

République Algérienne Démocratique et Populaire  
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique

Département d'Automatique

## Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologie

Filière : Automatique

Spécialité : Automatique et Informatique Industrielle

*Présenté par*

**Salim KAROUF  
Jugurta MAHLOUL**

Mémoire dirigé par **Dr K.BENALIA**

### Thème

**Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz avec la réalisation d'un prototype d'une surveillance intelligente à l'aide des techniques de Machine Learning et TinyML à base d'une carte Arduino.**

*Mémoire soutenu le 20 Octobre 2022 devant le jury composé de :*

**M. LAKHLEF Ahcene**

Professeur, UMMTO, Président

**M. BENALIA Karim**

MCA, UMBB, Encadrant

**M. MAOUI Mohamed**

Ingénieur et Doctorant, CO-Encadrant

**Mme KHERRAZ. K**

MCB, UMMTO, Examinatrice

Année universitaire 2021/2022

## *REMERCIEMENTS*

*C'est avec humilité et gratitude que nous reconnaissons ce que nous devons :*

*Nous commencerons par remercier et rendre grâce à Dieu tout puissant pour nous avoir donnée le courage et la volonté de mener à bon terme ce travail.*

*Nos remerciements s'adressent à nos familles et surtout à nos parents.*

*Nous tenons à remercier tous ceux qui nous ont aidés à réaliser ce travail notamment notre promoteur Mr BENALIA Karim qui n'a jamais ménagé son temps et ses efforts pour nous prodiguer ses conseils.*

*Nous exprimons notre profonde gratitude à M. MAOUI de nous avoir proposé ce sujet et pour son aide inestimable et ses précieux conseils.*

*Nos remerciements les plus chaleureux s'adressent à M. BOUSSALEM et M. MALKI Ainsi pour le temps qu'ils nous ont consacré, leurs orientations et encouragements et surtout pour leurs patience tout au long de ce travail.*

*Notre grand respect à nos professeurs, transmetteurs du savoir, qui nous ont aidé à soutenir notre volonté de réaliser ce mémoire, ne serait-ce, que par un encouragement, nous leurs disons mille merci.*

*Tous nos remerciements à l'ensemble du personnel de la centrale électrique de Cap Djinet pour leurs collaborations et leur accueil chaleureux.*

*Aux membres du jury, pour avoir bien voulu accepter de juger ce travail et lui apporter les corrections nécessaires.*

*Enfin nos remerciements vont à tous ceux qui dans l'anonymat et dans la grandeur de leur modestie ont contribué de loin ou de près à la réalisation de ce travail.*

# **DEDICACE**

*Je dédie ce modeste travail :*

*À mes chers parents, en témoignage et en gratitude de leurs dévouement, de leurs soutien permanent durant toutes mes années d'études, leurs sacrifices illimités, leurs réconfort moral, eux qui ont consenti tant d'effort pour mon éducation, mon instruction et pour me voir atteindre ce but, pour tout cela et pour ce qui ne peut être dit, mes affectations sans limite.*

*À ceux qui sont la source de mon inspiration et mon courage, à qui je dois de l'amour et de la reconnaissance :*

*À mon Cher frère Walid,*

*À toute ma famille.*

*À tous mes très chers amis et tous mes camarades.*

*À mon binôme Jugurta.*

*Salim*

## **DEDICACE**

*Je dédie ce modeste travail :*

*À mon cher défunt père qui durant son vécu a toujours souhaité  
instructions à ses enfants*

*À ma mère pour ses sacrifices qu'elle a fourni pour mes études, que dieu  
la protège.*

*À mes frères Rabah, Mouh et Massi.*

*À mes chers cousins Said, Aissa, Melissa, Fatima, Anaïs.*

*À tous mes très chers amis et tous mes camarades.*

*À mon binôme Salim.*

*Jugurta*

*Table des matières*

Remerciements .....	I
Table des matières .....	IV
Liste des figures.....	X
Liste des tableaux .....	XII
Notations .....	XIII
Introduction générale .....	1

**Chapitre I : Présentation et fonctionnement de la centrale de à cycle combinée de RAS - DJINET**

I.1. Introduction .....	4
I.2. Présentation générale de la centrale de RAS –DJINET .....	4
I.2.1. Position de la centrale .....	4
I.2.2. Constitution de la centrale .....	5
I.3. Principaux équipement mécanique de la centrale RAS-DJINET .....	6
I.3.1. Turbine à gaz (GT) .....	6
I.3.2. Générateur de vapeur à récupération de chaleur HRSG .....	6
I.3.3. Turbine à vapeur ST .....	7
I.3.4. L'embrayage TV .....	8
I.3.5. Alternateur .....	8
I.3.6. Condenseur 50 MAG10AX001 .....	9
I.3.7. Les pompes .....	9
I.3.8 Transformateur principale (GSUT) .....	9
I.3.9. Unité de dessalement d'eau de mer .....	10

I.3.10. Reservoir de stockage du gasoil .....	11
I.3.11. Salle de commande .....	12
I.4. Principe de fonctionnement de la centrale pour la production .....	12
I.5. Conclusion .....	14

## **Chapitre II : L'instrumentation et moyens d'automatisation**

II.1. Introduction .....	16
II.2. Capteurs et transmetteurs .....	16
II.2.1. Définition .....	16
II.2.2. Les capteurs de surveillance .....	16
II.2.3. Les transmetteurs .....	18
II.2.3.1. Définition .....	18
II.3. Les Pré –actionneurs .....	19
II.3.1. Distributeur .....	19
II.3.2. Relais .....	20
II.3.3. Contacteur .....	21
II.4. Les actionneurs .....	21
II.4.1. Les vannes .....	21
II.4.2. L'électrovane .....	23
II.4.3. Les convertiseurs .....	24
II.4.4. Les détecteurs .....	24
II.4.4.1. Détecteur de survitesse .....	24
II.4.4.2. Détecteurs d'usure de la butée SB12K011 .....	25
II.4.4.3. Détecteur du manque de vide au condenseur .....	26
II.5. Conclusion .....	26

## **Chapitre III : Etude de l'automatisation et de la supervision d'un système de commande de vitesse d'une turbine à gaz**

III.1. Introduction .....	28
III.2. Généralité sur l'automate programmable industriel (API) .....	28
III.2.1. Définition d'un automate programmable .....	28
III.2.2. Architecture d'un API .....	29
III.2.2.1. Architecture interne d'un API .....	29
III.2.2.2. Architecture externe d'un API .....	30
III.2.3. Programmation des API .....	30
III.2.4. Présentation de l'automate S7-300 .....	31
III.2.5. Programmation de l'automate S7-300 .....	32
III.2.6. Les différents blocs de S7 .....	33
III.3. Cadre pratique : Programmation et supervision.....	34
III.3.1. Le logigramme fonctionnel séquentiel du système de commande de vitesse .....	34
III.3.2. Réalisation du programme de commande de la vitesse d'une turbine à gaz sous STEP7 .....	39
III.3.2.0. Démarrage de STEP7 et Création d'un nouveau projet .....	39
III.3.2.1. Configuration matérielle d'une station SIMATIC.....	40
III.3.2.2. Créer une table de mnémoniques .....	40
III.3.2.3. Simulation et visualisation du programme .....	42
III.4. Développement de la plateforme de supervision .....	46
III.4.1. Présentation de logiciel WinCC flexible .....	46
III.4.2. Elaboration de la plateforme de supervision.....	47
III.5. Conclusion .....	51

## **Chapitre IV : Notion sur la machine Learning et le Tiny ML**

IV.1. Introduction.....	53
IV.2. Définition d'une machine Learning (ML) .....	53
IV.3. Utilisation et domaine d'application de machine Learning .....	54
IV.3.1. Problématique du ML .....	54
IV.3.2. Domaine d'application du ML .....	54
IV.4. Approche d'une Machine Learning.....	55
IV.4.1. Apprentissage automatique supervisé .....	56
IV.4.1.1. Problème de classification.....	57
IV.4.1.2. Problème de régression .....	58
IV.4.2. Apprentissage automatique non supervisé.....	59
IV.4.3. Apprentissage semi-supervisé .....	60
IV.4.4. Apprentissage par renforcement .....	60
IV.5. Quelques algorithme d'apprentissage supervisé .....	61
IV.5.1 Arbre de décision DT ( Decision Tree).....	61
IV.5.2. Les forets aléatoire (Random Forest).....	62
IV.5.3. Le réseau de neurones artificiels .....	64
IV.6. Machine Learning vs Deep Learning (DL).....	65
IV.7. Internet des objets , ou internet of things (IOT).....	66
IV.7.1. Définition .....	66
IV.7.2. Composants d'un objet collecté.....	67
IV.7.2.1. Le micro contrôleur.....	67
IV.7.2.2. Les capteurs et les actionneurs .....	68
IV.7.3. Protocoles de communication et plateformes de développement .....	69
IV.7.4. Domaines d'application .....	69

IV.8. TinyML : Machine Learning sur microcontrôleur .....	70
IV.8.1. Définition .....	70
IV.8.2. Les opportunités et les défis pour TinyML .....	71
IV.8.3. TensorFlow Light pour les microleurs .....	72
IV.9. Conclusion .....	73

## **Chapitre V : Elaboration d'un système de surveillance intelligent à l'aide TinyML**

V. 1. Introduction .....	74
V.2. Position du problème .....	74
V.3. Application et résultats .....	76
V.3.1. Construction d'une base de données d'apprentissage .....	76
V.3.2. Déploiement du modèle d'apprentissage sur un microcontrôleur Atmega328P d'une carte ArduinoUno .....	78
V.3.3. Le matériel utilisé dans le projet.....	83
V.3.4. Réalisation d'un Prototype (banc d'essai) .....	83
V.4. Conclusion .....	86
Conclusion générale .....	87
Bibliographie .....	89
Annexe	
Résumé	

## ***Liste des figures***

### **Chapitre I**

Figure I.1. Vue générale de la centrale RAS DJINET .....	4
Figure I.2. Constitution de la centrale RAS DJINET .....	5
Figure I.3. Schéma synoptique d'une turbine à gaz.....	6
Figure I.4. Schéma fonctionnel d'un HRSG.....	7
Figure. I.5. Schéma d'une turbine à vapeur de RAS-DJINET.....	7
Figure. I.6. L'embrayage .....	8
Figure. I.7. Schéma synoptique d'un alternateur de type Siemens .....	8
Figure I.8. Vue générale du transformateur principal (GSUT) de la centrale .....	10
Figure I.9. Schéma d'une station de dessalement d'eau de mer.....	11
Figure I.10. Réservoir de stockage gasoil .....	11
Figure I.11. Salle de commande de la centrale à cycle combiné RAS-DJINET .....	12
Figure I.12. Schéma explicatif du fonctionnement de la centrale.....	13

### **Chapitre II**

Figure II.1. Schéma fonctionnel d'un capteur .....	16
Figure II.2. Positionnement des capteurs de surveillance dans la turbine.....	17
Figure II.3. Capteur et transmetteur en situation .....	18
Figure II.4. Types de transmetteurs .....	18
Figure. II.5. Distributeur pneumatique .....	20
Figure II.6. Schéma d'un Relais.....	20
Figure II.7. Schéma illustre les composantes d'un contacteur .....	21
Figure II.8. Vanne d'arrêt et soupape de réglage .....	22
Figure II.9. Schéma d'un électrovanne.....	23
Figure II.10. Détecteur de survitesse .....	24
Figure II.11. Détecteur d'usure de la butée .....	25

## Chapitre III

Figure III.1. Structures générales d'un système automatisé .....	29
Figure III.2. Architecture d'un automate programmable industriel.....	30
Figure III.3. Schéma extérieur d'un API s7-300.....	32
Figure III.4. Le logigramme fonctionnel séquentiel du système de commande de vitesse	35
Figure III.5. Logigramme fonctionnel séquentiel (suite 1).....	36
Figure III.6. Logigramme fonctionnel séquentiel (suite 2).....	37
Figure III.7. Fenetre pour un nouveau projet.....	39
Figure III.8. Configuration matérielle.....	40
Figure III.9. Les blocs du programme .....	40
Figure III.10. La table des mnémoniques .....	41
Figure III.11. Les Blocs .....	42
Figure III.12. Simulation S7-PLCSIM.....	43
Figure III.13. Visualisation de quelques réseaux de notre programme .....	46
Figure III.14. Vue d'accueil.....	48
Figure III.15. Vue principale .....	48
Figure III.16. Vue d'alarmes.....	49
Figure III.17. Vue du graphe de variation de fréquence de vitesse de la turbine à gaz ....	50
Figure III.18. Alarme TOR .....	50
Figure III.19. Alarme analogique.....	50

## Chapitre IV

Figure IV.1 Taxonomie des différentes techniques issues du ML.....	55
Figure IV.2. Schéma synoptique qui illustre l'Apprentissage automatique supervisé .....	56
Figure IV.3. Exemple la classification de spam .....	57
Figure IV.4. Problème de Régression .....	58
Figure IV.5. Schéma synoptique qui illustre l'apprentissage automatique non supervisé .	59
Figure IV.6. Apprentissage par renforcement.....	60
Figure IV.7. Exemple de l'arbre de décision (DT) .....	61
Figure IV.8. Schéma d'illustration du concept d'agrégation bootstrap (Bagging).....	63
Figure IV.9. Réseau de neurones biologique .....	64

Figure IV.10. Neurone biologique VS Neurone artificiel.....	65
Figure IV.11. ML vs DL .....	66
Figure IV.12. Carte Arduino UNO .....	67
Figure IV.13. Le microcontrôleur Raspberry Pi 3 Modele B .....	68
Figure IV.14. Les six piliers d'une Smart City .....	70
Figure IV.15. Conception du TensorFlow .....	73

## **Chapitre V**

Figure V.1. Alarme de sous-vitesse en simulink .....	75
Figure V.2. Schéma synoptique pour l'apprentissage automatique de défaut de vitesse .....	76
Figure V.3. Un aperçu sur le code d'installation de MicroML.....	78
Figure V.4. Un aperçu sur le code de configuration de l'environnement Python .....	79
Figure V.5. Programme d'une fonction pour gérer le jeu de données.....	80
Figure V.6. Commande de chargement des données d'apprentissage .....	81
Figure V.7. Code d'entraînement du modèle d'apprentissage .....	81
Figure V.8. Fonction de transfert du modèle en code C/C++ simple .....	82
Figure V.9. Déploiement du modèle sur le microcontrôleur ATmega328P .....	82
Figure V.10. Schéma synoptique global d'Interconnexion et câblage avec L'arduino .....	84
Figure V.11. Le banc d'essai de notre projet .....	85

## **Liste des tableaux**

### **Chapitre III**

Tableau III.1. Les paramètres de protection de fréquence de la turbine .....38

### **Chapitre V**

Tableau V.1. Tableau de base d'apprentissage pour le défaut de sous-vitesse.....77

Tableau V.2. Les différents composants utilisés dans notre projet .....83

## **Notations**

TAV : Turbine à Vapeur.

TAG : Turbine à Gaz.

NG : Gaz Naturel.

HRSG : Heart Recovery Steam Generator.

HP : Haute Pression.

MP : Moyenne Pression.

LP : Low Pression.

TV : Turbine à Vapeur.

TG : Turbine à Gaz.

GSUT : Generator Step Up Transformer.

TP : Transformateur Principal.

BU : Borne Usine.

ML : Machine learning.

DMP : Data Management Platform.

SEA : Search Engine Advertising.

SVM : Support Vector Machine.

SVR : Support Vector Regression.

GLM : Modèle Linéaire Généralisé.

RNA : Réseau de neurones artificiels.

DT : Arbre de décision.

RF : Forêt Aléatoire.

DL : Deep Learning.

IOT : Internet Of Things.

IDO: Internet Des Objets.

WWW : World Wide Web.

CTIM : la température à l'entrée du compresseur.

TOR : Tout Ou Rien.

API : Automate Programmable Industriel.

PLC : Programable Logic Controller.

PO : Partie Opérative.

PC : Partie Commande.

PR : Partie Relation.

E/S: Entrée Sortie.

CPE : Centrale Processing Unite.

PS : Module D'alimentation.

IM : Coupleur.

CP : Processeur de Communication.

FM : Module de Fonctinnement.

SM : Module de Signaux.

LIST : Language liste.

LOG : Language Logigramme.

CONT : Language Contatct.

OB : Bloc d'organisation.

FB : Bloc Fonctionnel.

FC : Fonction.

DB : Bloc de Données.

SFB : Bloc Fonctionnel Système.

SFC : Fonction Système.

SDB : Données Système.

ONU : Organisation Des Nations Unies.

BOOL : Booléen.

## Introduction générale

---

La production d'électricité, qu'elle soit assurée par un ou plusieurs centrales utilisant de l'énergie renouvelable ou conventionnelle, reste l'un des plus importants segments dans l'architecture globale du secteur électrique. En Algérie, l'énergie électrique est produite principalement à partir du gaz naturel. La part de la puissance installée de l'ensemble des centrales utilisant cette énergie primaire dépasse les 96 %, le reste des énergies employées sera partit entre le gasoil dans les centrales diesel et l'eau dans les centrales hydroélectriques [1]. L'énergie électrique reste toujours un besoin primordial pour le fonctionnement d'une machine ou d'une entreprise. Néanmoins, il faut assurer un rendement important afin d'améliorer la rentabilité d'une installation de production [1].

Dans un contexte de développement et de progression, les entreprises de la production d'électricité tel que la centrale thermique de RAS- DJENET- SONELGAZ (Société nationale de l'électricité et du gaz), sont obligées d'automatiser de plus en plus leurs installations de production et la supervision des unités et des machines entre dans ce contexte. Cette supervision est une forme évoluée du dialogue Homme-Machine, qui consiste à surveiller l'état de fonctionnement d'un procédé dont les possibilités vont bien au-delà de celles des fonctions de conduite et de contrôle réalisées avec des interfaces.

Dans notre travail, on a effectué un stage pratique à la centrale thermique à cycle combiné de RAS-DJENET. On s'est intéressé particulièrement à l'étude d'un système de commande de vitesse et à la surveillance automatique de défaut de vitesse de la turbine à gaz à l'aide des techniques d'apprentissage automatique.

L'introduction des automates dans les procédés industriels permet un gain de temps, une souplesse accrue dans la manipulation, une grande fiabilité et une maintenance des équipements facile. L'automatisation d'un procédé industriel consiste à assurer la conduite et la commande par un dispositif technologique.

De ce fait, l'objectif de la première partie de notre projet est de remplacer le système de la logique câblée de la commande de vitesse d'une turbine à gaz qui existe dans la centrale de Ras-Djanet par un système automatisé en utilisant un automate programmable industriel SIEMENS S7-300.

# Introduction générale

---

D'un autre côté, nous avons également constaté pendant notre stage pratique que ces systèmes de machines tournantes sont complexes et ils sont composés de plusieurs éléments sensibles soumis à certains défauts et risques opérationnels.

En effet, le but de la deuxième partie de ce travail est de proposer un modèle de surveillance intelligente pour la détection des défauts d'une turbine à gaz, en utilisant une approche basée sur l'apprentissage automatique.

Ainsi, ce travail consiste à confirmer l'efficacité des techniques d'apprentissage automatique dans la surveillance intelligente en détectant un défaut de vitesse dans les turbines HP qui s'avère être très complexe par son seuil qui varie en fonction de la température .

Dans ce contexte nous allons réaliser un banc d'essai instrumenté, permettant de surveiller la vitesse de la turbine, à travers l'acquisition des données d'apprentissage sur ordinateur ce qui permet de visualiser les paramètres de cette dernière, tel que la vitesse de rotation et la température. Ce banc d'essai sera commandé par un système embarqué à base d'un microcontrôleur ATmega328P d'une carte ArduinoUno.

Le plan de rédaction de ce mémoire est de la manière suivante :

Dans le premier chapitre, nous présenterons une description technologique générale de la centrale thermique à cycle combiné de RAS-DJENET.

Dans le deuxième chapitre, nous allons faire une description de tous les éléments d'instrumentation associés à la centrale.

Le troisième chapitre est consacré à la programmation et l'élaboration de la supervision de système de commande de vitesse de la turbine à gaz.

Dans le quatrième chapitre, nous décrirons tous les concepts de base relatifs à l'apprentissage automatique, afin d'expliquer aux lecteurs cette approche.

