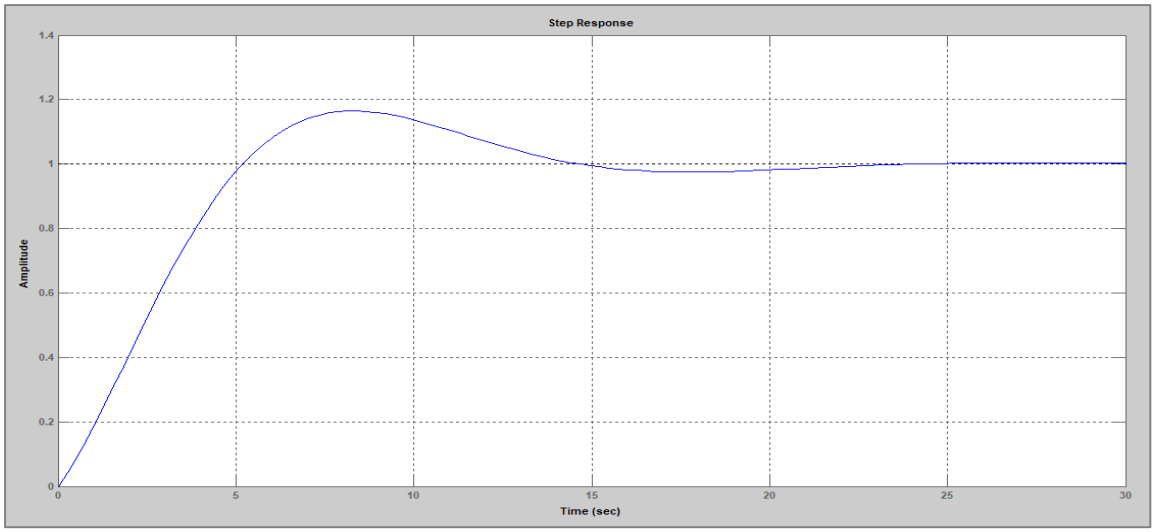


Figure III.9 :Réponse à l'échelon pour le système avec régulateur PID.

III.8. Programmation et Supervision

III.8.1. La mise en œuvre d'un régulateur PID sous step7

La programmation d'un SIMATICS7-300 en tant que régulateur PID se fait avec le logiciel STEP7, les entrées sorties choisis de régulateur PID (bloc FB41) seront sauvegardés dans le bloc de donné (DB) d'instance associé à l'appel du FB41. Pour réaliser un régulateur PID nous devrons suivre les sept étapes suivantes :

- ❖ Création nouveau projet avec le chemin.
- ❖ Elaboration d'un programme en langage Contact sous STEP7.
- ❖ Insérer un objet de type «station SIMATIC 300»
- ❖ Double clique sur l'icône matérielle et insérer les éléments (PS : PS 307 10A, CPU:313, AI/AO 4/2, DI/DO 8/8.
- ❖ Insertion du bloc d'organisation OB35

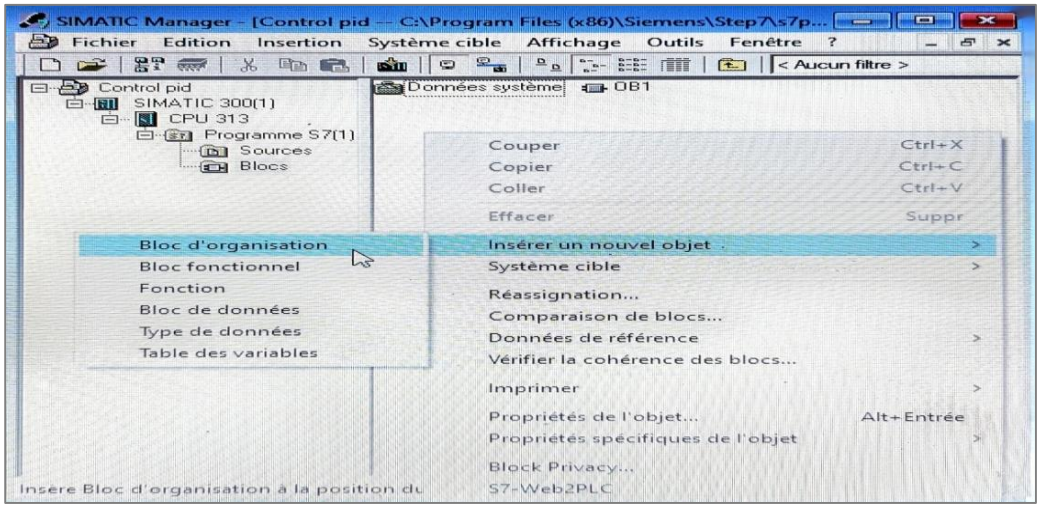


Figure III.10 :Insertion du bloc d'organisationOB35.

❖ La création du programme sous l'OB35

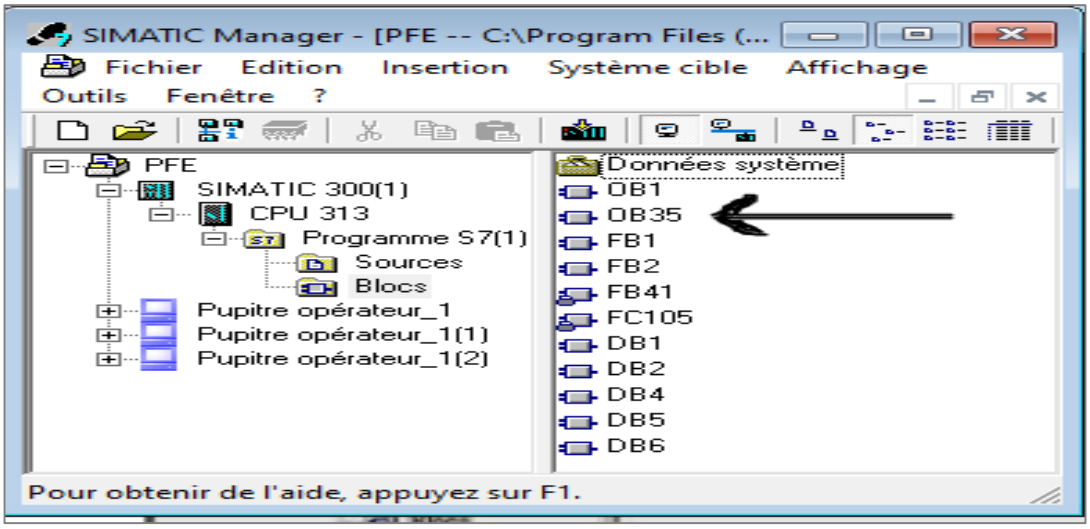


Figure III.11: Le bloc pour mise en place de programme.