Le cinquième chapitre, nous allons réaliser un banc d'essai pour la surveillance de défaut de vitesse de la turbine commandé par un système embarqué à base d'un microcontrôleur ATmega328P d'une carte ArduinoUno.

Enfin, nous terminons notre travail par une conclusion générale.

# **CHAPITRE I**

## **Présentation et fonctionnement de la centrale à cycle combinée de RAS-DJINET.**

---

**« Au grand porte de l'entreprise  
S'ouvre la pratique pleine de surprise. »**

*Joubert Joseph*

---

## I.1. Introduction

Les Centrales à Cycle Combiné Gaz sont des unités de production d'électricité basées sur l'association d'une turbine à gaz (TAG) et d'une turbine à vapeur (TAV), ont connu un essor important dans le monde au cours des années 90. La baisse des prix du gaz par rapport à ceux du pétrole et la mise sur le marché de turbines de forte puissance ont fortement favorisé leur développement.

La Centrale de RAS DJINET (relevant de La Société Algérienne de l'Electricité et du Gaz-Distribution, par abréviation *SONELGАЗ*) est formée de trois turbines fonctionnant en cycle combiné (30% vapeur et 70 % gaz), dotée chacune d'une capacité de production de 400 MW d'électricité /Heure, soit un total de 1200 MW pour les trois turbines [1].

Dans ce chapitre nous allons présenter une description générale de la centrale en cycle combiné de RAS-DJINET, ainsi que son principe de fonctionnement.

## I.2. Présentation générale de la centrale RAS-DJINET

### I.2.1. Position de la centrale

La nouvelle centrale thermique à système combiné est située à la commune de RAS DJINET au bord de la mer dans la Wilaya de BOUMERDES. Elle alimentera la région centre du pays et renforcera sa capacité d'alimentation en énergie électrique via trois sous stations (AKBOU, EL AFFROUN, SI MUSTPHA).

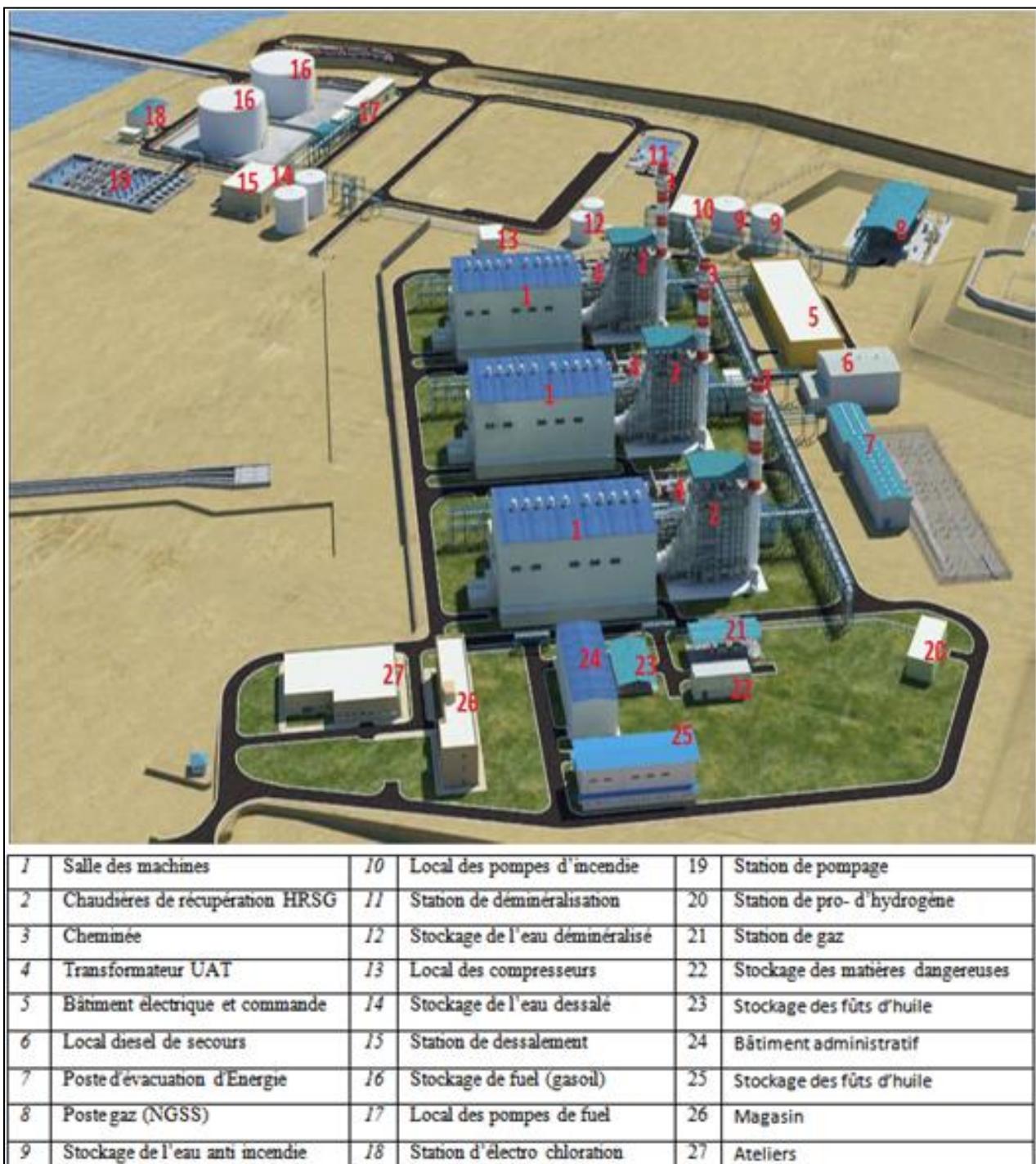


**Figure I.1. Vue générale de la centrale RAS DJINET [1]**

Elle est composée de trois groupes, dont chacun délivre approximativement une puissance de 400 MW.

### I.2.2. Constitution de la centrale

La nouvelle station de la centrale à cycle combinée est constituée de plusieurs unités comme indique la figure I.2.



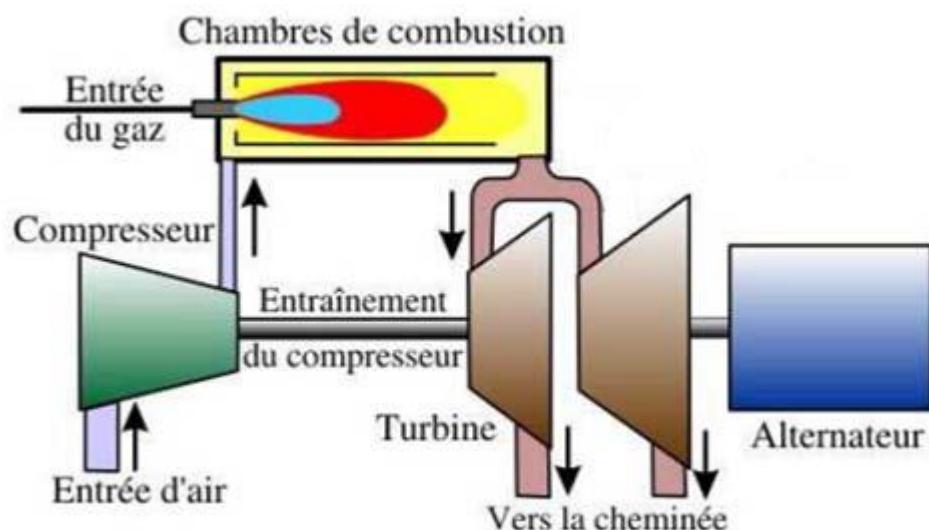
**Figure I.2. Constitution de la centrale RAS DJINET [1].**

### **I.3. Principaux équipements mécaniques de la centrale RAS-DJINET**

#### **I.3.1. Turbine à gaz (GT)**

Les turbines à gaz à chambre de combustion annulaire Siemens AG sont des turbines à arbre unique. Il s'agit d'une turbine à gaz mono-arbre avec un rotor commun utilisé par le compresseur et la turbine [2].

La turbine à gaz est conçue pour fonctionner en double allumage. Le gaz naturel (NG) doit être utilisé comme combustible primaire et le mazout doit être utilisé comme combustible de secours.



**Figure I.3. Schéma synoptique d'une turbine à gaz**

#### **I.3.2. Générateur de vapeur à récupération de chaleur HRSG**

Un générateur de vapeur à récupération de chaleur (HRSG) est un échangeur de chaleur à récupération d'énergie qui récupère la chaleur d'un flux de gaz chaud, comme une turbine à combustion ou un autre flux de gaz résiduaires. Il produit de la vapeur qui peut être utilisée pour entraîner une turbine à vapeur en cycle combiné.

L'HRSG se compose de quatre composants principaux : l'économiseur, l'évaporateur, le surchauffeur et le préchauffeur d'eau. Les différents composants sont assemblés pour répondre aux exigences de fonctionnement de l'unité (voir Figure I.4).

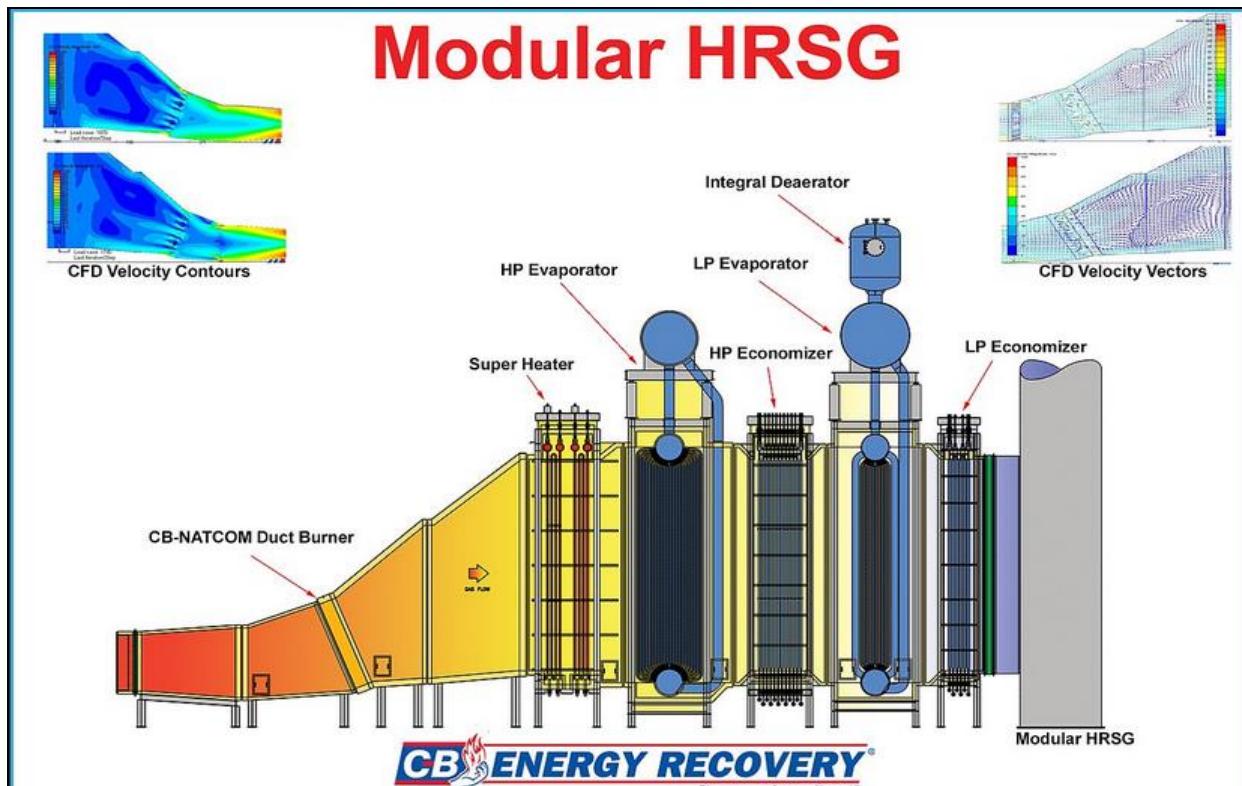


Figure I.4. Schéma fonctionnel d'un HRSG [2]

### I.3.3. Turbine à vapeur TV

La turbine transforme l'énergie thermique contenue dans la vapeur provenant de la chaudière en un mouvement de rotation de l'arbre. Le travail mécanique obtenu sert à entraîner l'alternateur.

La turbine à vapeur de RAS DJINET est constituée de trois (03) corps : HP, MP et BP, elle contient aussi six (06) soutirages de vapeur : alimentant 03 réchauffeurs BP, la bâche alimentaire et 02 réchauffeurs HP [2].

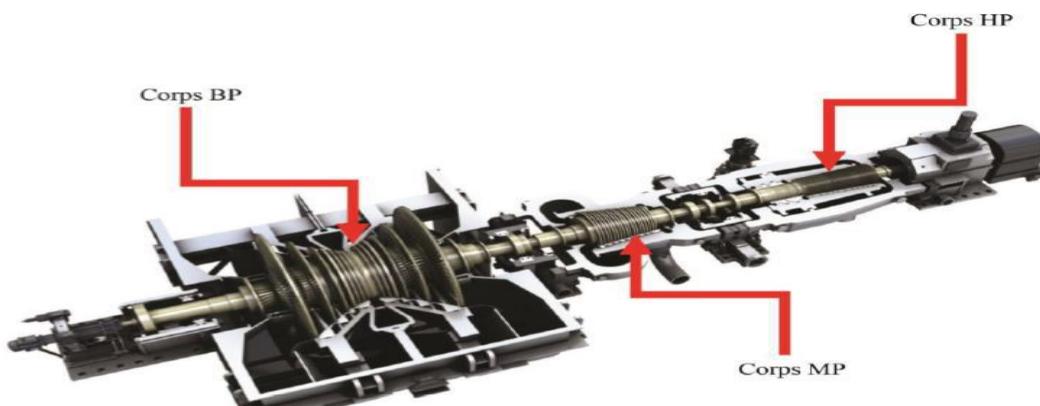
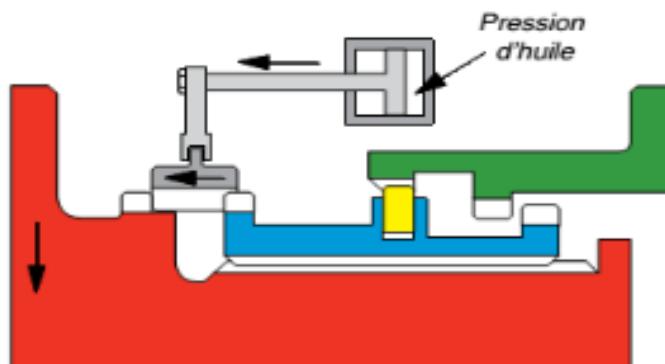


Figure I.5. Schéma d'une turbine à vapeur de RAS-DJINET

### I.3.4. L'embrayage TV (turbine à vapeur)

L'embrayage permet de connecter la TV à l'alternateur, alors que celui-ci est déjà entraîné par la TG.

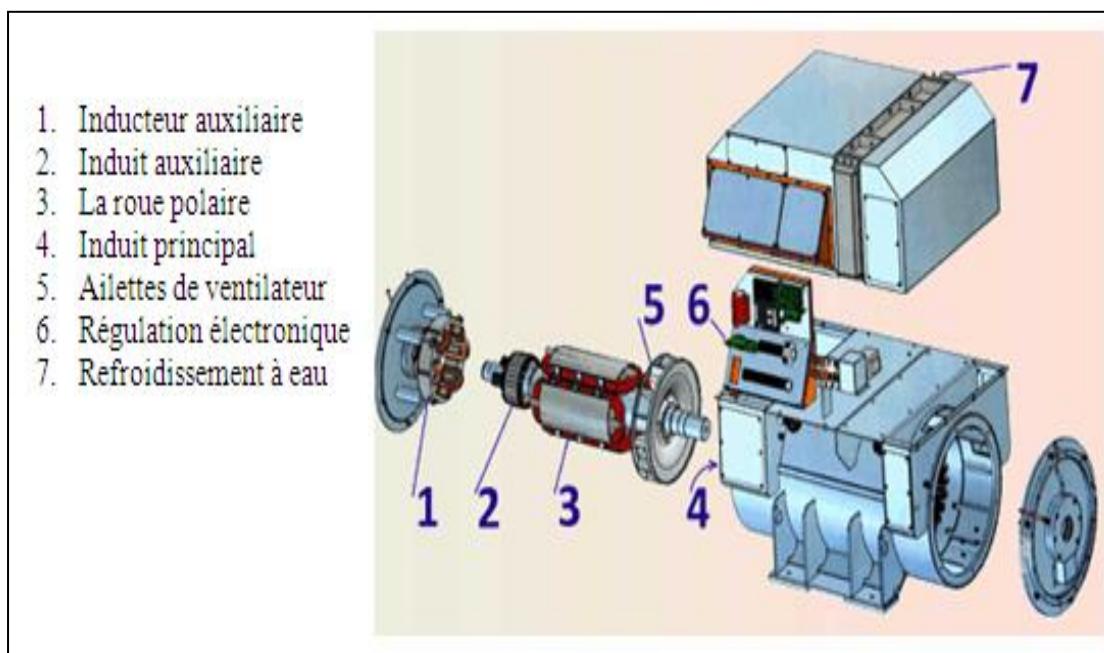
L'embrayage s'engage automatiquement aussitôt que le couple de la TV devient positif et se désengage automatiquement aussitôt que le couple de la TV devient négatif.



**Figure I.6. L'embrayage de TV**

### I.3.5. Alternateur

C'est un générateur d'électricité de type Siemens (SGEN5-2000H). Il sert à transformer l'énergie mécanique produite par l'arbre de la turbine en énergie électrique, il est l'équipement électrique le plus important de toute la centrale électrique. Le générateur est capable d'évacuer le courant de la sortie de la turbine (turbine à gaz + turbine à vapeur).



**Figure I.7. Schéma synoptique d'un alternateur de type Siemens**

### I.3.6. Condenseur 50 MAG10AX001

Le condenseur axial (50MAG10AC001) est un échangeur thermique refroidi à l'eau remplissant la fonction de condenser la vapeur d'échappement de la turbine à vapeur et/ou du système de BYPASS de vapeur haute pression (HP), pression intermédiaire (IP) et basse pression (LP). Le condenseur reçoit la vapeur d'échappement de la turbine dans le sens horizontal.

Le condenseur est équipé d'une soupape de dépression, de transmetteurs de pression, de transmetteurs de température et de transmetteurs de niveau pour la protection du condenseur et de la turbine.

### I.3.7. Les pompes

#### ➤ Pompes d'alimentation (50LAB10AP001 / 50LAB20AP001)

La fonction des pompes d'alimentation en eau de la chaudière est de fournir de l'eau du Ballon LP, à la section économiseur de pression (HP et IP) du générateur de vapeur à récupération (HRSG).

Les pompes d'alimentation en eau de la chaudière fournissent également de l'eau de pulvérisation désurchauffeur pour le système de BYPASS HP et de l'eau de pulvérisation désurchauffeur pour la surchauffeur HP [2].

#### ➤ Pompes d'extraction (50LCB10AP001/50LCB11AP001)

La fonction des pompes d'extraction de condensat est d'effectuer ce qui suit :

- ✓ Maintenir le niveau du puits du condenseur à son niveau normal en extrayant les condensats du puits du condenseur.
- ✓ Fournir de l'eau de condensation au dégazeur en passant par l'économiseur LP
- ✓ Fournir de l'eau pulvérisée désurchauffeur pour les systèmes de BYPASS IP et LP.

### I.3.8. Transformateur principal (GSUT)

Le transformateur principal (GSUT) est connecté entre le générateur et le poste d'évacuation d'énergie principal de 400kV à isolation gazeuse. La puissance générée par le générateur est élevée par le GSUT pour l'évacuation à 400kV. Cette même puissance est doit être utilisée pour l'alimentation des charges auxiliaires de la centrale provenant du réseau de 400kV pendant le démarrage [3].