III.8.2. La programmation de régulation PID sous step7

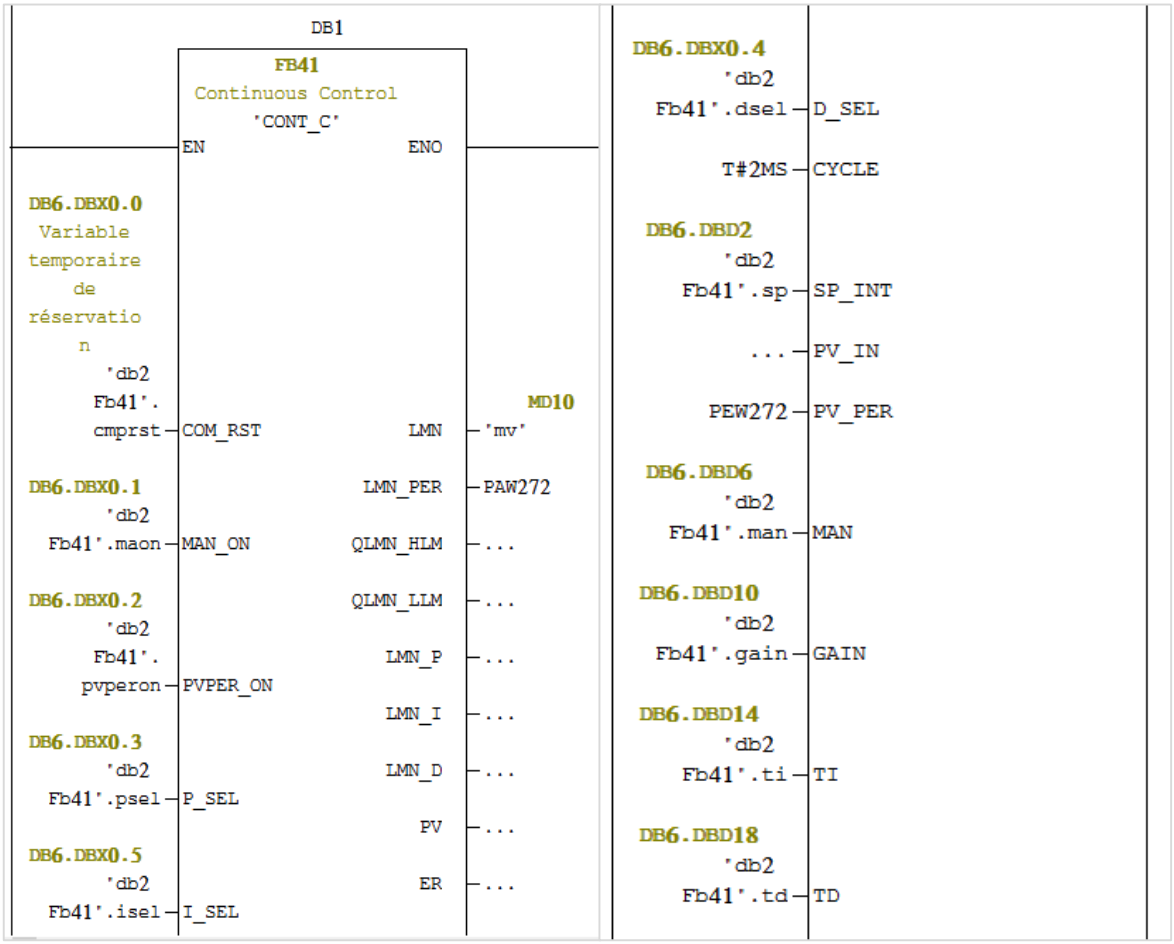


Figure III.12 : paramétrage du régulateur sous Step 7 (FB 41).

On va utiliser un bloc de données (**DB6**) pour les paramètres de cette régulation :

PFE\SIMATIC 300(1)\CPU 313\...\DB6]

e cible Test Affichage Outils Fenêtre ?

Adresse	Nom	Type	Valeur initia
0.0		STRUCT	
+0.0	cmprst	BOOL	FALSE
+0.1	maon	BOOL	FALSE
+0.2	pvperon	BOOL	FALSE
+0.3	psel	BOOL	FALSE
+0.4	dsel	BOOL	FALSE
+0.5	isel	BOOL	FALSE
+2.0	sp	REAL	0.000000e+000
+6.0	man	REAL	0.000000e+000
+10.0	gain	REAL	0.000000e+000
+14.0	ti	TIME	T#0MS
+18.0	td	TIME	T#0MS

Figure III.13 : Bloc de donnés de (FB41).

III.8.3. Création du programme

1/ Le fonctionnement de Démarrage de la Chaudière.

* Ouvrir la vanne 2 d'eau pour remplir la chaudière

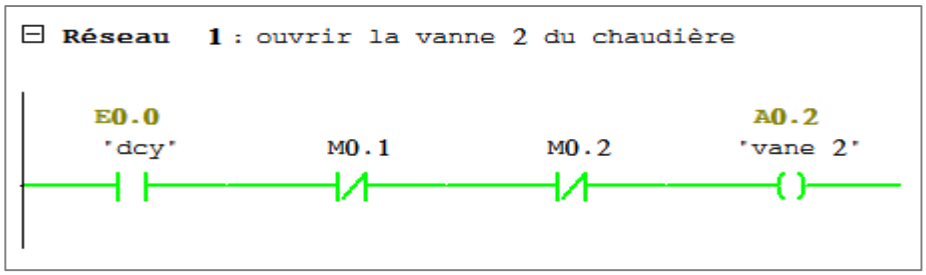


Figure III.14 : Programmation sous step7(la chaudière)

M0.1 : Il est prévu de fermer la vanne 2 lorsque le niveau de la Cuve atteint **85 %**.
M0.2 : La vanne 2 ne s'ouvre que si le niveau d'eau dans le réservoir est supérieur à **5%.**

* Ouverture de la vanne 2 si la (PV) est supérieure à **5%**.

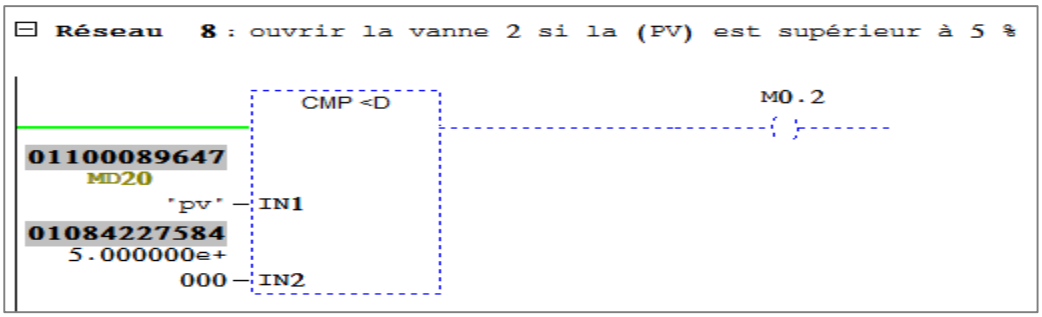


Figure III.15 : Programmation sous step7(vanne 2 ouverte)

***** Allumer le brûleur " Le brûleur s'allume automatiquement si le niveau d'eau dans le réservoir est supérieur à **5%** ".

M0.0: Arrêter le brûleur si la température de l'eau dépasse **90 °C**.

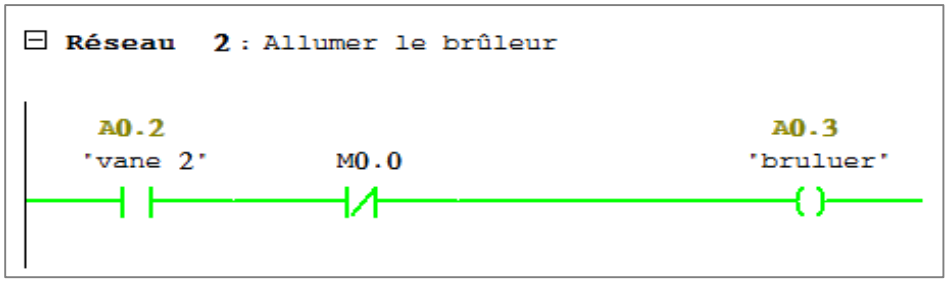


Figure III.16 : Programmation sous step7(bruleur)

***** Le fonctionnement de la régulation de température de la chaudière.

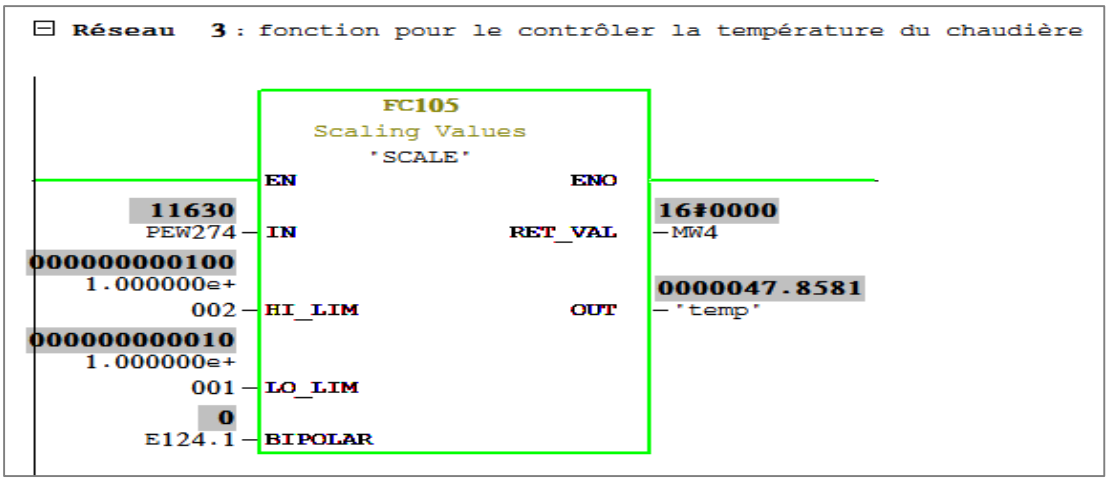


Figure III.17 : Programmation sous step7 (TEMP de la chaudière)

***** Faire fonctionner la pompe 2 pour remplir le Cuve " Elle est activée si la température de l'eau de la chaudière est supérieure ou égale à 75 °C".