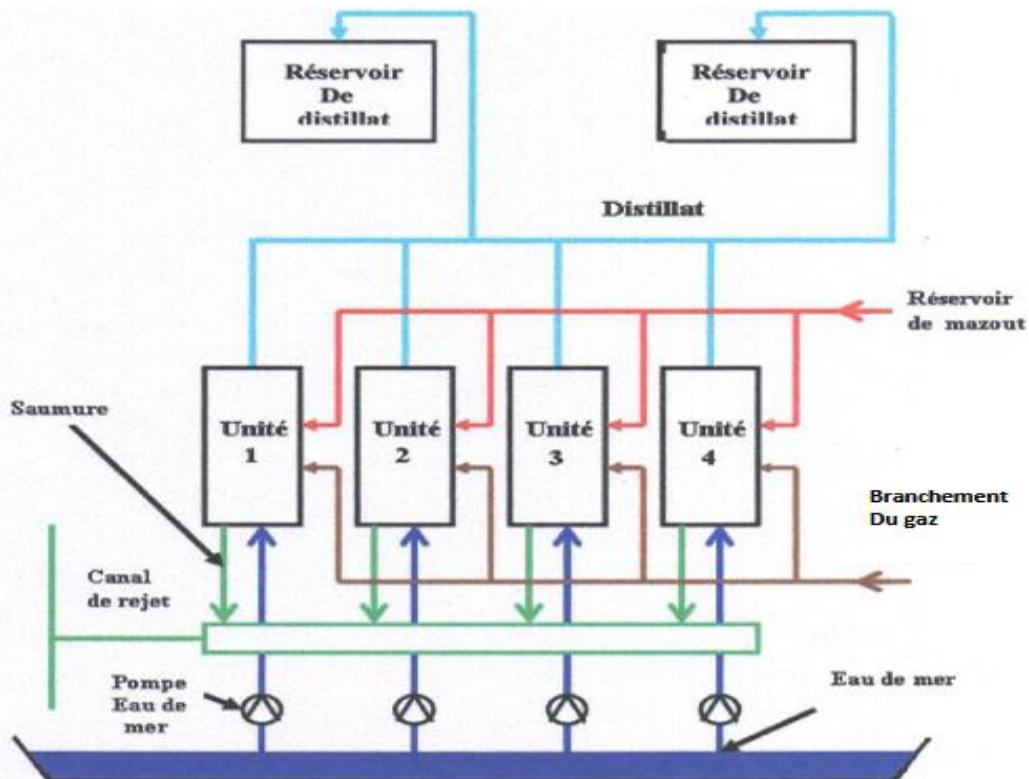
**Figure I.8. Vue générale d'un transformateur principal (GSUT) de la centrale [1].**

### I.3.9. L'unité de dessalement d'eau de mer

Toute centrale, type vapeur dispose généralement d'une installation de dessalement et de déminéralisation de l'eau brute (eau de mer) pour alimenter les chaudières.

Les besoins journaliers en eau dessalée de la centrale thermique sont de l'ordre de 1100 m<sup>3</sup>.

L'installation comprend quatre unités de dessalement indépendantes les unes des autres (trois unités sont en service et la quatrième est désinstallée), fonctionnant selon le principe de la distillation par détentes successives à 18 étages, et qui produit 500 m<sup>3</sup> d'eau dessalée par jour chacune, qui sera stockée dans deux réservoir de 2700 m<sup>3</sup> [2].



**Figure I.9. Schéma d'une station de dessalement d'eau de mer [1]**

### I.3.10. Réservoir de stockage gasoil

Pour le combustible de réserve (gasoil/fuel) qui est utilisé en cas de rupture du gaz naturel est reçu via des camions citernes et stocké dans des réservoirs puis livré à chaque turbine à gaz au moyen des pompes d'acheminement de combustible. [1]



**Figure I.10. Réservoir de stockage gasoil**

### **I.3.11. La Salle de commande**

C'est là où en vielle à la bonne exécution de la centrale, est ont gérer l'état des machines et les différents problèmes qui peuvent ce poser. C'est un sort intermédiaire entre l'ingénierie et la machine.



**Figure I.11. Salle de commande de la centrale à cycle combiné RAS-DJINET [3].**

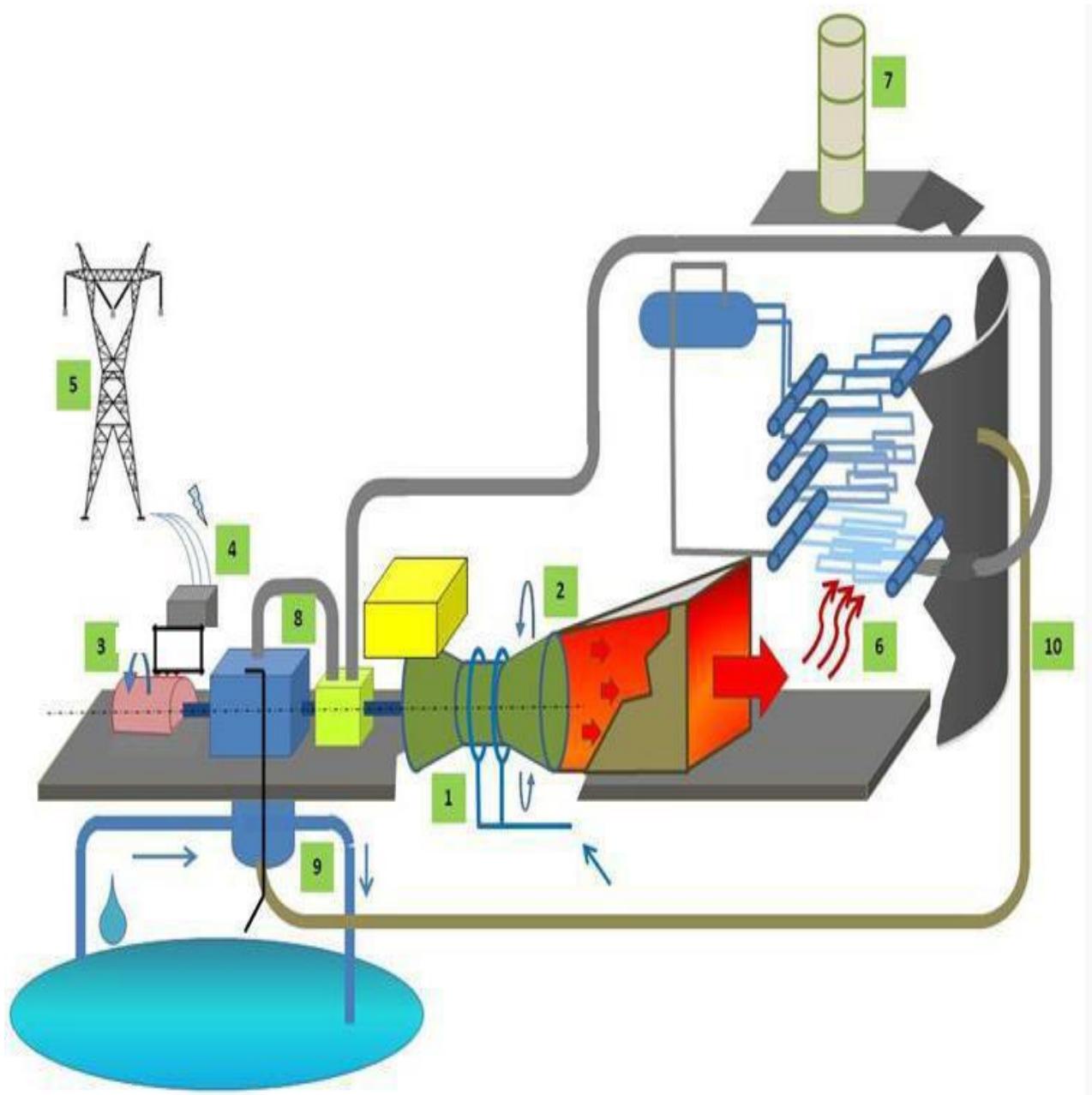
#### ➤ **Un système de contrôle-commande SPPA-T3000 (Interface utilisateur)**

L'interface utilisateur est mise à disposition sur un client léger avec un navigateur Web. L'accès au processus est possible depuis la salle de commande ou via d'autres liaisons de communication sécurisées. Seuls des utilisateurs autorisés peuvent interroger l'état actuel des tranches et les paramètres d'exploitation, ou commander le processus.

### **I.4. Principe de fonctionnement de la centrale pour la production**

Le système de production d'électricité dans la centrale RAS-DJINET est composé de deux cycles, d'où son nom de cycle combiné :

Le premier cycle est constitué par la turbine à gaz et le deuxième cycle s'effectue au niveau de la sortie de gaz des gaz chauds de la turbine à gaz (voir figure I.12).



**Figure I.12. Schéma explicatif du fonctionnement de la centrale.**

- ❖ **Le premier cycle** est constitué par la turbine à gaz (1). Celle-ci est constituée de quatre étages : dans le premier il y a un compresseur qui comprime de l'air venant alimenter un second étage contenant la chambre de combustion où le gaz naturel est brûlé. La combustion effectuée, les gaz chauds viennent faire tourner les ailettes du troisième étage de détente de la turbine à combustion. Cette dernière est solidaire d'un axe qui entraîne un alternateur (3) qui produit l'électricité. Le transformateur (4) élève la tension pour permettre son injection dans le réseau électrique (5). [2]

❖ **Le deuxième cycle** a lieu à la sortie des gaz chauds de la turbine à gaz (2). Ces gaz sont injectés dans une chaudière de récupération (6) puis sont ensuite évacués dans l'atmosphère par la cheminée. Leur composition est de l'ordre de 86% d'air, 9% de vapeur d'eau et 4% de CO<sub>2</sub>. La vapeur produite dans cette chaudière est utilisée dans une turbine à vapeur (8), elle-même couplée à l'axe de l'alternateur (3). La vapeur, après avoir restitué son énergie, est envoyée dans un condenseur (9) pour repartir ensuite vers la chaudière de récupération sous forme d'eau (10). Puis cette eau sera de nouveau vaporisée. [2]

L'énergie électrique fournie par alternateur va être évacuée à travers le disjoncteur alternateur ou disjoncteur groupe pour ensuite aller vers le transformateur principal qui verra la tension de **15,5KV** à **400KV**. Après le transformateur principal TP, énergie fournie transitera à travers le disjoncteur ligne (poste blindé **SF6**) pour alimenter le réseau **400KV** au niveau des postes Affroun, Akbou et Si Mustapha. [2]

A noter que juste avant le **transfo TP**, il y a un soutirage d'une partie de l'énergie électrique produite qui ira alimenter le transformateur de soutirage **TS** pour transformer la tension de sortie alternateur **22KV en 6KV** et alimenter ainsi les auxiliaires électriques du groupe. C'est une auto alimentation des auxiliaires propres au groupe qui consommeront environ **8MW** par tranche. On fournira donc une puissance unitaire **BU** (borne usine) de **168 MW** par groupe. La puissance totale fournie au réseau. [5]

### I.5. Conclusion

Dans ce chapitre nous avons présenté la centrale à cycle combiné de RAS-DJINET avec ces principaux équipements dont on a expliqué le principe et le fonctionnement de chaque composant de production d'énergie électrique. Le chapitre suivant sera consacré pour la partie instrumentation de notre système étudié.

## Partie I :

---

# Etude de l'automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz.

---

- Chapitre II : L'instrumentation et moyens d'automatisation
- Chapitre III : Automatisation et supervision
  - Le logigramme fonctionnel séquentiel du système de commande de vitesse (cahier de charge)
  - Programmation de la commande de la vitesse d'une turbine à gaz sous STEP7.
  - Simulation et visualisation du programme
  - Développement de la plateforme de supervision

## CHAPITRE II

L'instrumentation et moyens  
d'automatisation.

---

### II.1. Introduction

Le contrôle est nécessaire pour l'observation, le suivi et le réglage des valeurs de différents paramètres qui caractérisent la centrale de RAS-DJENET, cela nécessite un ensemble d'éléments d'instrumentation. Dans cette optique, le rôle de l'instrumentation industrielle (des capteurs et des actionneurs) prend toute son importance, car ce sont les capteurs qui lui permettent de récupérer les données dont il a besoin et les actionneurs lui permettent d'interagir avec le monde extérieur.

Nous aborderons dans ce chapitre les différents capteurs et actionneurs qui caractérisent le système des turbines de la centrale RAS-DJENET.

### II.2. Capteurs et transmetteurs

#### II.2.1. Définition

Le terme **capteur** désigne un *dispositif d'entrée* qui fournit une sortie utilisable en réponse à une entrée physique donnée. Un capteur est un élément d'un appareil de mesure auquel est directement appliquée une grandeur à mesurer et dont le signal de sortie n'est pas directement utilisable comme signal d'entrée dans une boucle de mesure ou de régulation.

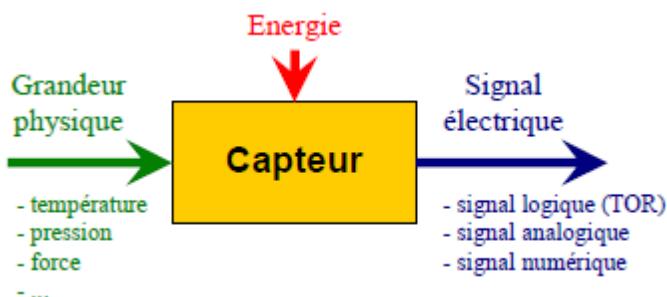


Figure II.1. Schéma fonctionnel d'un capteur.

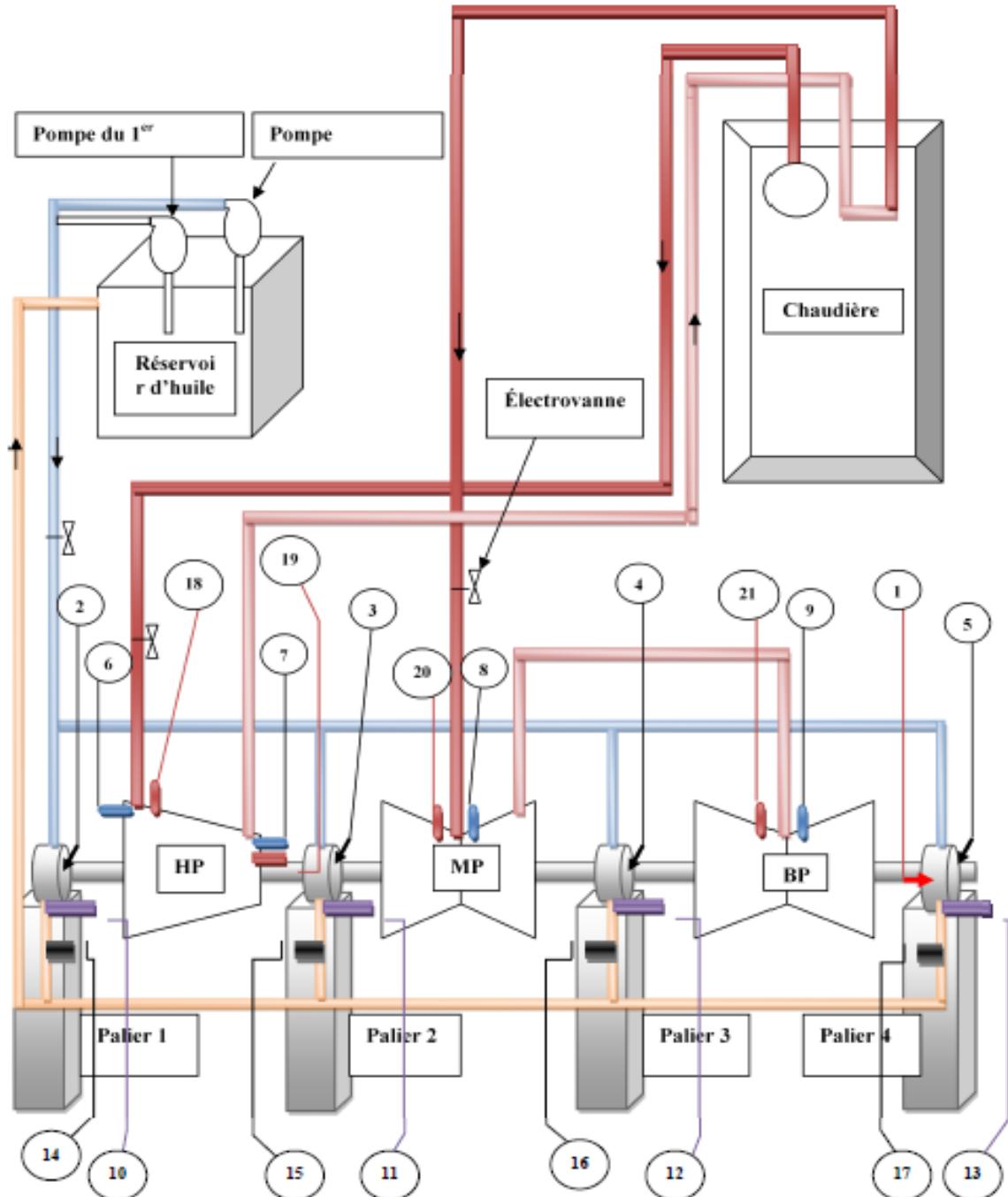
#### II.2.1. Les capteurs de surveillance

Les capteurs sont des éléments indispensables à tout projet d'automatisation, d'où on dispose dans notre système les capteurs suivants (Figure II.2):

- Un capteur de vitesse est en avant de l'arbre de la turbine (1);
- Quatre capteurs de vibrations pour les quatre paliers de la turbine (2) (3) (4) (5);

## CHAPITRE II : L'instrumentation et moyens d'automatisation

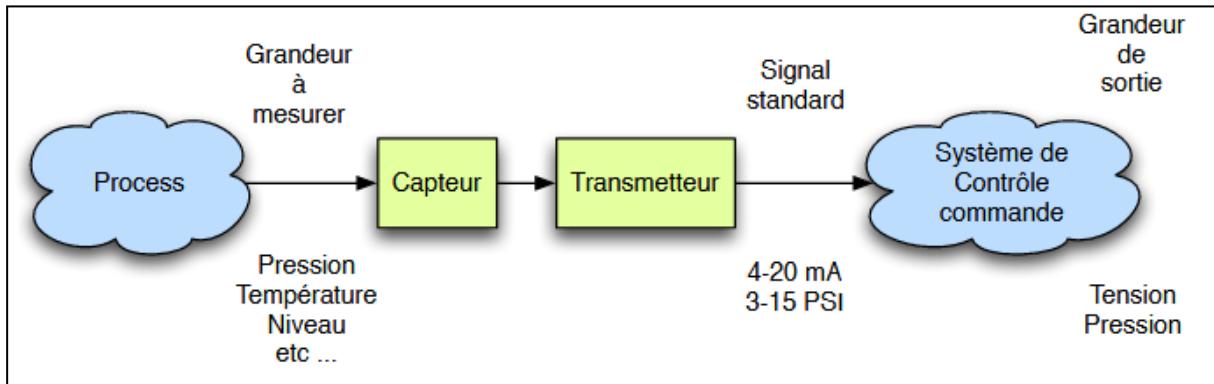
- Quatre capteurs de pression : deux à l'entrée et à la sortie du corps HP(7) (6), un à l'entrée du corps MP (8) et un à la sortie du corps BP (9) ;
- Quatre capteurs de débit d'huile dans les quatre paliers (10) (11) (12) (13) ;
- Quatre capteurs de température d'huile à la sortie de chaque palier (14) (15) (16) (17) ;
- Quatre capteurs de température : deux à l'entrée et à la sortie du corps HP (19) (18), un à l'entrée du corps MP (20) et un à la sortie du corps BP (21) ;



**Figure II.2. Positionnement des capteurs de surveillance dans la turbine [3].**

### II.2.3. Les Transmetteurs

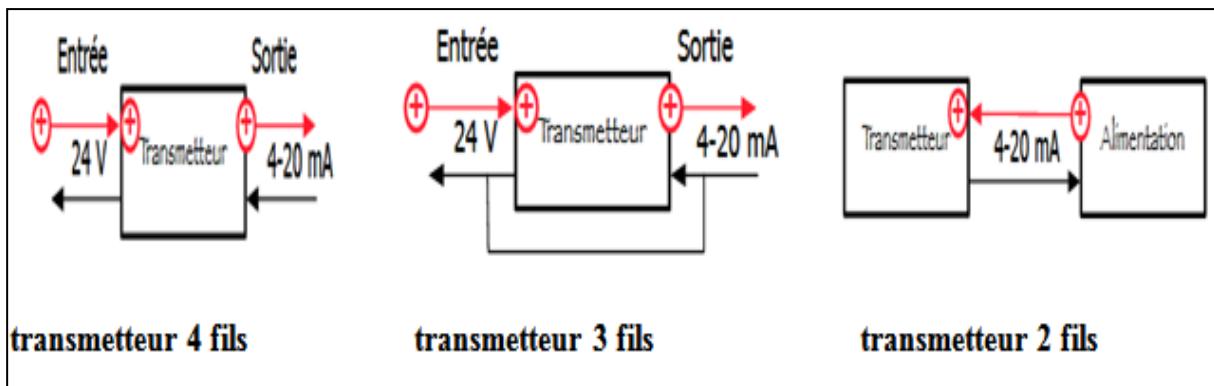
Un transmetteur est un dispositif qui convertit le signal de sortie du capteur en un signal de mesure standard. Il fait le lien entre le capteur et le système de contrôle commande (voir figure II.3). Le couple capteur+transmetteur réalise la relation linéaire entre la grandeur mesurée et son signal de sortie.