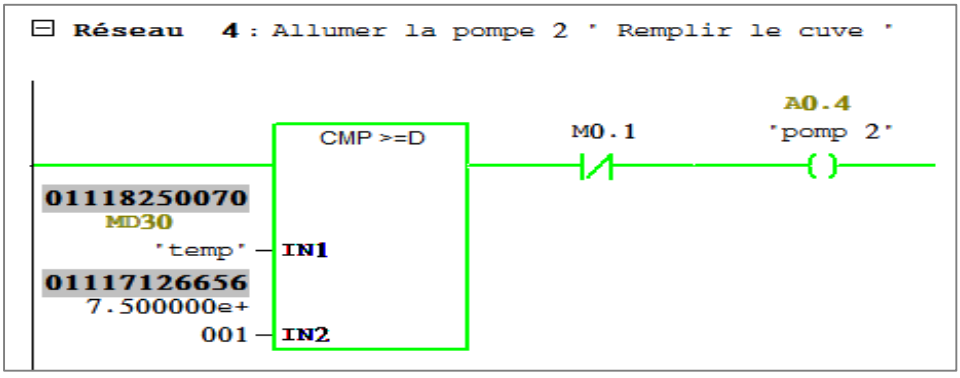


Figure III.18 : Programmation sous step7(Allumer la pompe)

***** Le fonctionnement de contrôle du niveau d'eau dans la Cuve.

***** Le brûleur s'arrête automatiquement si la température de la chaudière dépasse 75 °C.

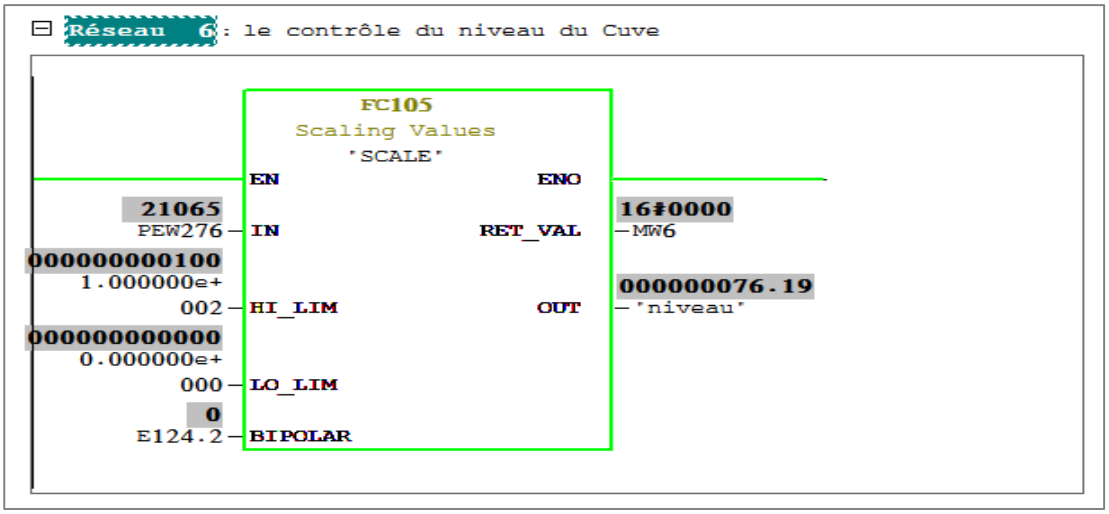


Figure III.19 : Programmation sous step7(niveau du réservoir)

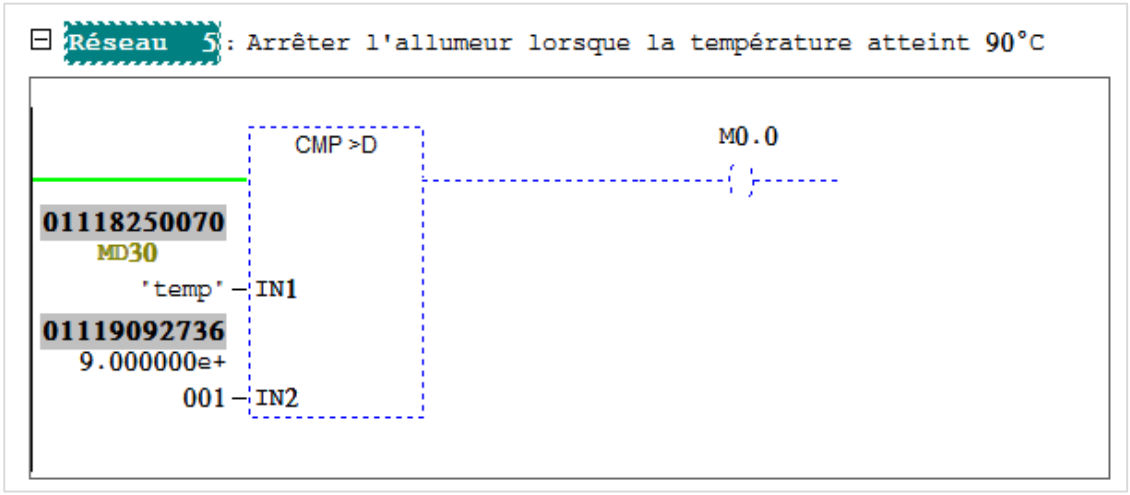


Figure III.20 : Programmation sous step7 (éteindre le brûleur)

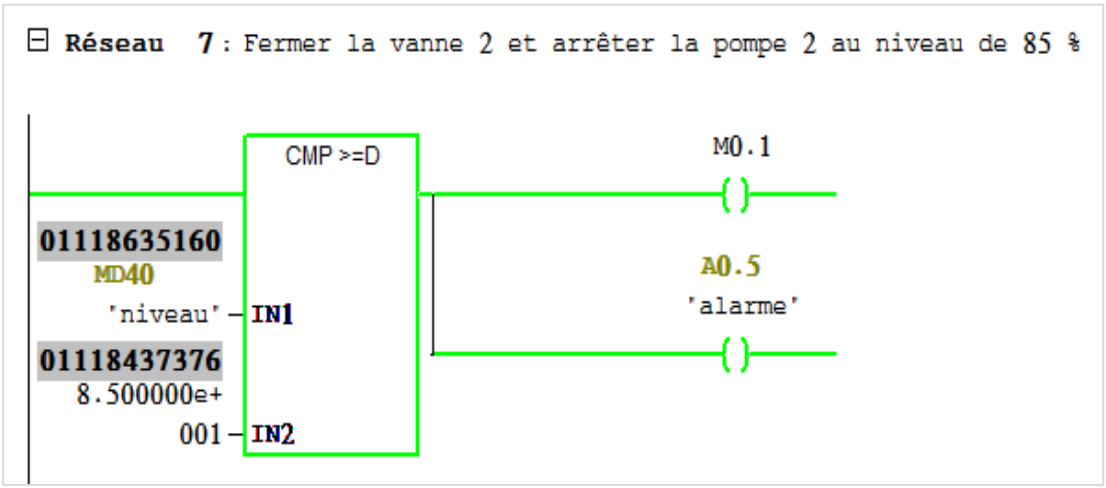


Figure III.21 : Programmation sous step7 (Arrêter la pompe 2)

III.8.4. Programmation de la régulation PID en utilisant le bloc FB 41 intégré [OB35]

Pour des mesure de sécurité de système et après le démarrage des pompes nous avons programmé cette fonction FB41 du régulateur PID qui va surveiller le niveau du bac.