**Figure II.3. Capteur et transmetteur en situation**

On distingue trois types de transmetteur :

- 1) Les transmetteurs 4 fils (dits actifs) qui disposent d'une alimentation et qui fournissent le courant I. Leur schéma de câblage est identique à celui des régulateurs.
- 2) Les transmetteurs 3 fils sont des transmetteurs 4 fils, avec les entrées moins reliées.
- 3) Les transmetteurs 2 fils (dits passifs) qui ne disposent pas d'une alimentation et qui contrôlent le courant I fourni par une alimentation externe.



**Figure II.4. Types de transmetteurs**

Dans notre système on dispose des transmetteurs suivants [3]:

➤ **Transmetteurs de température :**

- ✚ Température admission HP, SA11 T133/144
- ✚ Température palier 1 SB11 T021- 24
- ✚ Température palier 2 SB12 T011- 14
- ✚ Température palier 4SB14 T 014/18
- ✚ Température palier de butée SB 12 T 031-34
- ✚ Température huile de graissage SC17 T016

➤ **Transmetteurs de pression :**

- ✚ Pression vapeur échappement HP, SA12P036
- ✚ Pression vapeur entrée MP, SA12 P051-57
- ✚ Pression huile de commande SC11 POO1
- ✚ Pression huile Pression de graissage SC17 P006
- ✚ Pression huile de soulèvement SC18 P001
- ✚ Pression huile palier de la butée SC17 P031
- ✚ Transmetteur de niveau caisse à l'huile turbine SC10 L001/2/3.

### II.3. Les Pré-actionneurs

C'est un organe capable de réaliser la commande d'un actionneur, en lui distribuant un courant fort, tout en étant commandé par un courant faible provenant de la partie commande. Le dispositif est intégré à la partie opérative ou à l'interface et dimensionné en fonction de l'énergie demandée par l'actionneur.

#### II.3.1. Distributeur

Distributeur est un pré-actionneur qui permet de :

- ✓ contrôler le mouvement de la tige d'un vérin ou la rotation d'un moteur électrique, hydraulique ou pneumatique (distributeurs de puissance).
- ✓ choisir le sens de circulation d'un fluide (aiguiller, dériver, etc.).
- ✓ exécuter à partir d'un fluide des fonctions logiques (fonctions ET, OU, mémoire, etc.).
- ✓ démarrer ou arrêter la circulation d'un fluide (robinet d'arrêt, bloqueur, ...).

Les éléments de base d'un distributeur pneumatique sont :

## CHAPITRE II : L'instrumentation et moyens d'automatisation

Le corps (en alliage léger), Le tiroir ou le coulisseau (cylindrique en acier), Les pilotages (pneumatiques ou électriques) et une série de joints.

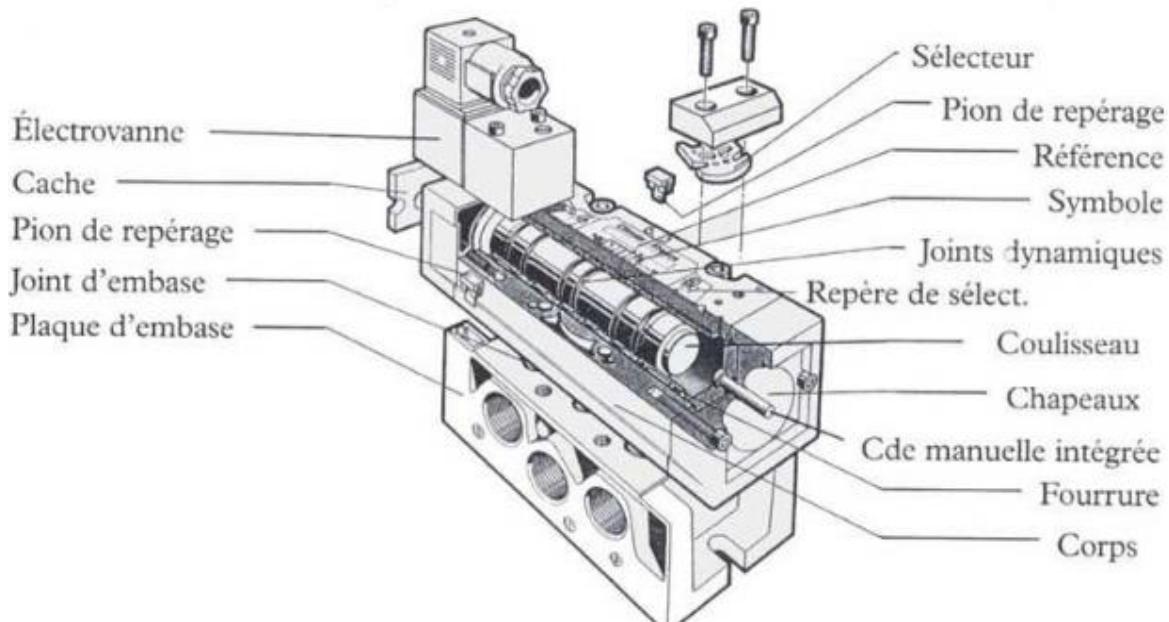


Figure II.5. Distributeur pneumatique

### II.3.2. Relais

Un relais est un composant électromagnétique qui permet la transition entre un courant faible et un courant fort. Il est constitué d'une bobine alimentée par le circuit de commande dont le noyau mobile provoque la communication de contacts pour alimenter un circuit de puissance. (Voir figure II.6).

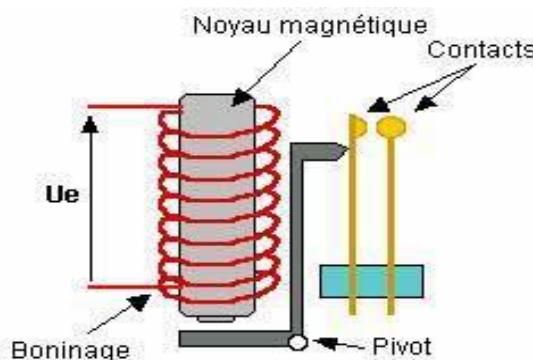
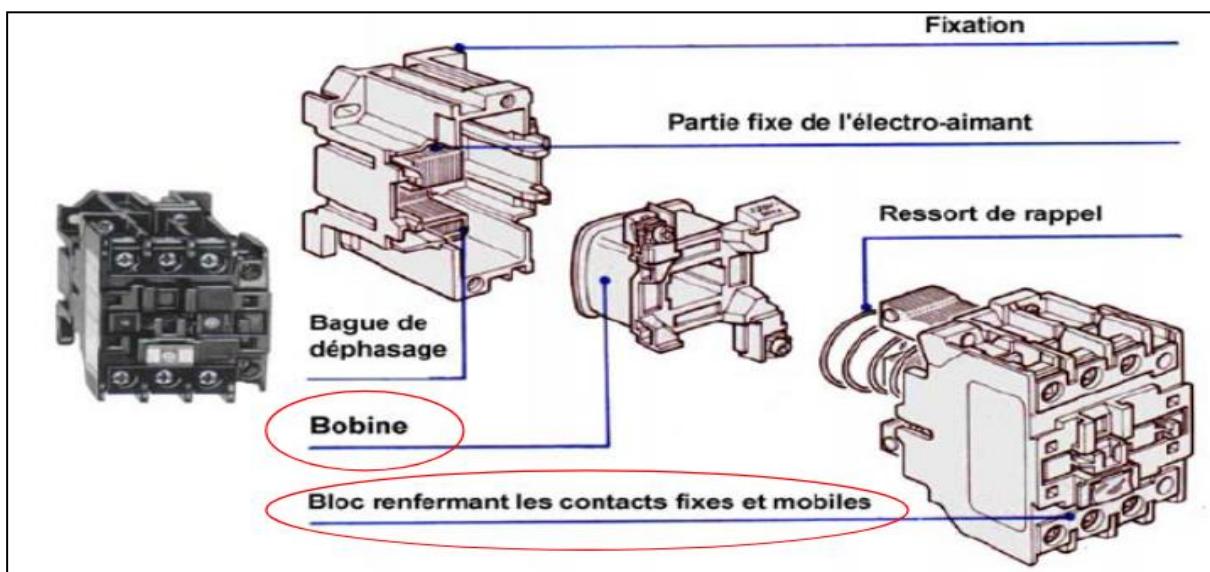


Figure II.6. Schéma d'un Relais.

### **II.3.3. Contacteur**

Un contacteur est un appareil électrique destiné à établir un passage du courant à partir d'une commande électrique. Les contacteurs sont utilisés afin d'alimenter les moteurs industriels de grande puissance.



**Figure II.7. Schéma illustre les composantes d'un contacteur**

### **II.4. Les actionneurs**

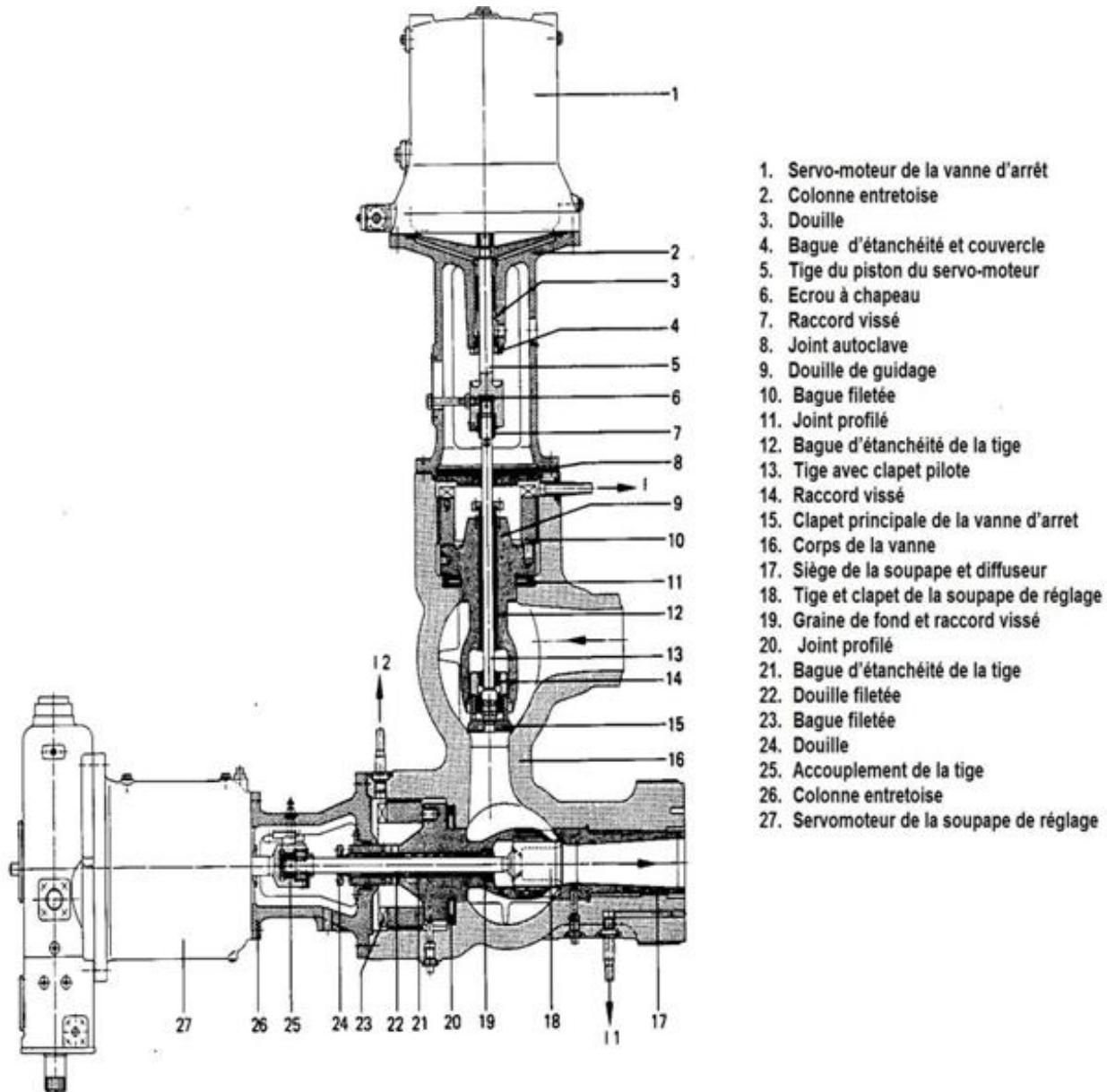
Dans un système automatique, un actionneur est un organe de la partie opérative qui sur ordre de la partie commande via le pré-actionneur, convertit l'énergie qui lui est fournie sous une forme utile pour les tâches programmées d'un système automatisé.

#### **II.4.1. Les vanne**

Une vanne est un dispositif de réglage commandé manuellement par un volant ou à distance par un signal électrique ou pneumatique. Parmi les différents types de vannes, on peut citer :

➤ **Vanne d'arrêt SA11S001 et soupape de réglage (figure II.8) :**

La vapeur pénètre dans le corps de la vanne (16) par la turbine d'admission et arrive au-dessus du clapet principal monobloc avec la tige de la vanne (13). Des bagues (12) assurent l'étanchéité de la tige au passage à travers la douille de guidage (9) (voir figure II.8).



**Figure II.8. Vanne d'arrêt et soupape de réglage [5]**

De plus, lorsque la vanne est ouverte, le clapet pilote vient appuyer sous l'action de la vapeur, avec son portage d'étanchéité arrière contre la vis (14) qui se trouve repoussée à son tour contre le grain de fond. Ce système complète l'étanchéité de la traversée de la tige. La tige et le clapet sont dotés d'une protection anti-torsion. La douille (9) est fixée au corps (16) par une filetée (10). L'étanchéité entre la douille et le corps est assurée par une bague élastique à profil (11) dont les branches s'écartent sous l'effet de la pression de la vapeur et, viennent appuyer contre les surfaces d'étanchéité du corps.

## CHAPITRE II : L'instrumentation et moyens d'automatisation

Le clapet (18) et la tige de manœuvre de la soupape de réglage forment un ensemble monobloc. Des orifices de décharge sont percés. Dans le clapet pour diminuer l'effort d'ouverture de la soupape. Le clapet et sa tige (18) sont guidés par la douille, des bagues (21) assurent l'étanchéité de la tige au passage à travers la douille.

Lorsque la soupape est complètement ouverte, le portage à d'étanchéité arrière du clapet appuie contre le gain de fond (19). Ce système complète l'étanchéité de la traversée de la tige, comme sur la vanne d'arrêt, la douille est fixée au corps (16) par une bague filetée (23) et l'étanchéité entre la douille et le corps est assurée par une bague filetée (23) et l'étanchéité entre la douille et le corps est assurée par une bague filetée (23). Le piston de servomoteur (27) qui est actionné par un ressort Belleville à la fermeture et par l'huile motrice à l'ouverture. Le déclencheur de la fermeture rapide ou un incendie sur le circuit de régulation provoque la fermeture brusque de la vanne d'arrêt et de la soupape de réglage. [5].

### II.4.2. L'électrovanne

L'électrovanne est un robinet électromécanique, qui est contrôlé par un courant électrique. Le courant électrique traverse un solénoïde, qui est une bobine de fil enroulée autour d'un noyau métallique. Le solénoïde crée un champ magnétique contrôlé au moment où un courant électrique passe par elle. Ce champ magnétique affecte l'état de l'électrovanne, causant l'ouverture ou la fermeture de la valve. Les électrovannes sont utilisées pour le transport de gaz ou de liquides. Elles ont une grande variété d'applications, notamment l'irrigation des systèmes de gicleurs et des usages industriels.

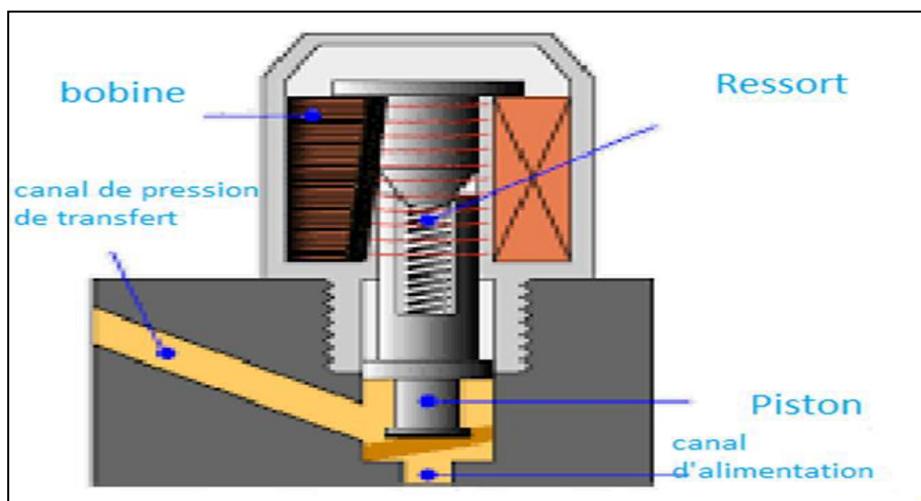


Figure II.9. Schéma d'un électrovanne

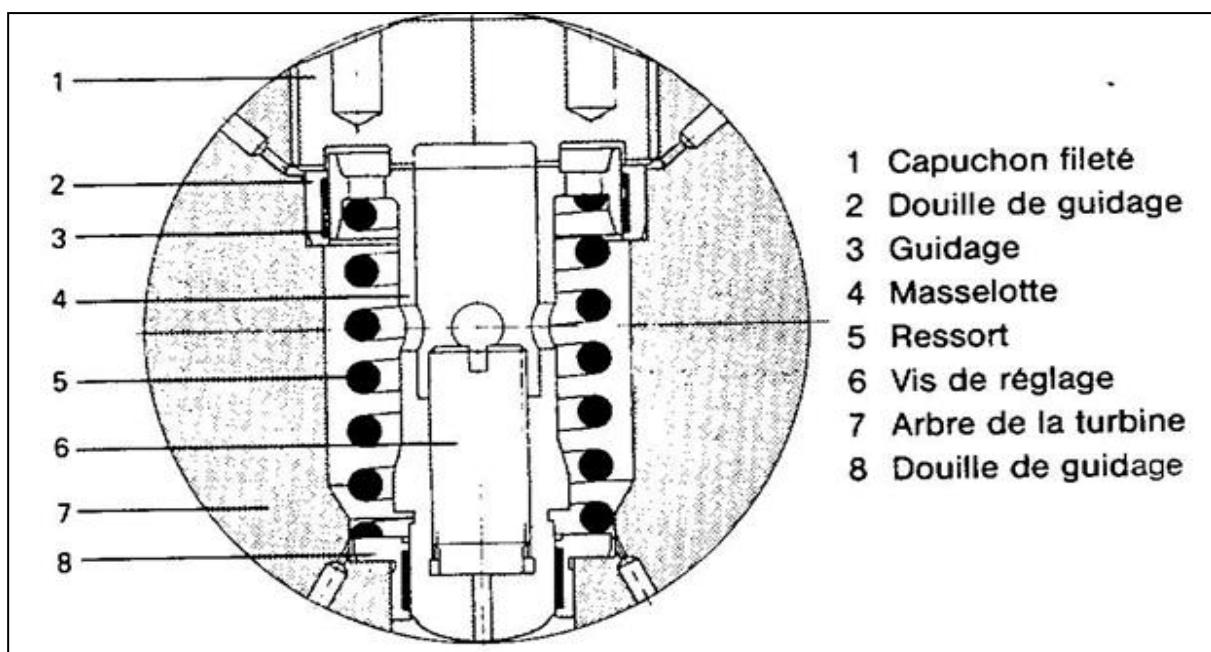
### **II.4.3. Les convertisseurs**

L'indication à distance de la vitesse de rotation est assurée par un convertisseur de mesure électrique de vitesse de rotation, monté sur l'arbre de la pompe d'huile principale. Le convertisseur se compose d'un disque (en aluminium) comportant 60 paires d'aimants montés dans des orifices répartis au périphérique du disque. Etant donné que ce disque magnétique est solidaire de l'arbre de la pompe d'huile principale qui elle est même solidaire à l'arbre de la turbine, il tourne à la vitesse de rotation de la turbine.