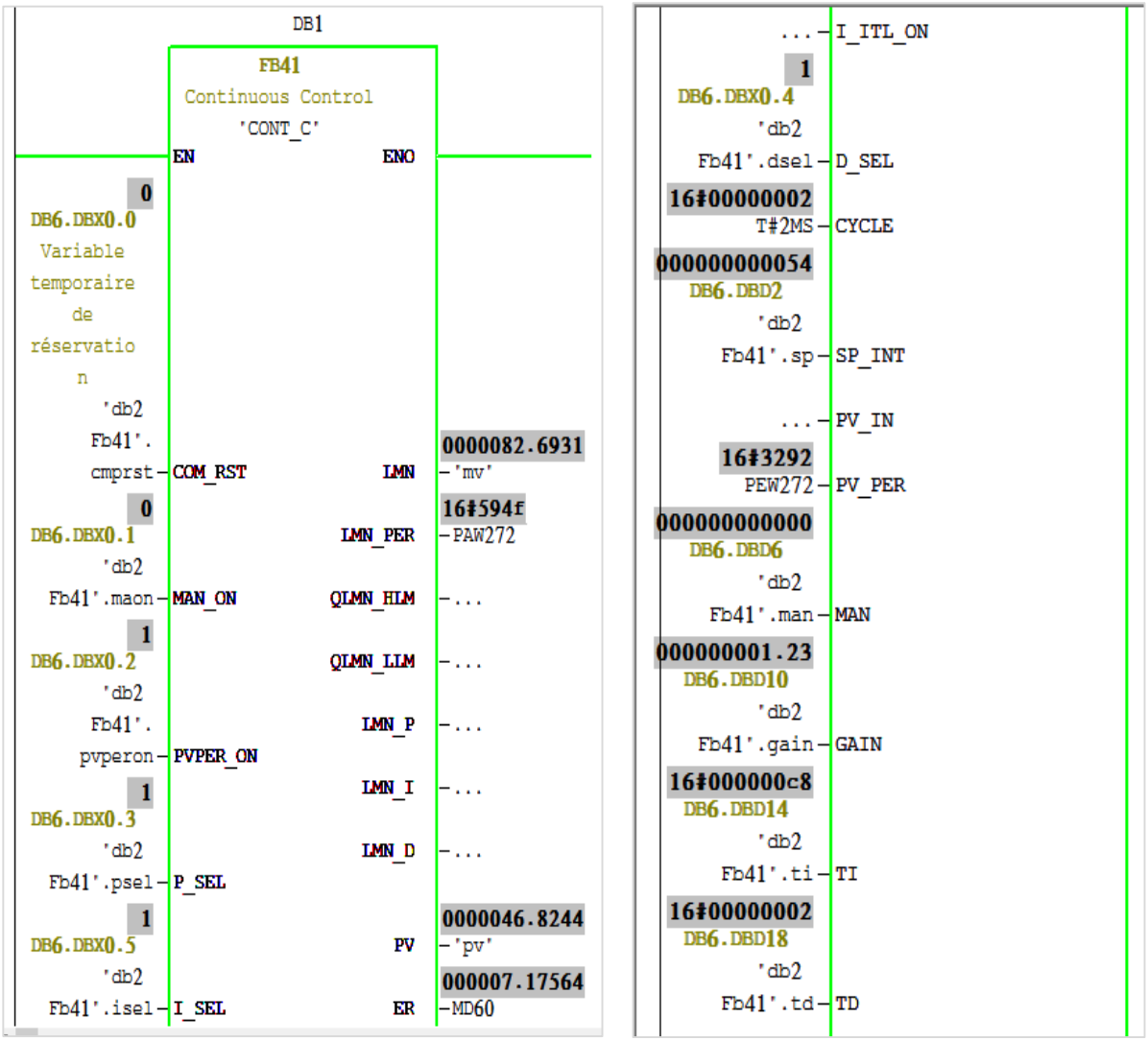


Figure III.22 :Paramétrage du régulateur sous step7 (FB41)

* La vanne située en dessous du réservoir s'ouvre automatiquement si le niveau d'eau dépasse SP

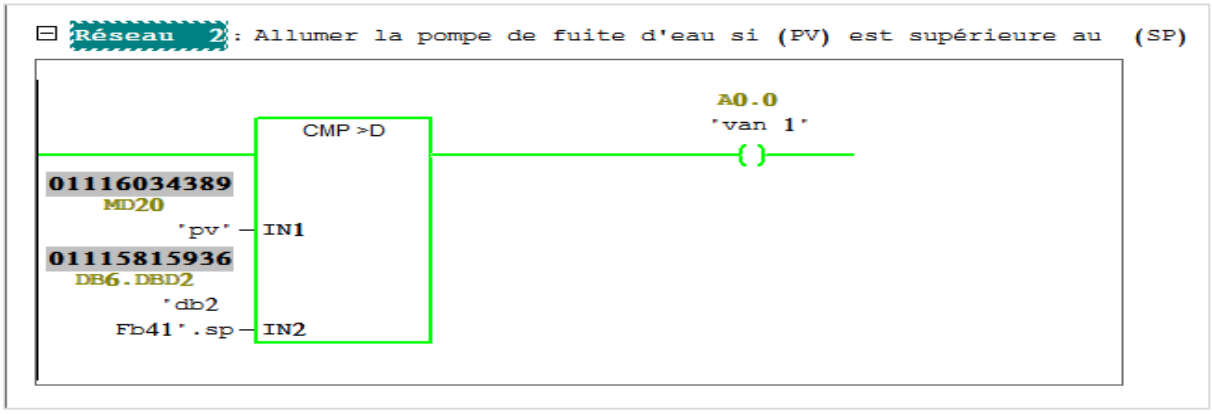


Figure III.23 : Programmation sous step7 (Arrêter la pompe 2).

III.9. Logiciel WINCC

III.9.1. Présentation WINCC flexible

SIMATIC WinCC " Windows Control Center " est un logiciel de **SIEMENS** qui permet de concevoir des IHM dans le domaine des automates. Il permet de créer des interfaces graphiques des vues, de configurer les protocoles de communication, d'établir des laissons et de créer des variables (interne ou externe) entre l'interface **IHM**, Step7et le procédé pour pouvoir lire les valeurs réelles du processus via le câble MPI, les afficher pour que l'operateur peut les interprétées et ajuster, éventuellement, le processus IHM, toujours via l'automate, pour visualiser le fonctionnement dynamique en temps réel du processus par des objets symbolisant les différents éléments des procédés, et de contrôler aussi le processus ou l'opérateur peut intervenir, modifier ou faire entrer des nouvelle valeurs, à tout moment. **[20]**

a. Création d'un projet

Pour créer un nouveau projet sur Wincc flexible, il faut de faire :

1/ Premièrement, on clique deux fois sur l'icône SIMATIC WinCC flexible 2008.

Figure III.24 : Simatic WinCC.

2/ Deuxièmement, après l'ouverture, on constate cette fenêtre **(** comme le montre la **Figure III.25)**.