### **II.4.4. Les détecteurs**

#### **II.4.4.1. Détecteur de survitesse**

Le détecteur de survitesse a pour fonction d'arrêter la turbine en cas de dépassement de la vitesse admissible, il est monté dans le rotor de la turbine (figure II.10).



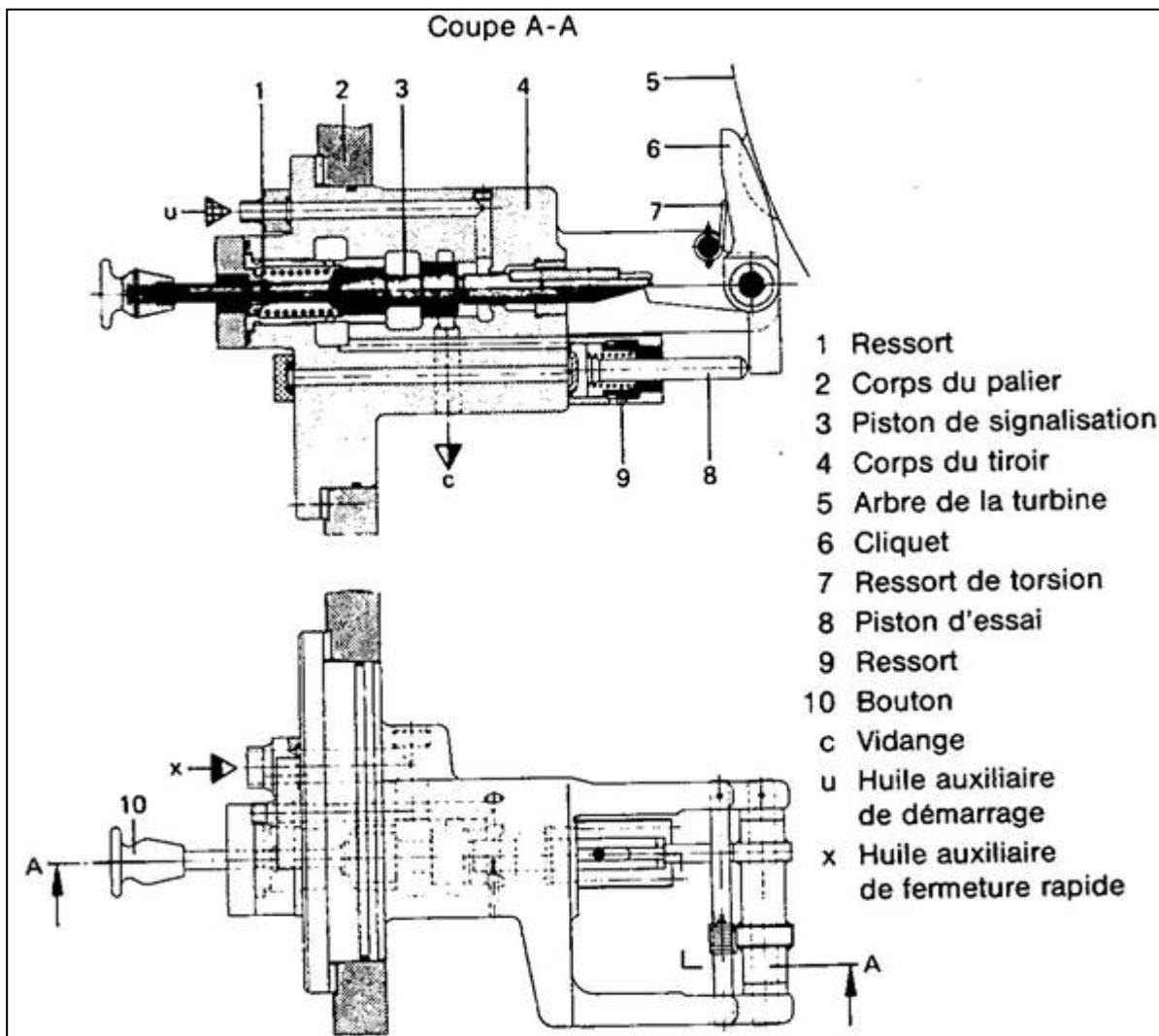
**Figure II.10. Détecteur de survitesse [5].**

La vitesse de déplacement est réglée au moyen de la vis (6). Cette vis de réglage sert à déplacer le centre de gravité de la masselotte (4) et à la centrer par rapport à l'arbre de la turbine de façon que la force de rappel du ressort soit supérieure à la force centrifuge sollicitant masselottes ; aussi longtemps que la vitesse de la turbine est inférieure à la vitesse de déclenchement .

Dans cette position, la masselotte appuie contre le capuchon fileté (1). Si la vitesse de déclenchement est atteinte, la force centrifuge l'emporte sur la force de rapport de ressort (5) et la masselotte (4) est poussée hors de l'arbre de la turbine, ce qui déclenche la fermeture rapide (voir figure II.10).

#### **II.4.4.2. Détecteur d'usure de la butée SB12K011**

En cas de déclenchement axial du rotor par rapport à une force d'usure des patins de butée, le dispositif de déclenchement ouvre le circuit d'huile auxiliaire de sécurité qui provoque l'arrêt de la turbine (voir Figure II.11)



**Figure II.11. Détecteur d'usure de la butée [5]**

### ➤ Principe de fonctionnement :

En cas de déplacement axial du rotor (5), le cliquet (6) enclenché entre deux cames du rotor vient chevauche l'une des cames. En tournant sur son axe, il libère le tiroir (3) qui se dirige vers le rotor (5) sous l'action de ressort (1) et de la pression différentielle de l'huile. Le circuit d'huile auxiliaire de sécurité X est alors mis en communication avec la vidange C par les orifices ménagés dans le corps du tiroir (4).

La chute de pression qui en résultant provoque la fermeture rapide de la turbine.

### II.4.4.3. DéTECTEUR de vide au condenseur

Le détecteur de vide déclenche la fermeture rapide de la turbine lorsque la pression au condenseur dépasse la valeur admissible.

L'espace vapeur du condenseur, communique par le raccord de la chambre de manostat située au-dessus de la membrane. Le vide est situé au-dessous de la membrane est la pression atmosphérique. Si la pression au condenseur dépasse la limite admissible, la membrane et le tiroir sont pilotés vers le bas, par suite de l'augmentation de pression et sous l'action d'un ressort. L'huile auxiliaire de sécurité est alors mise à la vidange, ce qui est résulté un déclenchement rapide de la turbine.

## II.5. Conclusion

L'instrumentation des chaînes de mesure et de contrôle de la turbine assure la protection des éléments les plus sensibles de la centrale. La qualité des informations récoltées par ces instruments augmentent le taux de sécurité de l'installation.

Dans le chapitre suivant, nous nous intéresserons à l'automatisation et supervision du système de commande de vitesse d'une turbine à gaz.

## **CHAPITRE III**

Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz.

---

### **III.1. Introduction**

La turbine à gaz de type SIEMENS implantée au niveau de la centrale RAS-DJENET est munie de plusieurs systèmes de commande et de protection prévus pour assurer la fiabilité et la sécurité de fonctionnement de la machine. La conduite de la turbine est effectuée principalement par la commande de lancement, la commande de vitesse et la commande de température. Les systèmes de protection de survitesse et de sous vitesse sont conçus pour protéger la turbine à gaz contre tout dommage éventuel causé par une vitesse insuffisante ou excessive d'un des arbres de la turbine. En fonctionnement normale, la vitesse de ces arbres est contrôlée par la boucle de vitesse ou la boucle de température.

Ce chapitre est divisé essentiellement en deux parties. Dans la première partie on va s'intéresser à l'automatisation de système de commande de vitesse de la turbine à gaz. La deuxième partie sera consacrée au développement de la plateforme de supervision.

Avant d'entamer l'automatisation de notre système proposé, il est intéressant de donner quelques généralités sur l'Automate programmable (L'API).

### **III.2. Généralité sur l'automate programmable industriel (API)**

#### **III.2. 1. Définition d'un automate programmable**

L'automate programmable industriel API (Programmable logic Controller PLC) est un appareil électronique programmable destiné, à l'aide d'un langage adapté, pour piloter en ambiance industrielle et en temps réel des procédés ou des systèmes automatisés.

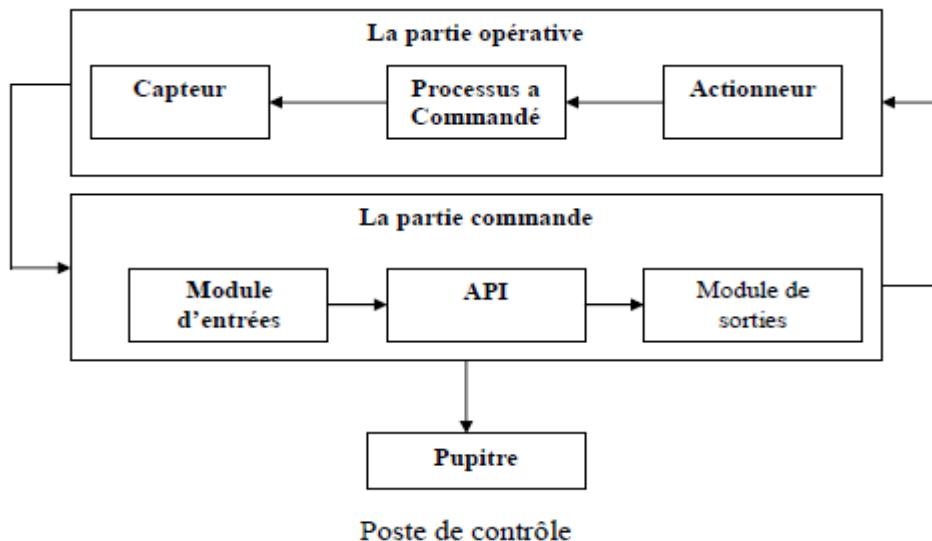
En fonction du programme chargé en mémoire, l'automate reçoit des informations logiques, analogiques ou numériques à partir des dispositifs d'entrées et ordonne des données pour la commande des dispositifs de sortie. Le programme de fonctionnement est effectué à partir d'instructions établies en fonction du processus à réaliser [7].

Les systèmes automatisés utilisés dans le secteur industriel possèdent une structure de base identique, ils sont constitués de plusieurs parties qui sont reliées entre elles, à savoir :

- **La partie opérative (PO) :** C'est la partie qui assure la fonction globale de transformation de la matière d'œuvre. La partie opérative comprend (actionneurs, prés-actionneurs, détecteurs).

## CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

- **La partie commande (PC)** : elle élabore les ordres nécessaires à l'exécution d'un processus et reçoit en retour des comptes rendus qui informent sur l'état des opérations effectuées.
- **partie relation (PR)** : ou pupitre de dialogue, c'est un appareil de contrôle qui permet une intervention rapide sur la machine, par exemple : la lecture des données machines.



**Figure III.1. Structures générales d'un système automatisé.**

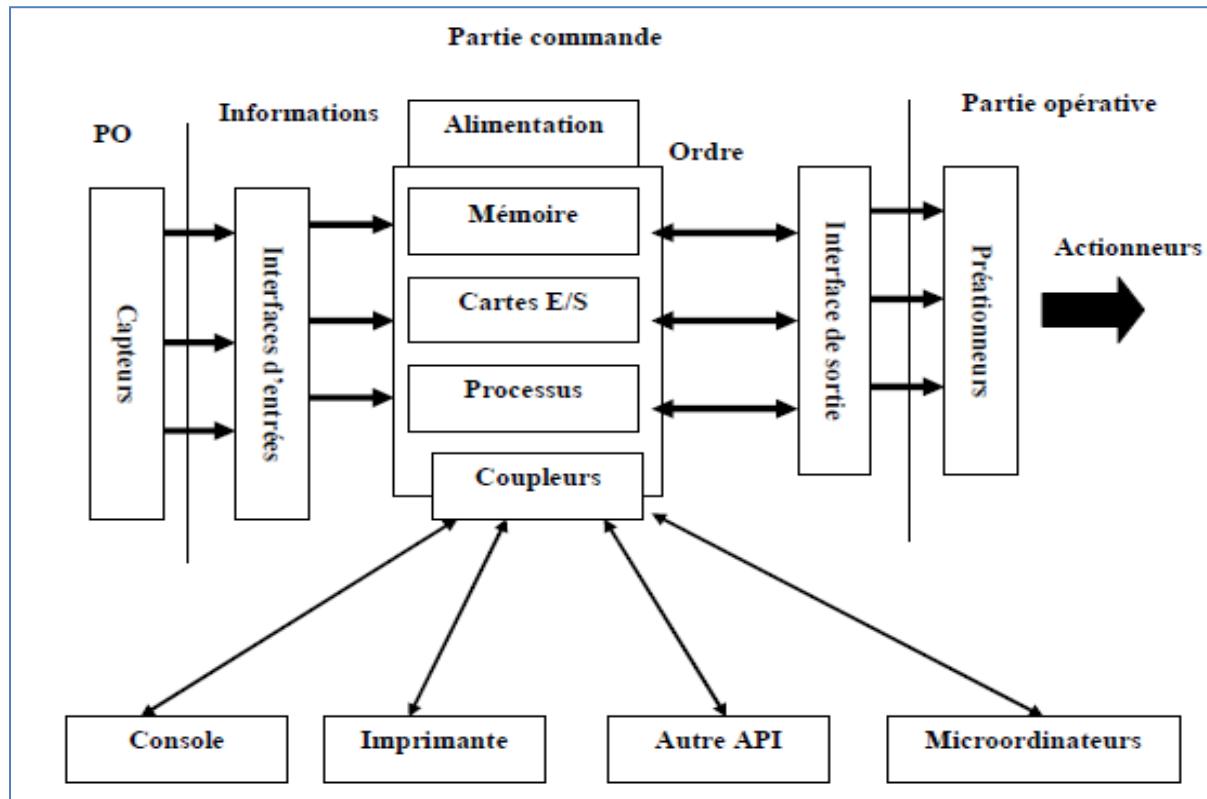
### III.2.2. Architecture d'un API

#### III.2.2.1. Architecture interne d'un API

Cette structure comporte les parties principales suivantes :

- ✓ L'unité centrale ;
- ✓ Les modules d'entrées /sorties ;
- ✓ Le module d'alimentation ;
- ✓ Les modules de communications ;
- ✓ Les coupleurs.

Ces parties sont reliées entre elles par des bus (ensembles des fils autorisant le passage des informations entre ces secteurs de l'automate). Donc on peut représenter ces parties dans le schéma synoptique suivant :



**Figure III.2. Architecture d'un automate programmable industriel**

### III.2.2.2. Architecture externe d'un API

On distingue deux types d'architectures externes :

- ✓ L'architecture modulaire
- ✓ l'architecture compacte (non modulaire).

Dans les API compacts, tous ses composants (CPU, modules d'entrée et sortie...), sont intégrés dans un seul boîtier.

Dans les API modulaires, chaque élément est indépendant de l'autre (CPU, modules d'entrées et sorties, alimentation...) [10].

### III.2.3. Programmation des API

Les programmes utilisés avec les API peuvent être écrits dans différents formats. Au début, chaque fabricant d'automates a eu tendance à développer ses propres méthodes de programmation pour les API.

## **CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz**

---

Une norme fut créée en 1993 par la commission électrotechnique internationale, désignée sous la référence **CEI 61131-3** qui est adapté à l'environnement industriel, et réalise des fonctions d'automatisme pour assurer la commande de prés actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques, par exemple, on cite les automates SIMATIC S7 [8].

La gamme SIMATIC S7 comprend les systèmes d'automatisation suivants :

- ✓ S7-200 : Un micro-automate compact de l'entrée de gamme.
- ✓ S7-300 : Un micro-automate modulaire de milieu de gamme.
- ✓ S7-400 : Il couvre le haut et très haut de gamme.

Les critères de choix d'un API sont [10] :

- Le nombre et la nature des entrée/sorties.
- La nature de traitement (temporisation, comptage..).
- Les capacités de traitement du processus (vitesse, taille du programme, opérations, temps réel,...).
- Le dialogue (la console détermine le langage de programmation).
- Les moyens de sauvegarde du programme (disquettes, cassettes..).
- La fiabilité et la robustesse et l'immunité aux parasites.

### **III.2.4. Présentation de l'automate S7-300**

L'automate programmable industriel SIMATIC S7-300 est un appareil électronique destiné à la commande et à la surveillance en temps réel des processus industriels. Le S7-300 est un automate modulaire de moyenne gamme SIMATIC S7 fabriqué par la firme SIEMENS.

Ces modules sont simplement accrochés sur un profilé-support et vissés pour former un ensemble robuste.

La riche gamme de modules de l'automate S7-300 permet la réalisation d'extensions centralisées et de structures décentralisées [9].

### CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

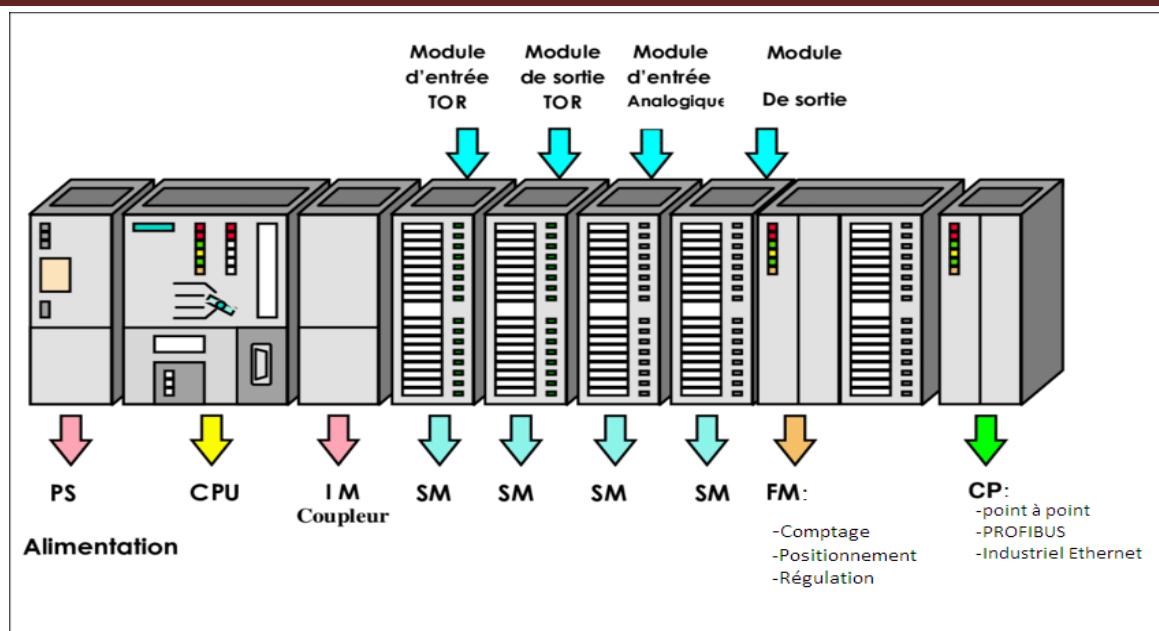


Figure III.3. Schéma extérieur d'un API s7-300.