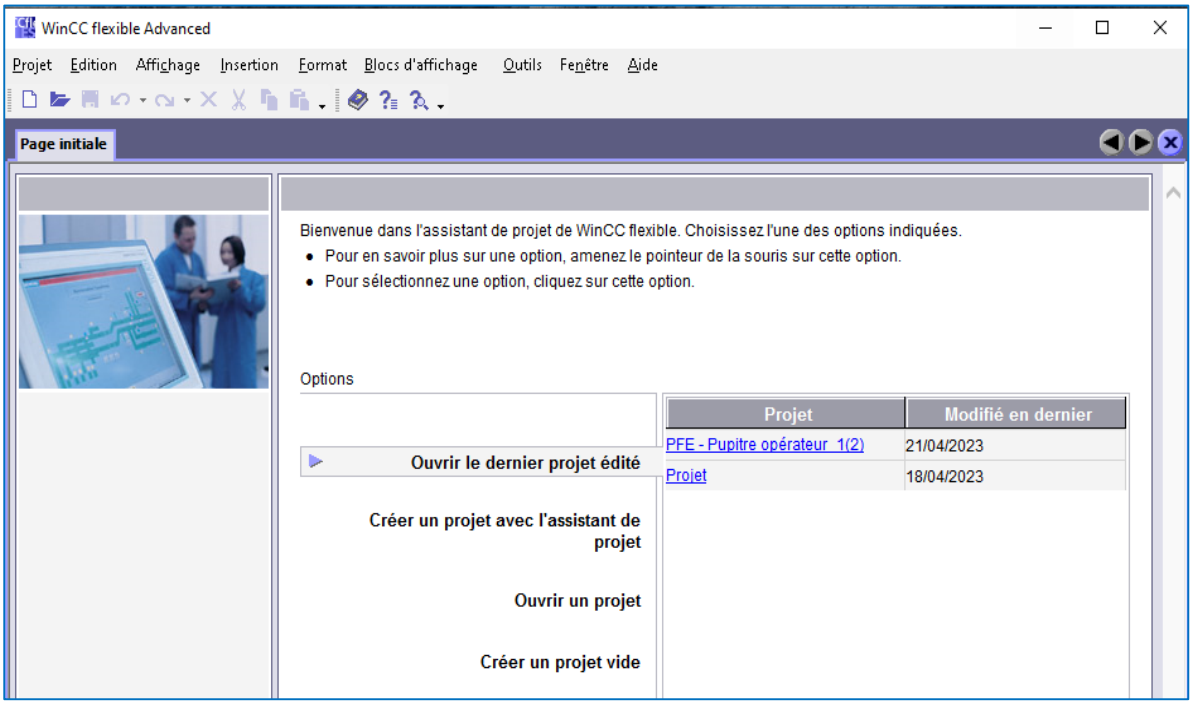


Figure III.25 : Plateforme de logiciel WinCC flexible.

On a créé facilement un nouveau projet, on choisit l'HMI appropriée pour le travail. L'interface comprend les parties montrées dans cette **Figure III.26**.

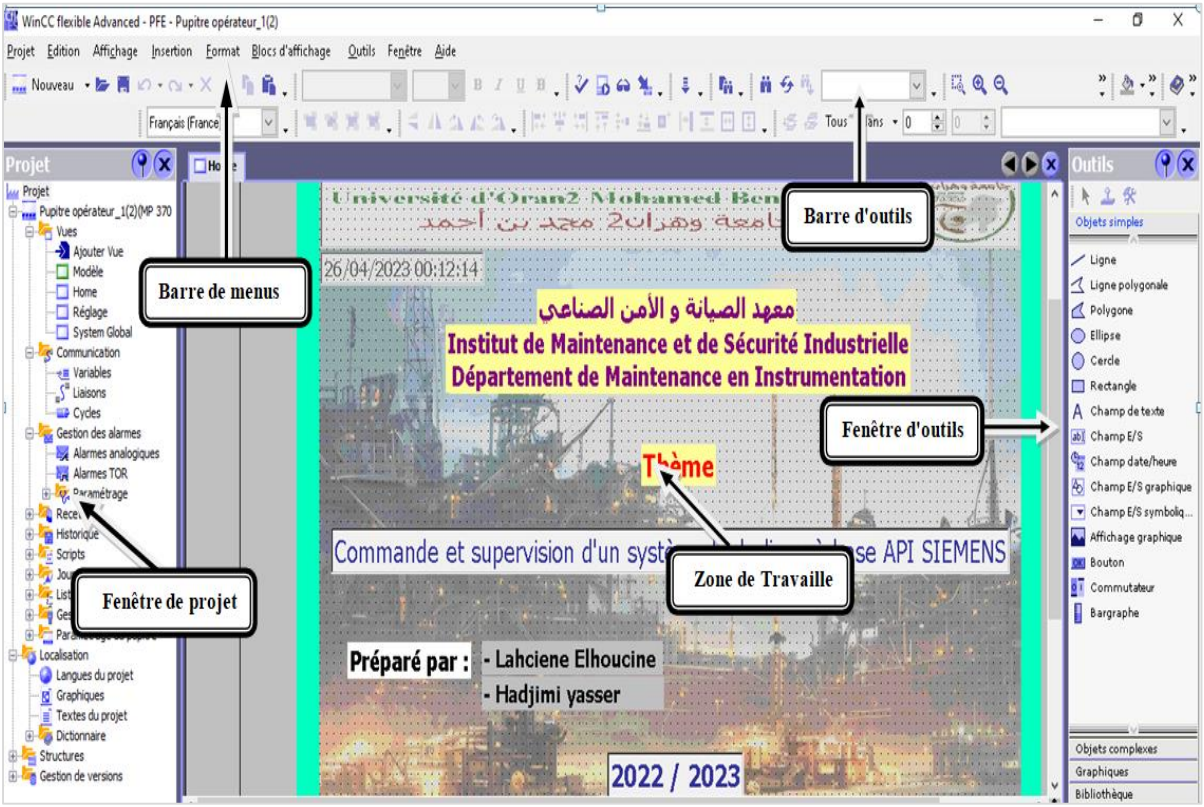


Figure III.26 : Les parties d'interface winCC.

III.9.2. Intégration d'un programme crée en WinCC dans le projet STEP7

L'intégration du projet Wincc à l'étape 7 est une bonne solution, en passant au traitement du projet via l'interface homme-machine qui a été préparée par le logiciel Wincc pour le Régulation de débit et niveau d'un système hydraulique. Pour cela, on clique sur « **projet**, **intégrer dans le projet Step 7** » puis on choisit le nom de projet " **PFE** " dans la barre d'outils de Wincc flexible.

a. Création des variables WinCC

Les variables externes permettent de communiquer, c.-à-d. d'échanger des données entre les composants d'un processus automatisé, p. ex entre un pupitre opérateur et un automate.

b. Tableau des variables

En ajoutant les variables qui nous voudrons par double clic sur la ligne de variables et modifier les paramètres de la variable (le nom, liaison, type de données, adresse et n'oublie pas choisir le cycle d'acquisition pour chaque variable ex: **100ms)**

Home | Variables

	Nom	Connexion	Type d...	Mnémoniq...	Adresse	Elé...	Cycle d'ac...
	db2 Fb41.psel	CPU 313	Bool	psel	DB 6 DBX ...	1	100 ms
	v33	CPU 313	Bool	v33	Q 2.3	1	100 ms
	v21	CPU 313	Bool	v21	Q 1.7	1	100 ms
	vane 2	CPU 313	Bool	vane 2	Q 0.2	1	100 ms
	van 1	CPU 313	Bool	van 1	Q 0.0	1	100 ms
	v19	CPU 313	Bool	v19	Q 1.6	1	100 ms
	db2 Fb41.maon	CPU 313	Bool	maon	DB 6 DBX ...	1	100 ms
	v15	CPU 313	Bool	v15	Q 1.2	1	100 ms
	pomp 2	CPU 313	Bool	pomp 2	Q 0.4	1	100 ms

Figure III.27:Les paramètres des variables de WinCC.

III.10. Simulation du programme sous WinCC

Après programmation sous Step 7, on a intégré le programme dans le WinCC flexible pour réaliser le schéma de notre système.

Les vues de supervision de projet

Il permet la visualisation et la surveillance du système. L'interface graphique du système de hydraulique " chaudière et réservoir " se compose de plusieurs vues **:**

1) La vue de la page d'accueil:

Figure III.28:La page d'accueil du projet.

2) La vue de Système global

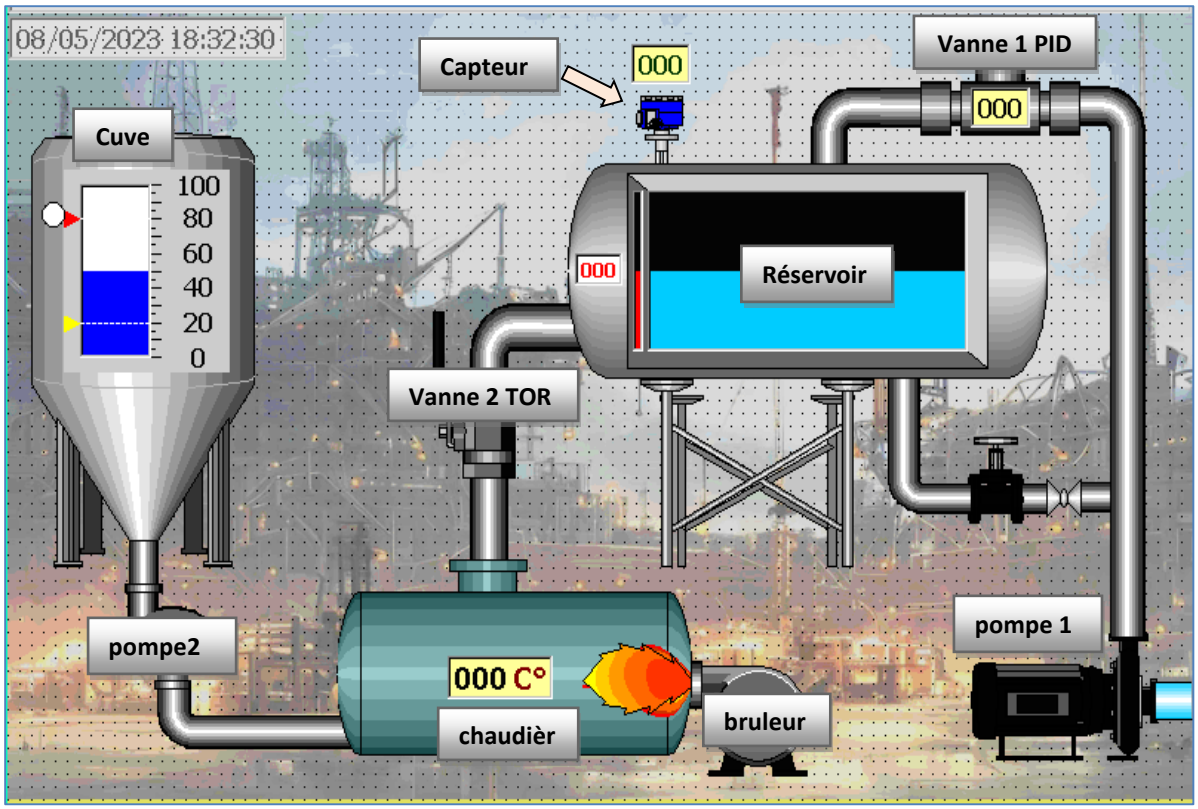


Figure III.29: Système global.

III.10.1. Simulation du programme sous WinCC

III.10.2. WinCC flexible Runtime

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

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

III.10.3. Simulation du programme sous STEP7 avec PLCSIM

Pour simuler ce système hydraulique, plusieurs étapes doivent être effectués. Dans notre PLC on clique sur le bouton droit puis on choisit compiler. Après création du programme, on le charge dans l'automate, et comme il ne s'agit que d'une simulation(en l'absence de l'automate), on utilisera le logiciel S7-PLCSIM. Après le chargement des programmes dans le simulateur, on met la CPU en mode RUN-P (exécution du programme) voir la **Figure III.30**.

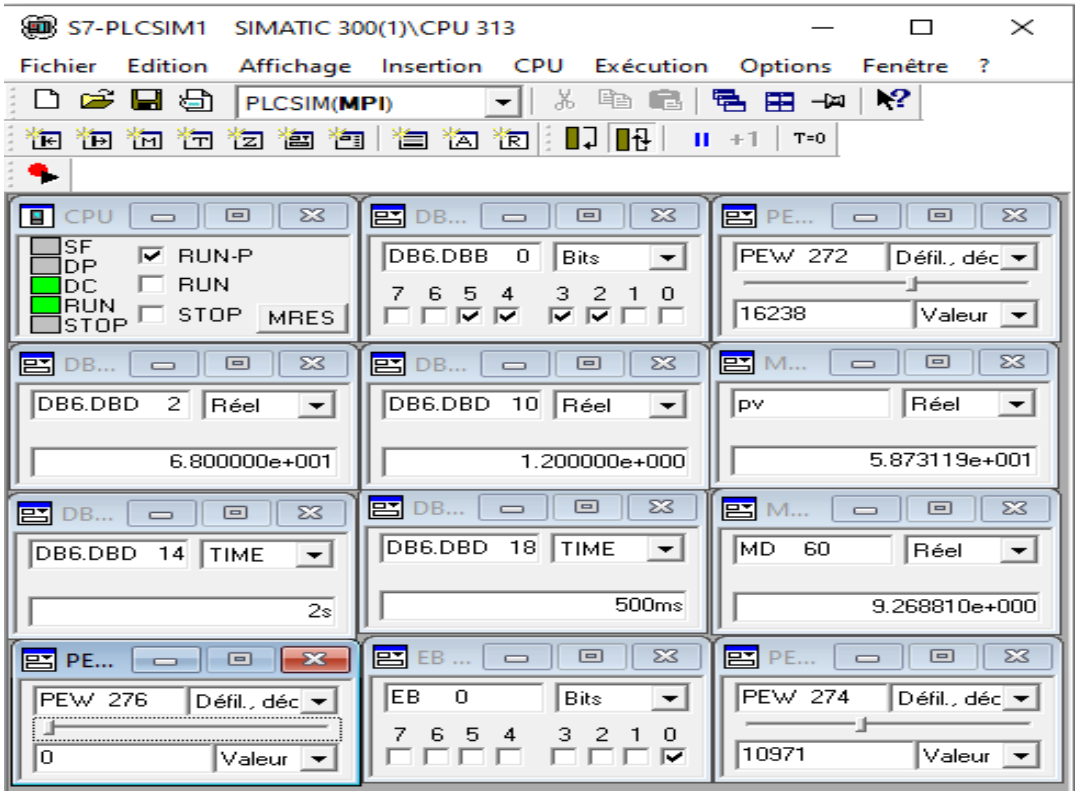


Figure III.30: Simulation avec S7-PLCSIM

On va montrer cette expérience avec PLCSIM pour voir le signale de commande :

PEW 272 : c'est la valeur du niveau d'eau du réservoir (capteur) .
PEW 274 : la valeur de la température.
PEW 276 : la valeur du niveau d'eau du Cuve .
DB6.DBD 0 : Activer ou désactiver les paramètres **P**, **I**, **D** et passer le contrôle de la Régulation PID en mode automatique et prendre l'entrée " **PVPER_ON** ".
DB6.DBD 2 : Set Point (consign)
DB6.DBD10 : Gain _ Kp = **1.2 .**
DB6.DBD14 : Ti = **2s .**
DB6.DBD 18 : Td = **500ms .**
PV : niveau
MD : erreur
EB 0 : Contrôler l'ouverture et la fermeture de la vanne 2 pour remplir le chaudière.

3. Vue de régulation PID

On propose une consigne de **68 %** (**SETPOINT**), Le capteur mesurera le niveau d'eau dans le réservoir **(PV)**, le comparera à la consigne et enverra un signal à l'actionneur **(Vanne 1)**.

✓ Dans ce système, nous prenons les valeurs obtenues dans les équations précédentes.

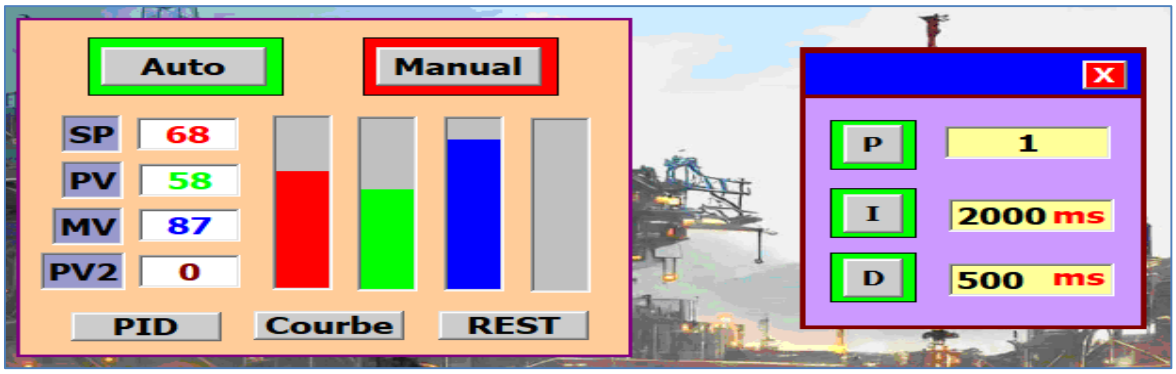


Figure III.31: Activation des valeurs PID ainsi que l'entrée de SP.