**CPU:** unité centrale. ; **PS:** module d'alimentation. ; **IM:** coupleur.

**CP:** processeur de communication ;

**FM:** module de fonctionnement il comporte : le comptage ; la régulation ; la positionnement

**SM:** module de signaux.

#### III.2.5. Programmation de l'automate S7-300

Les automates programmables industriels effectuent des tâches d'automatisation traduites sous formes de programme d'application qui définit la manière dont l'automate doit commander le système par une suite d'instructions. Le programme doit être écrit dans un langage déterminé avec des règles définies pour que l'automate puisse l'exécuter. Pour cela, les automates de la famille Siemens sont programmés grâce au logiciel STEP7 via une console de programmation ou PC et sous un environnement Windows. Le logiciel STEP7 offre les fonctions suivantes [9] :

- Configuration et paramétrage du matériel et de la communication.
- La création et la gestion des projets.
- La gestion des mnémoniques.
- La création des programmes pour des systèmes cibles S7.
- Le chargement des programmes dans les systèmes cibles.
- Test de l'installation d'automatisation.

## **CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz**

---

Le STEP7 dispose de trois langages de programmation ainsi que d'une méthode utilisant le GRAFCET comme outil :

- ✓ ***Langage liste (LIST)***: image textuelle proche du comportement interne de l'automate.
- ✓ ***Langage logigramme (LOG)***: langage graphique, utilisant les symboles de l'électronique numérique (porte logique)
- ✓ ***Langage contact (CONT)***: suite de réseaux parcourus séquentiellement dont les entrées sont représentées par des interrupteurs et les sorties par des bobines. Il s'adapte à la plupart des automates existants. Il permet une programmation aisée des systèmes séquentiels tout en facilitant la mise au point des programmes ainsi que le dépannage des systèmes. On peut également l'installer sur tout type d'automate. C'est pour cela, nous l'avons choisi pour notre projet.

### **III.2.6. Les différents blocs de S7**

Dans le STEP7, il existe plusieurs blocs utilisateurs pour un programme structuré :

- **OB (bloc d'organisation)** : 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.[5]
- **FB (bloc fonctionnel)** : 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 à un FB. Il est possible d'appeler d'autres FB et FC dans un bloc fonctionnel via des instructions d'appels de blocs.
- **FC (fonction)** : Une FC ne possède pas une zone de mémoire propre. Les données locales d'une fonction sont perdues après l'exécution de la fonction. Il est également possible d'appeler d'autres FB et FC dans une fonction via des instructions d'appels de blocs.
- **DB (bloc de données)** : 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.

## **CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz**

---

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.

Le STEP 7 dispose de blocs nommés **bloc-système**. Ces blocs peuvent être appelés par l'utilisateur et utilisés dans le programme, mais on ne peut pas changer ni accéder à leurs programmes.

- **SFB (bloc fonctionnel système)** : Bloc fonctionnel stocké dans le système d'exploitation de la CPU et pouvant être appelé par l'utilisateur.
- **SFC (fonction système)** : Fonction stockée dans le système d'exploitation de la CPU et pouvant être appelée par l'utilisateur.
- **SDB (données système)** : Zone de mémoire dans le programme configurée par différentes applications de STEP 7 (par exemple S7 Configuration, Communication...), pour le stockage des données dans les systèmes d'automatisation.

### **III.3. Cadre pratique : Programmation et supervision**

L'objectif de notre projet est de remplacer le système de la logique câblée de la commande de vitesse d'une turbine à gaz qui existe dans la centrale de Ras-Djanet par un système automatisé en utilisant un automate programmable industriel SIEMENS S7-300.

Pour ce faire, nous commençons en premier lieu par présenter le logigramme fonctionnel séquentiel de la turbine à gaz. Ensuite, nous réalisons le programme correspondant au logigramme et tester notre application par simulation en utilisant le logiciel STEP 7. A la fin de ce chapitre, nous visualiserons le fonctionnement de notre système par le logiciel SIMATIC WIN CC flexible.

#### **III.3.1. Le logigramme fonctionnel séquentiel du système de commande de vitesse (cahier de charge)**

Dans cette partie, nous avons modélisé le fonctionnement du système de commande de vitesse de la turbine à gaz par le logigramme fonctionnel séquentiel pour réaliser des programmes en utilisant un SIMATIC STEP 7. Ces programmes sont implantés sur L'API.

Les figures ci-dessous montrent le fonctionnement de ce système [6].

### CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

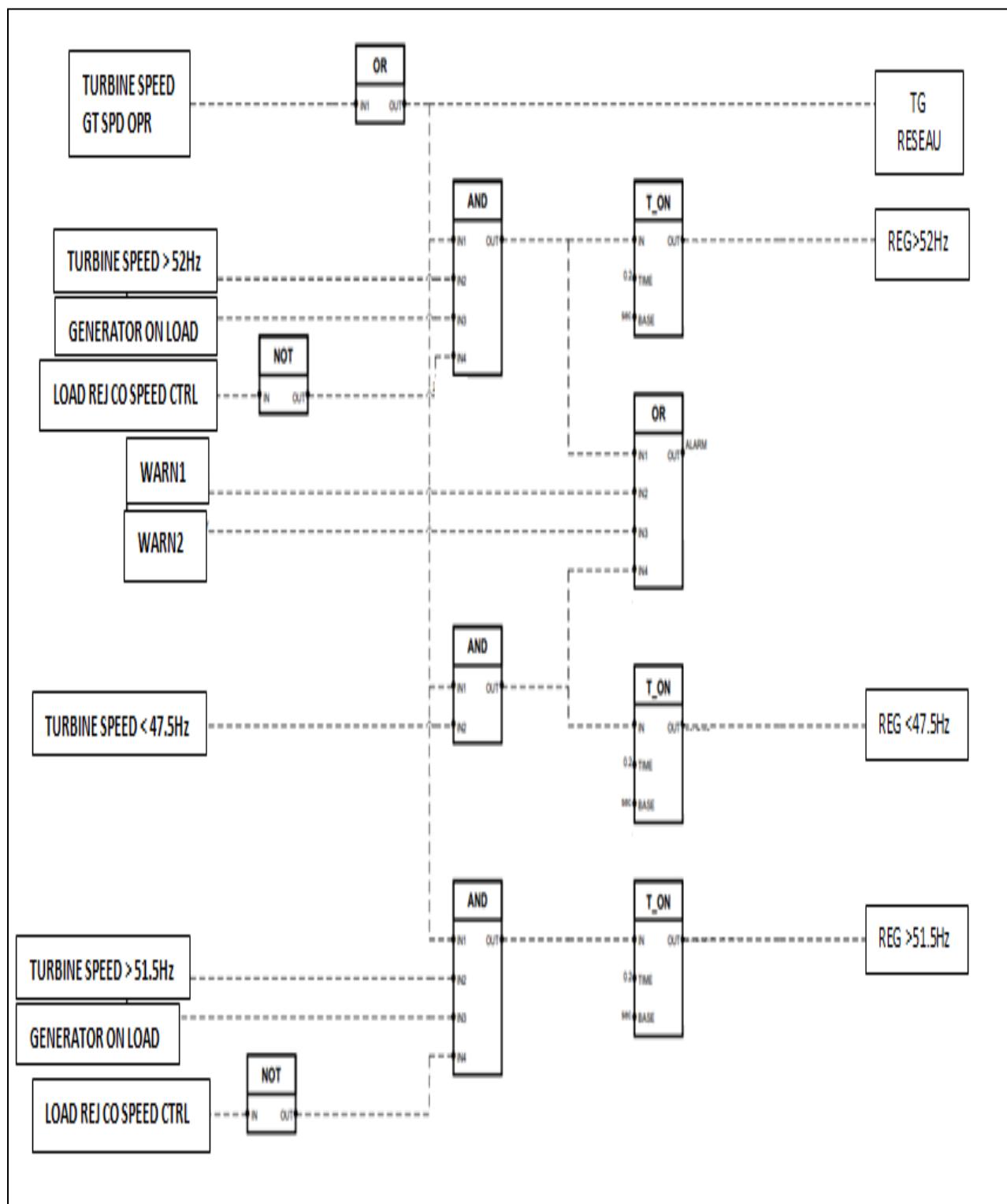


Figure III.4. Le logigramme fonctionnel séquentiel du système de commande de vitesse.

### CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

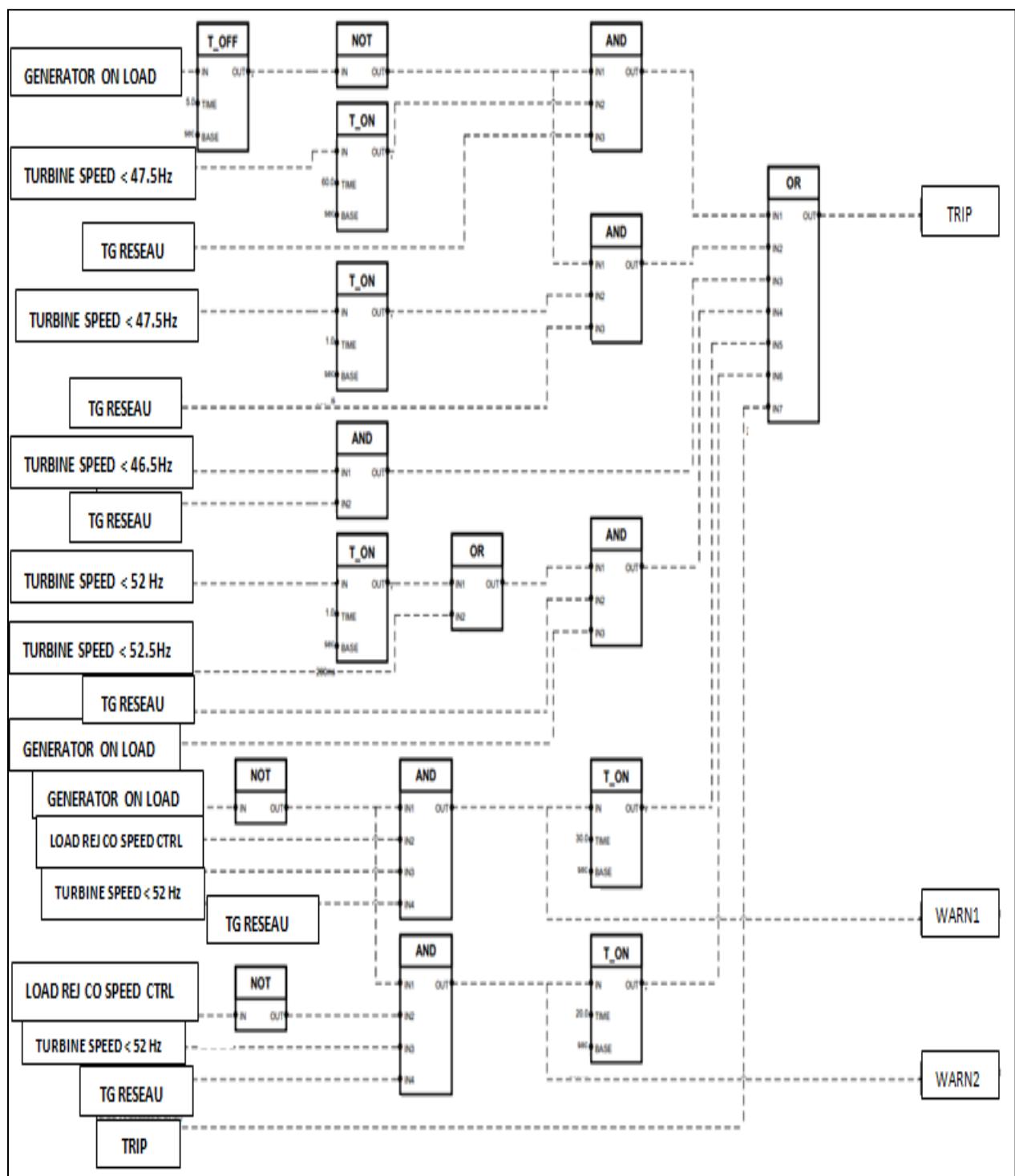


Figure III.5. Logigramme fonctionnel séquentiel (suite 1)

### CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

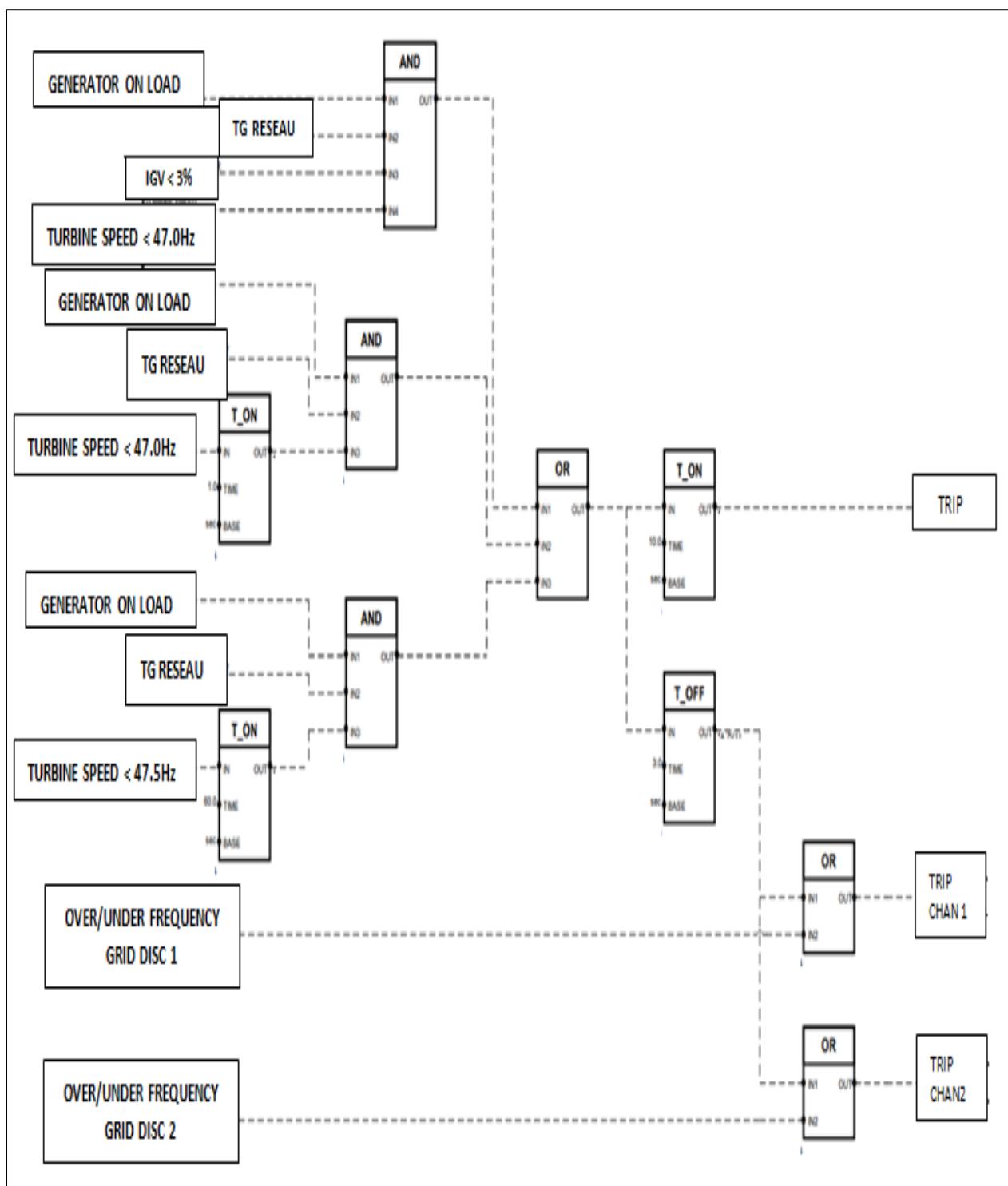


Figure III.6. Logigramme fonctionnel séquentiel (suite 2)

### **CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz**

---

#### **➤ Principe de fonctionnement de logigramme avec les paramètres de protection de fréquence (ANSI 81) :**

Cinq niveaux de fréquence de haute précision sont disponibles dans notre système de commande de vitesse. Chaque étage peut être défini comme étage de surfréquence (excès de vitesse) ou étage de sous-fréquence (vitesse insuffisante). Celle-ci est déterminée par la fréquence nominale configurée sous l'adresse 0270 et la valeur limite réglée.

Lorsque la valeur limite est inférieure à la fréquence nominale, le seuil fonctionne comme seuil de sous-fréquence. Lorsque la valeur limite est fixée égale à la fréquence nominale, l'étage concerné est inactif. Pour l'étage de fréquence f4, les circonstances expliquées précédemment sont également pertinentes si l'adresse 4214 est réglée sur automatique. En principe, les valeurs de réglage de la fréquence et des temps de retard dépendent des valeurs fournies par l'exploitant de la centrale et des exigences du fabricant de la turbine et du générateur.

Les paramètres suivants ont été fournis par le fabricant pour les turbines à gaz de 50 Hz.

**Tableau III.1. Les paramètres de protection de fréquence de la turbine.**

<b>Adresse</b>	<b>Paramètres / [Commentaire]</b>	<b>Étape de réglage 2</b>	<b>Étape de réglage</b>
4199	Protection de sur/sous fréquence	ON	
4200	Récupérer f0	46,50 Hz	
4201	Temporisation T f0	00,00 sec	
4202	Récupérer f1	47,00 Hz	47,00 Hz
4204	Temporisation T f1	26,00 sec	20,00 sec
4205	Récupérer F2	47,50 Hz	47,50 Hz
4207	Temporisation T f2	30,00 sec	20,00 sec
4208	Récupérer F3	52,00 Hz	
4210	Temporisation T f3	20,00 sec	
4211	Récupérer f4	52,50 Hz	
4213	Temporisation T f4	0,00 sec	
4214	Traitements de l'étape de seuil f4	Fréq. prot. stade automatique	
4215 ;	Tension minimale requise pour le fonctionnement [Si la tension est inférieure à cette valeur, la protection de fréquence est bloquée. Réglage 65% de l'ONU]	65,0 V	

## **CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz**

Noter que Étape 1 exprime « déconnexion du réseau » et Etape 2 exprime « déclenchement de la turbine »

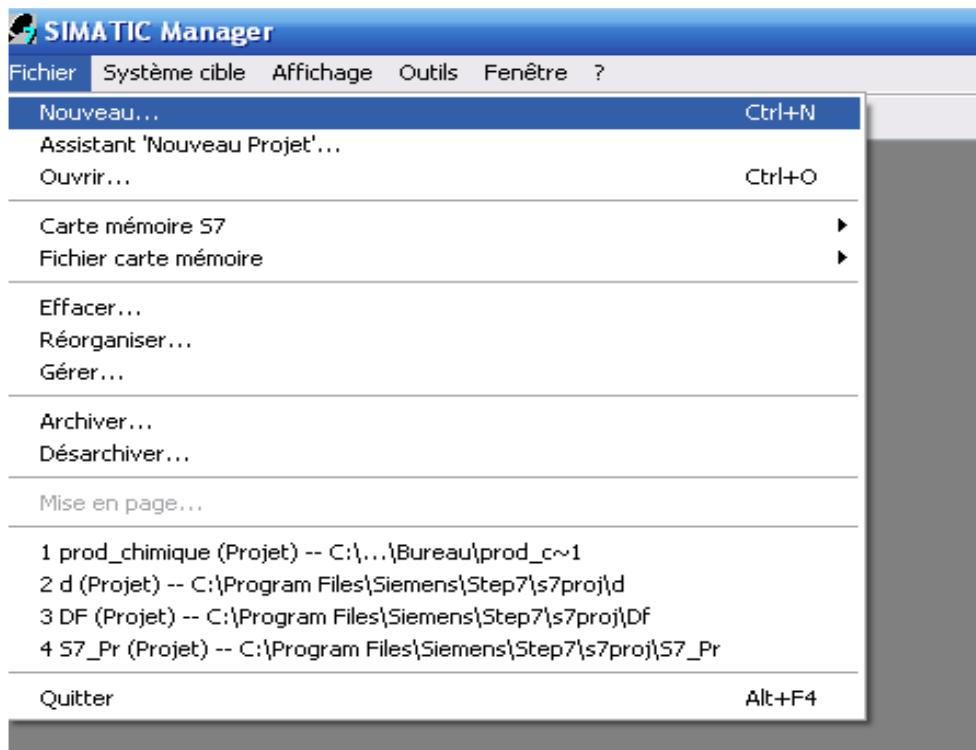
L'étage de fréquence f1 a une temporisation pour l'étage 1 de 20s et pour l'étage 2 de 26s.  
L'étage de fréquence f2 a une temporisation pour l'étage 1 de 20s et pour l'étage 2 de 30s.