4.Vue générale du système hydraulique simulateur WinCC Runtime:

Nous avons constaté que durant l'utilisation de la régulation DISCONTINUE la vanne ne peut pas prendre des valeur pour que le système rend stable par ce que le niveau ne va jamais stabilisé et la vanne va prend une valeur de tout ou rien et ça rend notre procès non confortable sur tout lorsque le système atteint la consigne après chute avec une petite valeur la vanne va s'ouvrir tout et le niveau dépasse la consigne. Au contraire avec la régulation PID nous avons observé que lorsque le système approche a la consigne (SETPOINT) la vanne va recevoir un signal de commande pour la fermeture et ne dépassée pas le seuil ou bien la limite et il va commander le système et le rend plus stable que la régulation discontinue.

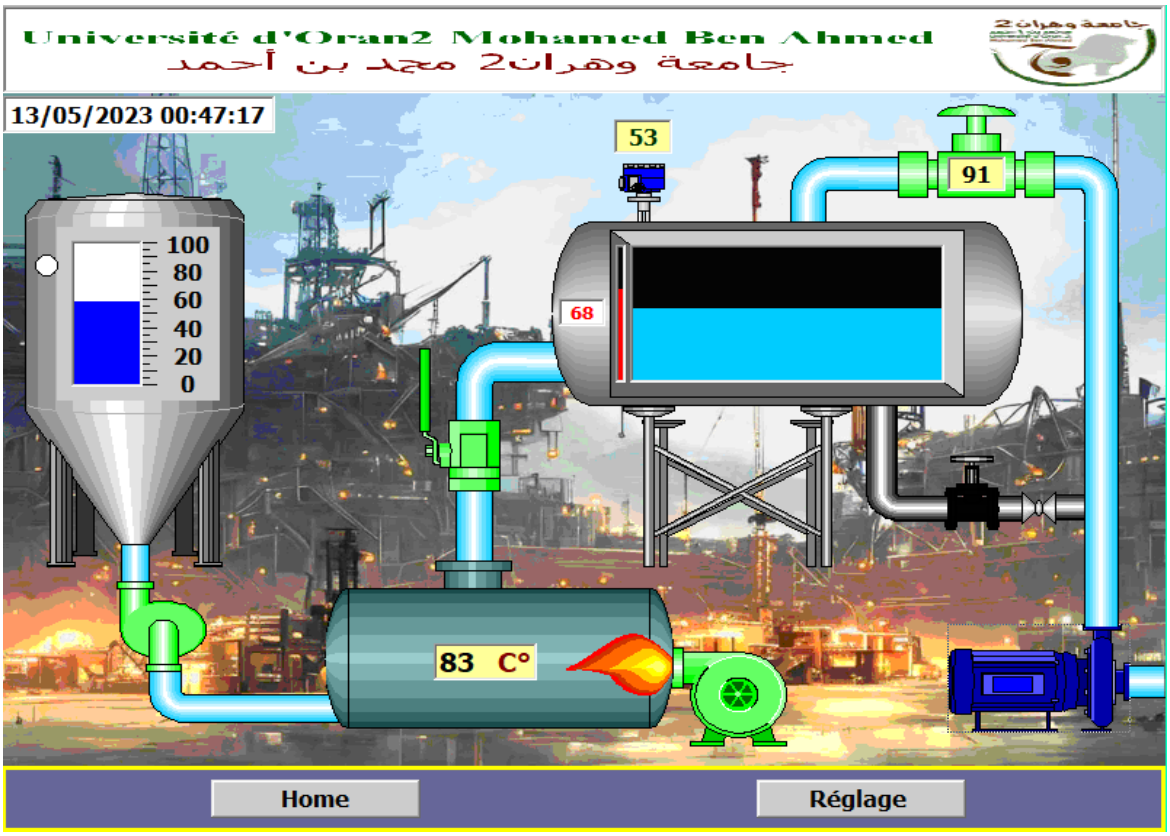


Figure III.32:Runtime du système hydraulique.

III.11. Conclusion

Dans ce chapitre, nous avons étudié la régulation du débit et du niveau dans un système hydrauliques, ainsi que les bases de la régulation PID et ses applications dans le processus industriel. De plus, une interprétation des concepts de PID , a été présentée à l'aide d'outils fournis dans la bibliothèque du logiciel de programmation des APIs (S7-300).

Pour faciliter la tâche de contrôle de l'opérateur, une plateforme de supervision a été développée sous le logiciel WinCC flexible, permettant de suivre l'évolution du processus au fil du temps. Cette plateforme est composée de vues permettant une visualisation dynamique des entrées/sorties pour contrôler le bon fonctionnement du système.

Enfin, les paramètres du régulateur PID ont été expliqués pour obtenir une réponse adéquate en termes de précision, rapidité, stabilité et robustesse en sortie du procédé.

Conclusion Générale

Conclusion Générale

L'objectif essentiel de ce mémoire était de proposer une solution d'automatisation et de supervision d'un système hydraulique. Notre étude visait à automatiser et superviser un système hydraulique simple en utilisant une régulation PID pour contrôler le niveau et le débit. Pour mettre en œuvre notre solution, nous avons utilisé un automate programmable SIEMENS S7-300. nous avons commencé notre étude par une investigation bibliographique, en effectuant une présentation générale des différents types de systèmes dynamiques, ainsi que les notions, fonctions et méthodes de modélisation pour chaque système (systèmes continus, systèmes à événements discrets (SED) et systèmes hybrides). Ce projet particulièrement, aborde la commande, la modélisation, visualisation et supervision d'un système hydraulique simple constituant d'un réservoir, chaudière et une cuve associé avec des vannes multiples (sont TOR et de contrôle analogique). Notre étude s'est portée sur la modélisation, la régulation et la réalisation d'un programme d'automatisation de système globale avec des entrées TOR et analogique et vice versa pour les sorties.

La régulation PID est une technique de régulation qui utilise un algorithme de calcul en boucle fermée pour maintenir une variable de processus (dans notre cas, le niveau et le débit) à une valeur prédéfinie. Cette technique est basée sur la mesure de la différence entre la valeur de consigne (la valeur souhaitée) et la valeur réelle de la variable de processus. Cette différence est appelée erreur, et l'algorithme de régulation calcule une commande proportionnelle, intégrale et dérivée (d'où le nom PID) pour corriger cette erreur et maintenir la valeur de la variable de processus à la valeur souhaitée. En comparaison, la régulation TOR (tout-ou-rien) est une technique de régulation qui utilise des commandes binaires (ouvert/fermé) pour contrôler une vanne ou un autre dispositif de régulation. Cette technique est moins précise que la régulation PID, car elle ne permet pas d'ajuster la commande en fonction de la mesure de la variable de processus. Elle est généralement utilisée pour des applications simples où une précision élevée n'est pas nécessaire.

L'étude que nous avons menée, nous avons choisi d'utiliser la régulation PID pour obtenir une précision élevée dans la régulation du niveau et du débit. Nous avons élaboré un programme de régulation PID en utilisant le logiciel Step7 en langage CONT-C, et nous avons validé ce programme par des simulations sur un automate virtuel. Les résultats ont montré que notre programme était fonctionnel et pouvait être appliqué sur le système réel. Notre solution a permis d'améliorer le fonctionnement du système hydraulique en termes de précision, de rapidité et de fiabilité.

Enfin, Notre étude a démontré l'efficacité de la régulation PID dans la régulation du niveau et du débit d'un système hydraulique. Nous avons acquis une expérience précieuse dans la programmation des automates S7-300, la simulation avec PLCSIM et le logiciel de supervision WinCC, ainsi qu'en modélisation de systèmes et en synthèse de régulateurs. Nous espérons que notre travail pourra servir d'exemple aux futurs étudiants cherchant à mettre en œuvre des projets similaires ou plus sophistiqués.

Références bibliographiques

[01] A.Meghebbar , "Cours-Asservissements-Continus " [**Consulté le : 14-04-2023**], https://n9.cl/athus

[02] A. JUTARD et M.BETEMPS , 1997 - cours de " Mécatronique et Automatique ",3ème année du département de Génie Mécanique , INSA de Lyon.

[03] Ioan-Cristian Trelea , " contribution à la commande prédictive optimale sous contraintes des procèdes discontinus non linéaires utilises dans l'industrie alimentaire ", 1997 – ENSIA France.

[04] Branislav Hrúz et Mengchu Zhou ," Modeling and Control of Discrete-event Dynamic Systems/ with Petri Nets and Other Tools " , Published – 2007.

[05] T.EL Mezyani , " Méthodologie de surveillance des systèmes dynamiques hybrides ". Thèse de doctorat, spécialité automatique et informatique industrielle, préparé au laboratoire d'automatique Génie informatique et Signal UMR CNRS 8146 de l'université des sciences et technologies de Lille, 2005.

[06] P. Mosterman et G. Biswas, "A Hybrid Modeling and Simulation Methodology for Dynamic Physical Systems", Institute for Robotics and System Dynamics, 1998.

[07] Shahin Hashtrudi Zad, Thesis " Fault Diagnosis in discrete-event and hybrid systems " University Toronto, Canada 1999.

[08] Christos G. Cassandras et Stéphane La fortune , Introduction to Discrete Event Systems Second Edition.Boston University,1999 .

[09] M. Kurovszky, " Etude des Systèmes Dynamiques Hybrides par représentation d'état discrète et automate hybride ". Thèse de doctorat ,l'INPG, France , 2002.

[10] P. Bonhomme, " Réseaux de Petri P-Temporels : Contribution a la Commande Robuste " Thèse pour obtenir le grade de Docteur préparé à l'université de Savoie juillet 2001.

[11] Monika kurovszky , Thèse de doctorat " étude de système dynamiques hybrides par représentation d'état discrète et automate ` hybride Université Joseph Fourier - GRENOBLE 1 2002.

[12] Kebbas Salah, MÉMOIRE DE MAGISTER « Contribution à la Correction et l'Amélioration de la Qualité de Service dans une Entreprise Publique, en utilisant les Réseaux de Files d'Attente » Département de Génie Industriel- Faculté de Technologie - Université Hadj-Lakhdar -Batna , 2013.

[13] R. David et H. Alla ,"Continuous Petri Nets_Proceedings of the Eight European workshop on application and theory of Petri nets", Pages 275-294, Zaragoza (Espagne), juin 1987.

[14] A.BENHOCINE, Cours de post-graduation, " Théorie des graphes et applications" , Université de Sétif, 2001.

[15] BOUKHECHEM ISMAIL, Mémoire Fin D'étude « Etude Et Réalisation De Système Automatise Didactique Mise En Œuvre De L'automate Siemens S300»,2014 Département Electrotechnique , Université Constantine I .

[16] Dr. HERIZI Abdelghafour , Cours Système d' automatisés , [**Consulté le : 20 -02- 2023**] https://elearning.univ- msila.dz/moodle/pluginfile.php/120235/mod_resource/content/3

[17] MOHAMED LAMINE DILMI , Mémoire de Master " Contribution à la modélisation des systèmes automatisés par un outil graphique " 2014, Département d'Electrotechnique -Université Ferhat Abbas Sétif1.

[18] H. Nussbaumer,« informatique industrielle III», Press polytechniques de Romandes, 1987.

[19] Gilles MICHEL, Claude LAURGEAU et Bernard ESPIAU , « les automates programmables industrielle », Editeur : Paris [France] , Dunod, 1979

[20] DJEROUROU Yacer et ELKHALDI Oussama Mémoire fin d'étude " Régulation de niveau et débit d'un système hydraulique par automate – SIEMENS S7-300 et supervision via WINCC flexible" 2021 ,Institut de Maintenance et de Sécurité Industrielle ,Université d'Oran 2 Mohamed Ben Ahmed.

[21] : Philippe Le Brun " Les Automates Programmables Industriels (API) " [**Consulté le : 02 - 06 -2023**] , https://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm

[**22**] **:** Michel. G ,(1988) Les A.P.I **«** Architecture et applications des automates programmables industriels **»**.Publisher: Dunod, 1988**.**

[**23**] **:** Y. Boudouaoui , Cours Automate programmable " Semester 02 " , Centre Universitaire Nour Bachir -EI-BAYADH,2016.

[24] : BENMESSAOUD Idris et OUZIRI Asma , Mémoire fin d'études " Automatisation d'un atelier de production de yaourt dessert au niveau de la laiterie SOUMMAM " , Département de Génie Electrique - Université A.MIRA-BEJAIA, 2018.

[25] : BENREKIA HOCINE , BOUAMAMA ABDARAOUF et TOUTI ABDERRAHMANE , Mémoire " Automatisation d 'une porte coulissante Programmation avec STEP 7 ", Département d'Électronique - Université Mohamed El Bachir El Ibrahimi - Bordj Bou Arreridj –2021.

[26] Grasset, Confucius et les automates / l'avenir de l'homme dans la civilisation des machines-Essai De Charles-Edouard Bouée, François Roche. 2014

[27] sekhok.s et loukili.k « étude d'une boucle de régulation de niveau : implémentation du régulateur et régulage du procède »,projet de fin d'étude, 2011.

[28] Karimi Alireza et Salzmann Christophe « Automatique et commande numérique » Ecole polytechnique fédérale de Lausanne, 2008/2009.

[29] DJEROUROU Yacer et ELKHALDI Oussama Mémoire fin d'étude " Régulation de niveau et débit d'un système hydraulique par automate – SIEMENS S7-300 et supervision via WINCC flexible" 2021 Institut de Maintenance et de Sécurité Industrielle, Université d'Oran 2 Mohamed Ben Ahmed.

[30] BTS CIRA Rouvière Avant cours "Régulation Industrielle "- Lycée Rouvière. [**consulté le 12-04-2023**], http://cira83.com

[31] Andre Simon « Automate programmable niveau » Edition L'ELAN-LIEGE 1991.

[32] SIMATIC Logiciel de base pour S7-300/400 Régulation PID, manuel Siemens AG, 1996, consultéle"24/04/2023",http://www.global.hs-mittweida.de/~ifa/www-f/s7manual/s7pidcoa/s7

[33] Dif Nihed et Triki Razika , Mémoire de Fin d'études " Synthèse d'un contrôleur PID pour la commande d'une MCC " Université Larbi BenM'hidi – Oum El Bouaghi- Institut : Science Et Technologie-Ain Beida- Département D'électronique , 2011.

[34] Bouchema Sonia , Projet de fin d'étude "Conception d'une régulation de température dans un bac d'eau chaude au niveau de CEVITAL ",Département de Génie Electrique-Université A.MIRA-BEJAIA ,2019.

[35] Ferdjoukh Yacine, Memoire De Fin D'études En Vue De L'obtention Du Diplôme De Master En Automatique " Commande Prédictive Généralisée : Application Au Moteur A Excitation Séparée "Département De Génie Electrique-Université Mohamed Boudiaf – M'sila,2016.

[36] K. J. Astromet T. Hagglund «Advanced PID Control», ISA, New York, 2006.

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

[38] Benyahia Abdesselam et Mouhoubi Salim, Mémoire De Fin D'études" Implémentation D'un Régulateur Floue Dans Un Automate S7-300 Pour La Commande De La Vitesse D'une Machine Asynchrone", Département Du Génie Electrique- Université Dr. Yahia Fares De Medea ,2020.

[39] HAREK Ahmed et CHAOUCHE Mehdi, Mémoire de Fin d'Etudes " Régulation de vitesse d'un moteur à courant continu utilisant PID sous step7 à base d'un API S314 IFM" Département d'électronique- Université des freres mentouri constantine, 2015