### **III.3.2. Réalisation du programme de commande de la vitesse d'une turbine à gaz sous STEP7**

#### **III.3.2.1. Démarrage de STEP7 et création d'un nouveau projet**

Le démarrage de STEP7 est réalisé en cliquant deux fois sur l'icône “SIMATIC Manager”, ce qui permet d'ouvrir sa fenêtre fonctionnelle.

La création d'un nouveau projet, est obtenue en cliquant sur l'icône « fichier» dans la barre de menu, alors on obtient la figure III.7 suivante :



**Figure III.7. Fenêtre pour un nouveau projet**

Après la sélection de « Fichier, nouveau », une fenêtre s'ouvre pour donner un nom à notre projet.

### **III.3.2.2. Configuration matérielle d'une station SIMATIC**

Une fois le projet est créé, l'étape suivante est la configuration matérielle qui est une étape très importante, elle permet de reproduire à l'identique le système utilisé (châssis (Rack), alimentation, CPU, modules d'entrées/sorties etc..). Pour notre système nous avons choisis une configuration matérielle présente en figure III.8 dans laquelle nous avons :

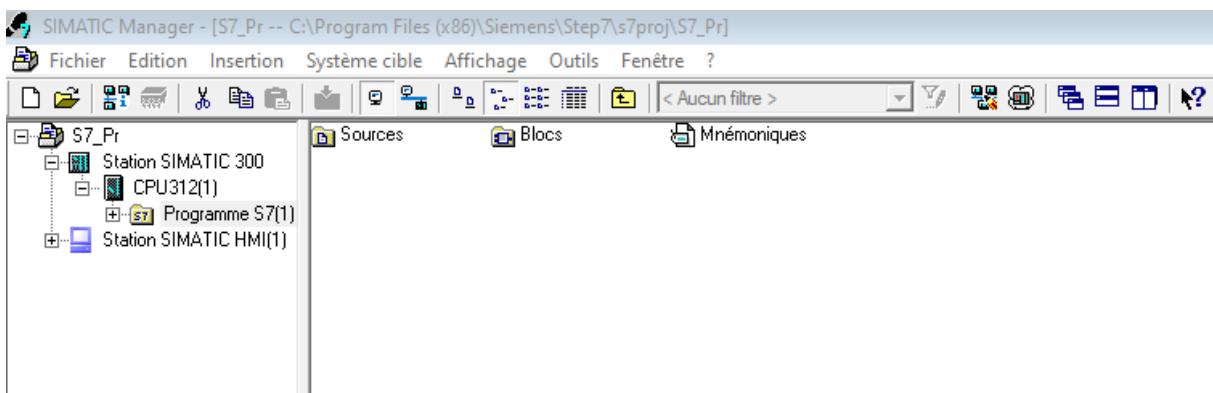
- Le module d'alimentation PS 307 10A ; la CPU 312.
- Un module d'entrée logique de 8 bits ; Un module de sortie logique de 8 bits.
- Un module d'entrée analogique de 4 bits.

Emplacement	Module	Référence	Firmware	Adresse MPI	Adresse d'entrée	Adresse de sortie	Commentaire
1	PS 307 10A	6ES7 307-1KA01-0AA0					
2	CPU312(1)	6ES7 312-1AE14-0AB0	V3.0	2			
3							
4	AI2x12Bit	6ES7 331-7KB00-0AB0			256...259		
5	DI8xAC230V	6ES7 321-1FF00-0AA0			0		
6	DO8xDC24V/0,5A	6ES7 322-8BF00-0AB0				5	
7							
8							
9							
10							
11							

**Figure III.8. Configuration matérielle.**

### **III.3.2.3 Table de mnémoniques**

Pour accéder à la table des mnémoniques (figure III.9), on clique sur « SIMATIC 300 », puis « CPU 312» et « Programme S7 ». Double cliquer sur l'icône « mnémoniques » qui apparaît dans la partie droite de l'écran.



**Figure III.9. Les blocs du programme**

### CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

---

En double clique sur mnémonique, pour lancer l'éditeur de mnémoniques. Après les avoir édité, la table des mnémoniques finales est illustrée dans la figure suivante.

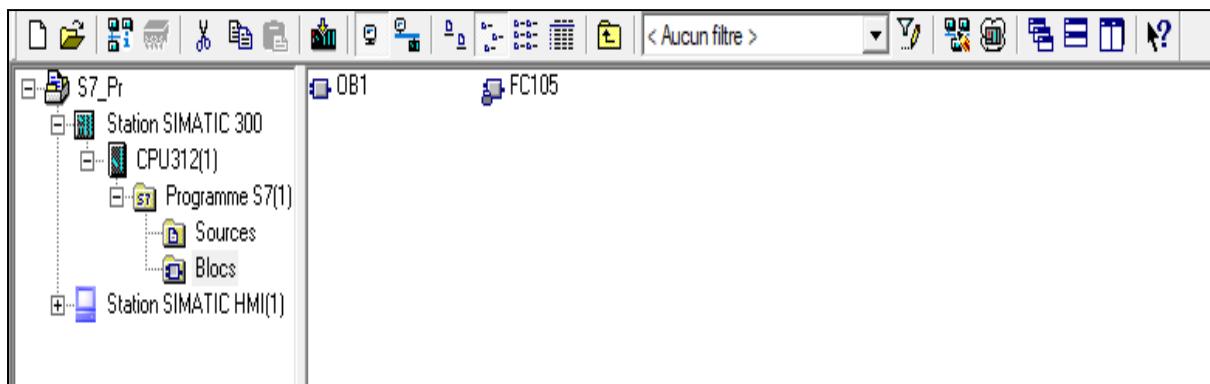
	Etat	Mnémonique	Opérant /	Type de do	Commentaire
1		REG SPD > 52Hz	A 5.0	BOOL	activation du regulateur de vitesse (Defaut survitesse > 52Hz)
2		REG SPD < 47.5Hz	A 5.1	BOOL	activation du regulateur de vitesse( Defaut sousvitesse < 47.5Hz)
3		REG SPD > 51.5Hz	A 5.2	BOOL	activation du regulateur de vitesse (Defaut survitesse > 51.5Hz )
4		START	E 0.0	BOOL	Début de cycle
5		over/under frq grd di...	E 0.1	BOOL	sur/sous frequence GRID DISC 2
6		generator on load	E 0.2	BOOL	generateur en charge
7		load rej spd ctr	E 0.3	BOOL	Activation du contrôleur de vitesse
8		over/under frqc grd ...	E 0.6	BOOL	sur/sous frequence GRID DISC
9		stop emergency	E 0.7	BOOL	" STOP " Arrêt d'urgence
10		TBL_TBL	FC 104	FC 104	Table to Table
11		SCALE	FC 105	FC 105	Scaling Values
12		A	M 0.0	BOOL	mémento
13		B	M 0.1	BOOL	mémento
14		C	M 0.2	BOOL	mémento
15		D	M 0.3	BOOL	mémento
16		E	M 0.4	BOOL	mémento
17		F	M 0.5	BOOL	mémento
18		G	M 0.6	BOOL	mémento
19		Pompe GAZ activé	M 0.7	BOOL	Pompe à gaz activé
20		Pompe GASOIL activé	M 2.0	BOOL	Pompe à gasoil activé
21		over/under frqc war...	M 2.1	BOOL	Avertissement de survitesse < 52 Hz (generateur non en charge,contrôleur activé)
22		over/under frqc war...	M 2.2	BOOL	Avertissement de survitesse< 52Hz (generateur non en charge,contrôleur désactivé)
23		Activation de la TG	M 2.3	BOOL	Activation de la turbine à gaz
24		TG réseau	M 2.6	BOOL	Turbine à gaz entre dans le réseau ( 50 Hz)
25		Trb spd = 50Hz	M 2.7	BOOL	vitesse turbine = 50.0 Hz
26		Trb spd < 47.0	M 3.0	BOOL	vitesse turbine < 47.0 Hz
27		Trb spd < 47.5	M 3.1	BOOL	vitesse turbine < 47.5 Hz
28		Trb spd > 51.5	M 3.2	BOOL	vitesse turbine > 51.5 Hz
29		Trb spd > 52.0	M 3.3	BOOL	vitesse turbine > 52.0 Hz
30		Trb spd > 52.5	M 3.4	BOOL	vitesse turbine > 52.5 Hz
31		Trb spd < 46.5	M 3.5	BOOL	vitesse turbine < 46.5 Hz
32		IGV Ouvert (> 0%)	M 3.6	BOOL	Ouverture des IGV > 0%
33		IGV < 3%	M 3.7	BOOL	Ouverture des IGV < 3%
34		STOP SUP	M 4.0	BOOL	Stop supervision
35		START SUP	M 4.1	BOOL	Demarrage supervision
36		over/under frqc chan 2	M 6.0	BOOL	Arrêt secondaire de la turbine à gaz ( CHAN2)
37		over/under frqc chan 1	M 6.1	BOOL	Arrêt secondaire de la turbine à gaz (CHAN1)
38		over/under frqc trip ...	M 6.2	BOOL	Arrêt de la turbine à gaz ( Defaut sous/sur vitesse)
39		capteur de vitesse a...	MD 12	REAL	Capteur de vitesse analogique
40		Capteur IGV analog	MD 20	REAL	Capteur analogique de contrôle des ouvertures des IGV
41		alarme_1	MW 10	WORD	Alarme défaut de vitesse <47.5Hz ou >52Hz( Activation du régulateur de vitesse)
42		alarme_2	MW 11	WORD	Alarme d'arrêt de la turbine à gaz
43		alarme_3	MW 12	WORD	Alarme de survitesse < 52Hz (generateur non en charge,contrôleur activé)
44		alarme_4	MW 13	WORD	Alarme de survitesse < 52Hz (generateur non en charge,contrôleur désactivé)
45		alarme_5	MW 19	WORD	Alarme Chan 1 ( TRIP)
46		alarme_6	MW 20	WORD	Alarme Chan 2 ( TRIP)
47		Cycle Execution	OB 1	OB 1	
48					

**Figure.III.10. La table des mnémoniques de notre programme**

## CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

Après avoir défini toutes les mnémoniques, il suffit d'enregistrer pour que les changements soient pris en compte dans le reste du projet.

Pour introduire le programme sur un A.P.I, nous avons choisi le langage de programmation à contact (CONT). Et pour la programmation des blocs : À l'aide de la commande «**insertion > bloc S7**» on crée des blocs ou dossier source.



**Figure.III.11. Les Blocs**

### III.3.2.4. Simulation et visualisation du programme

S7-PLCSIM est une application qui nous permet d'exécuter et de tester notre programme dans un automate programmable que nous simulons dans l'ordinateur (PC) ou dans une console de programmation, la simulation étant complètement réalisée au sein du logiciel STEP7. S7-PLCSIM dispose d'une interface simple nous permettant de visualiser et de forcer les différents paramètres utilisés par le programme, par exemple, activer ou désactiver des entrées.

Tout en exécutant notre programme dans la CPU simulée, nous avons également la possibilité de mettre en œuvre les diverses applications du logiciel STEP7, comme par exemple la table des variables (VAT) afin d'y visualiser et d'y forcer des variables.

### CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

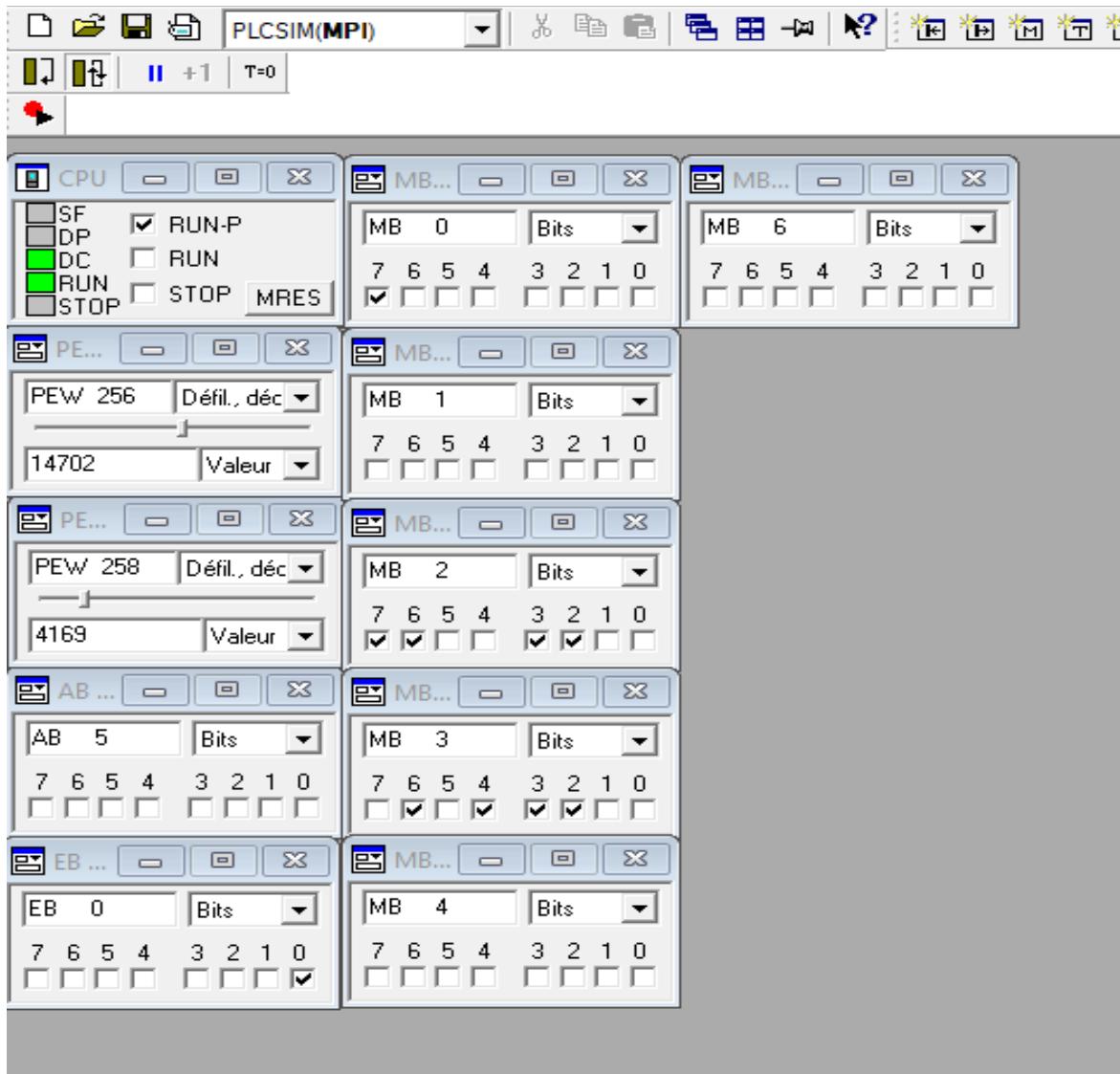


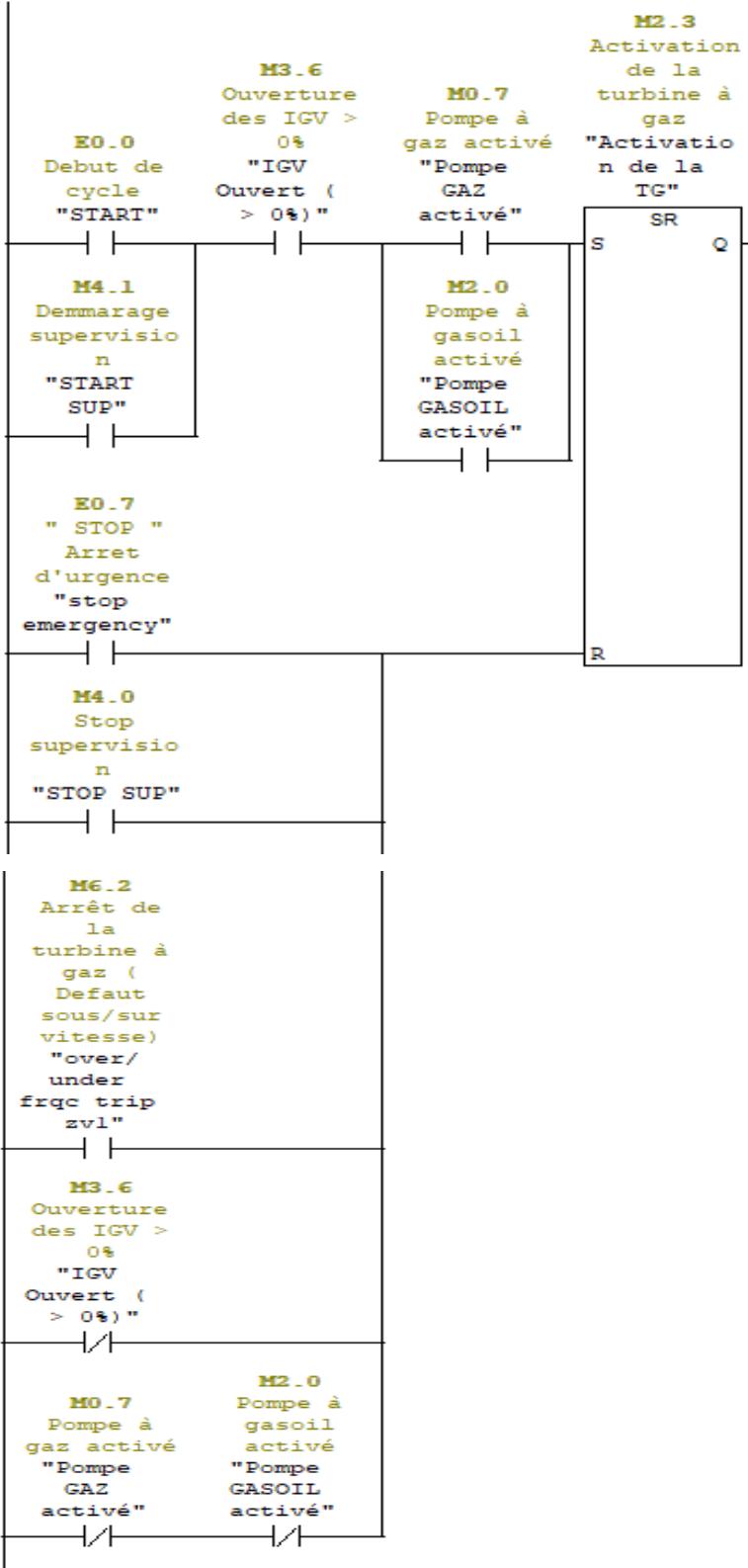
Figure III.12. Simulation S7-PLCSIM.

Après le chargement du programme de OB1 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. On présente ci-dessous quelques réseaux de notre programme (Figure III.13):

## CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

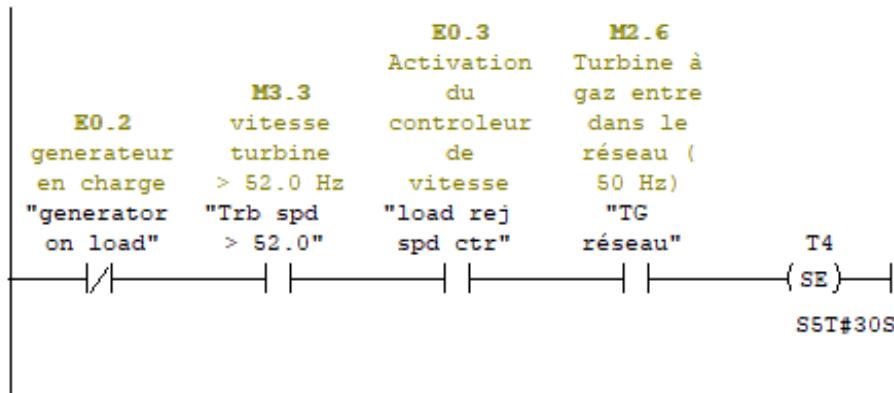
---

■ Réseau 1 : Réseau d'activation de la turbine à gaz

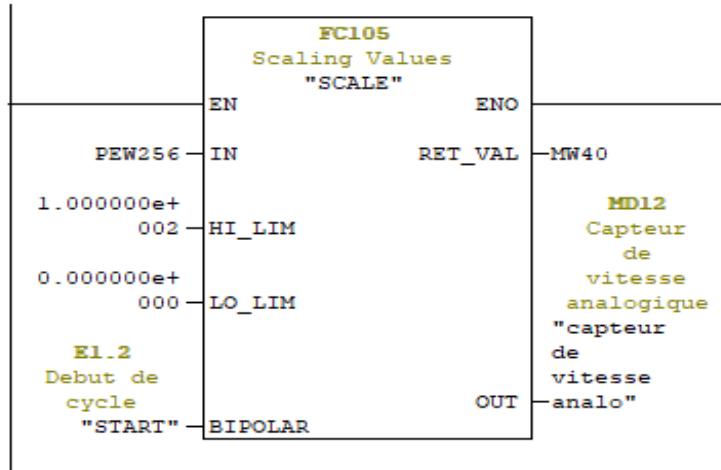


## CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

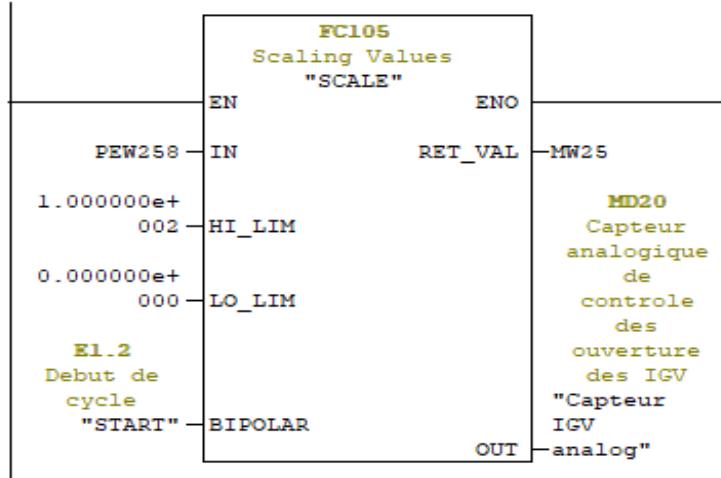
□ Réseau 11 : Réseaux d'activation de la temporisation T4



□ Réseau 32 : Capteur de vitesse analogique



□ Réseau 33 : Capteur analogique de contrôle d'ouverture des IGV analogique



## CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

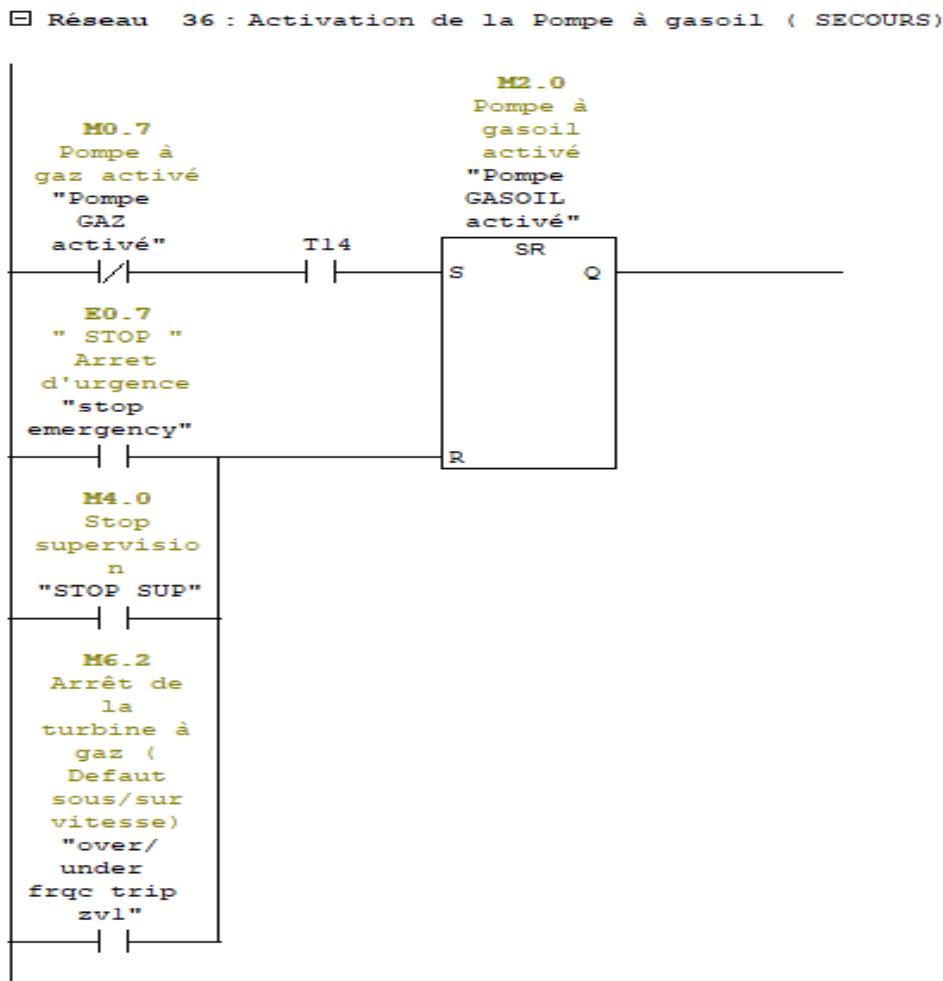


Figure III.13. Présentation de quelques réseaux de notre programme.

### III.4. Développement de la plateforme de supervision

La supervision est une forme évoluée de dialogue Homme-Machine, elle présente plusieurs avantages pour les processus industriels de production. Elle facilite à l'opérateur la Surveillance de l'état de fonctionnement d'un procédé ainsi que son contrôle-commande. Elle permet grâce à des vues créées, et configurées au préalable à l'aide d'un logiciel de supervision, d'intégrer et de visualiser en temps réel toutes les étapes nécessaires au processus. Elle permet aussi de détecter les problèmes qui peuvent survenir en cours de fonctionnement.

### **III.4.1. Présentation de logiciel WinCC flexible**

WinCC flexible est l'Interface Homme-Machine (IHM) pour les applications de la machine et du processus dans la construction d'installations, par des moyens d'ingénierie simples et efficaces, de concepts d'automatisation évolutifs, WinCC flexible est idéale grâce à sa simplicité, son ouverture et sa flexibilité [9].

➤ **Critères de choix de Win cc flexible :**

- ✓ Le panel de 'SIMATIC' ne peut pas fonctionner sans 'Win cc flexible'.
- ✓ Tendance de changer 'Protool' par 'Win cc flexible' pour 'SIEMENS'.
- ✓ Possibilité d'ouvrir d'autres panels de' Protool'.

➤ **Intégration de WinCC flexible dans le STEP7 :**

Une solution d'automatisation complète est composée d'une IHM telle que WinCC flexible, et d'un système d'automatisation, d'un bus système et d'une périphérie.

Pour intégrer le WinCC flexible dans un projet de STEP 7, on clic sur « Projet, intégrer dans le projet STEP 7 » puis on choisit le nom de projet dans la barre d'outils de WinCC flexible.

➤ **La mise en route de WinCC flexible :**

WinCC flexible met à disposition une boite d'outil qui contient les différents éléments pour la réalisation d'un projet, une bibliothèque, une zone de travail et d'une fenêtre de projet contenant l'ensemble des vues de projet.

### **III.4.2. Elaboration de la plateforme de supervision**

Le programme de supervision que nous avons développé a été élaboré avec le logiciel WinCC flixible (Windows Control Center), version 5.1 développé par SIEMENS.

Il est caractérisé par sa flexibilité c'est-à-dire qu'il peut être utilisé pour les composants hors SIEMENS. Il nous permet de visualiser le fonctionnement de la turbine et de contrôler la vitesse rotation.

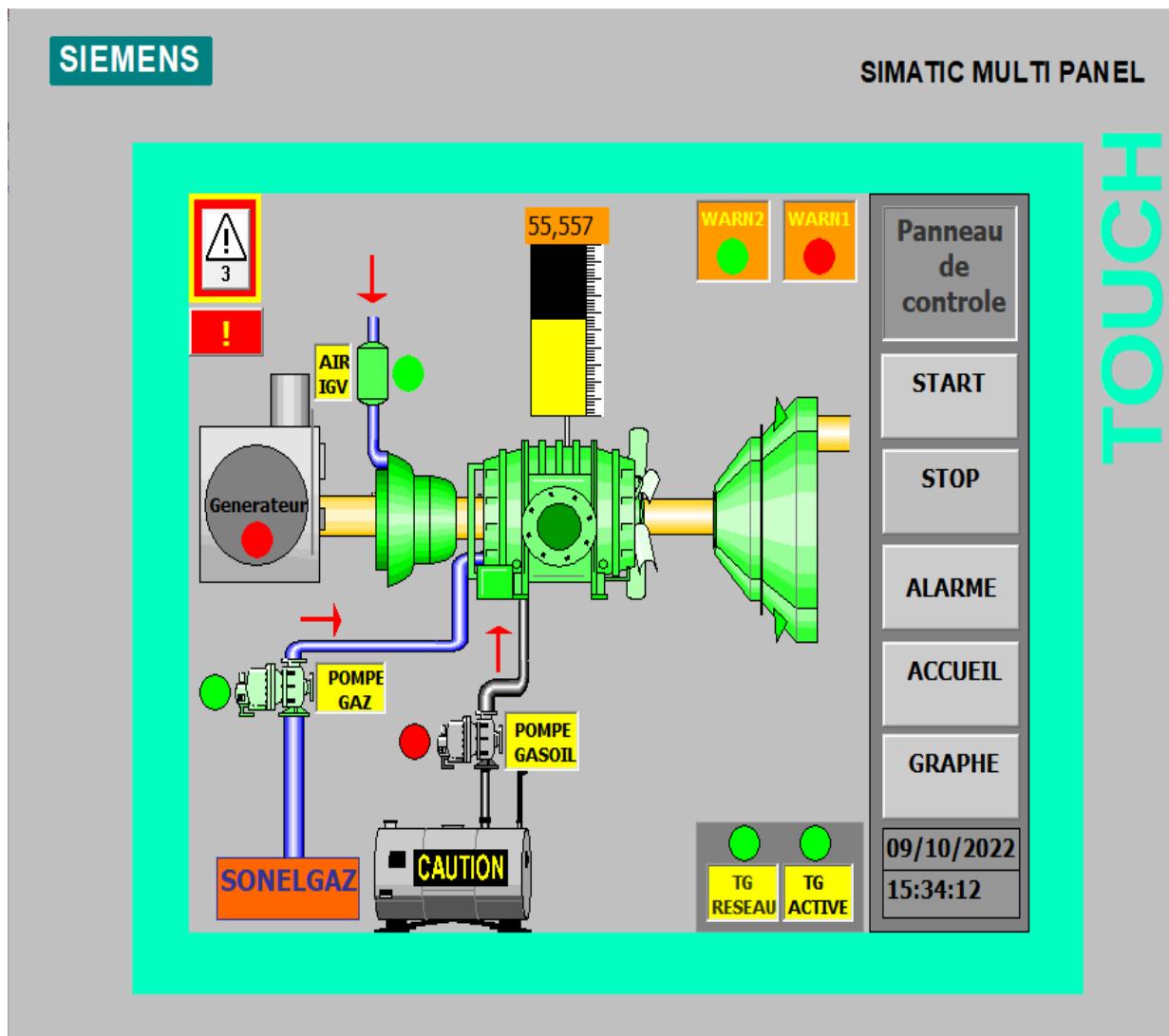
### **CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz**

Pour ce faire, nous avons programmés quatre vues de supervisions qui permettent à l'opérateur d'avoir toutes les informations nécessaire sur notre système de commande de vitesse de la turbine à gaz

Nous avons créé la vue d'accueil (figure III.14), qui contient le boutons d'entrer, ce bouton nous permettra d'afficher la vue principale (figure III.15) à partir de laquelle on peut sélectionner la vue à visualiser notamment : BLOC2 (figure III.16) et BLOC3 (figure III.17).



**Figure.III.14. Vue d'accueil**



**Figure III.15. Vue principale**

**La vue des ALARME** consiste à configurer les alarmes avec l'éditeur « Alarme TOR ». Ces alarmes sont utilisées pour alerter l'opérateur en signalant les défauts de vitesse ou de la température de la turbine ainsi que l'arrêt d'urgence. Ces alarmes indiquent aussi l'état des pompes « marche» ou « arrêt ».

En utilisant l'assistant de configuration on créera un système d'alarmes ; ensuite on sélectionnera dans la fenêtre de dialogue "Blocs d'alarme" les blocs systèmes "Date, Heure, Numéro" et les blocs utilisateur "Texte d'alarme". (Figure III.16).

Aussi dans la fenêtre de dialogue "Paramétrer classes", on sélectionnera "Classe dérangement avec alarme, erreur et avertissement (acquittement à l'arrivée)".

### CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

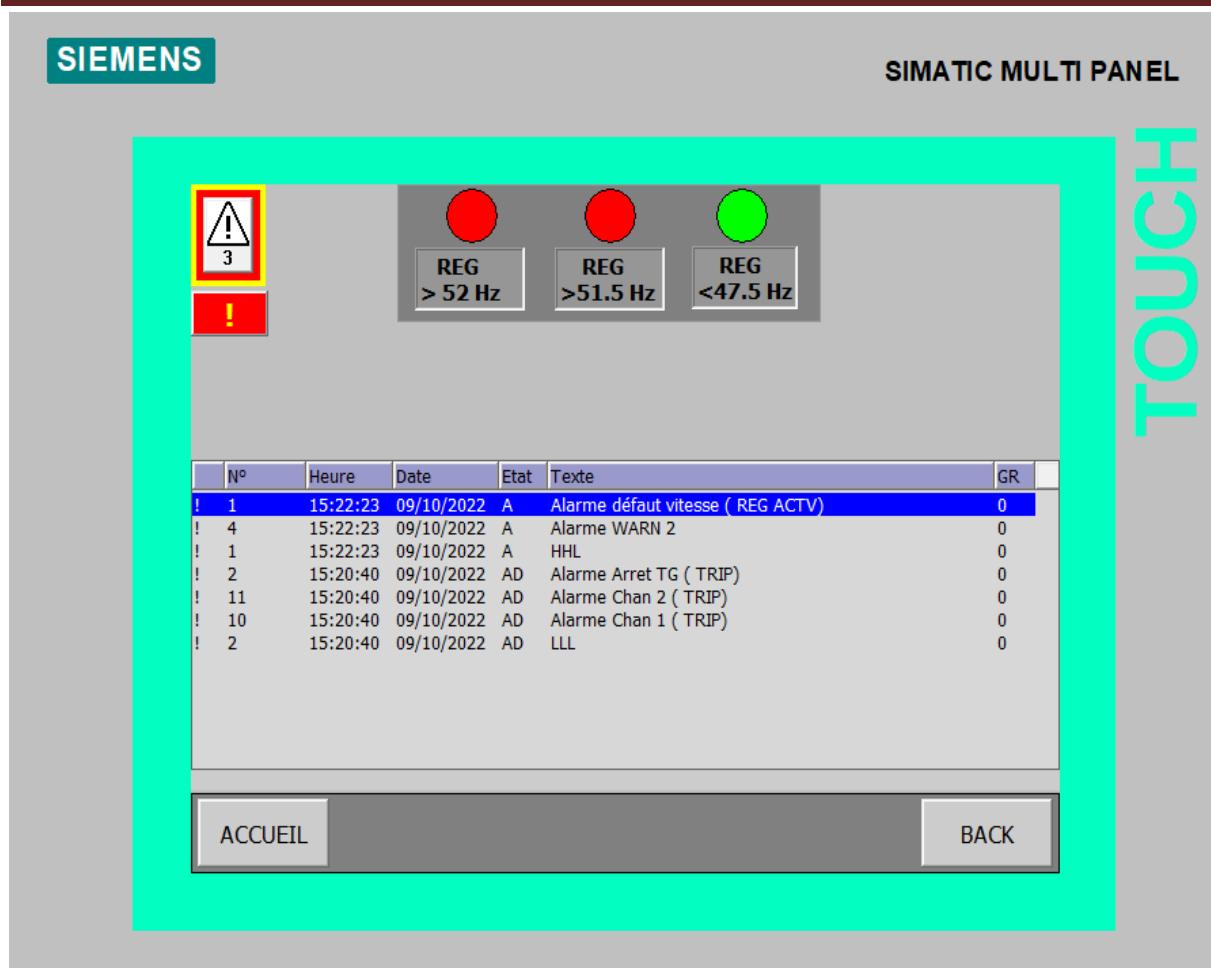


Figure III.16. Vue d'alarmes

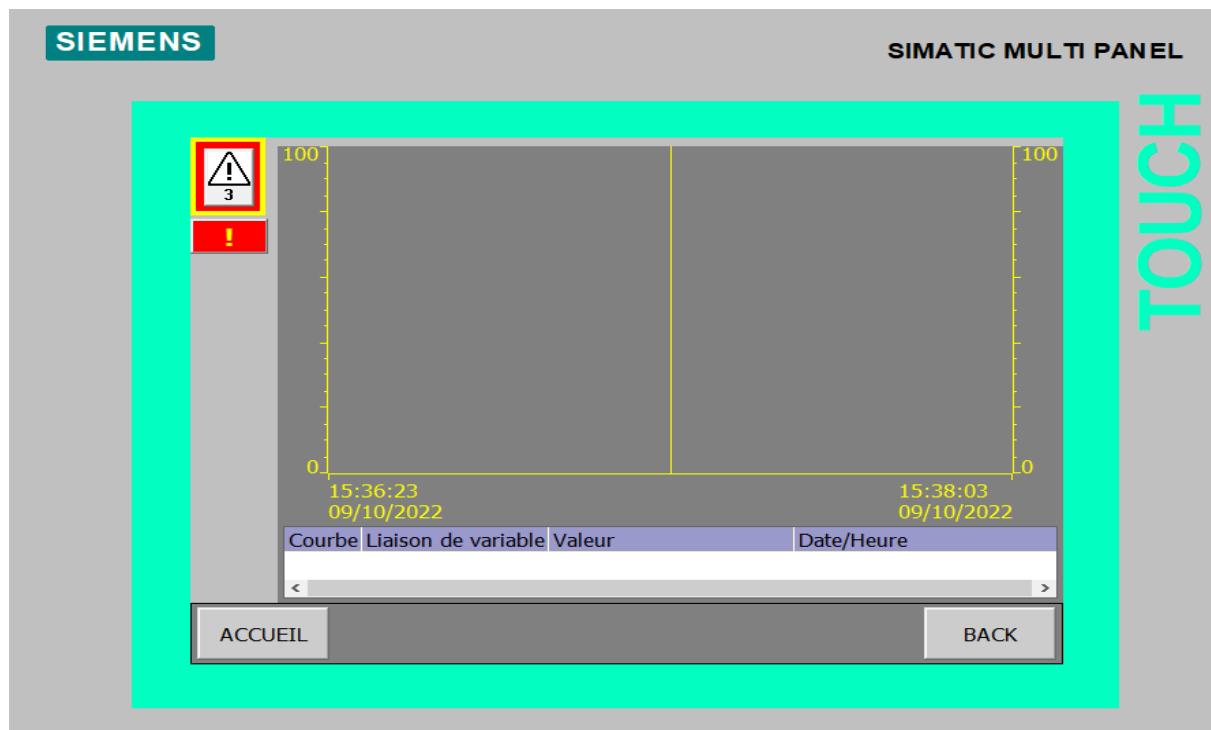


Figure III.17. Vue du graphe de variation de fréquence de vitesse de la turbine à gaz

### CHAPITRE III : Automatisation et supervision d'un système de commande de vitesse d'une turbine à gaz

Texte	Numéro	Classe	Variable de déclenchement	Numéro de bit	Adresse de dé...
Alarme Chan 2 (TRIP)	11	Erreurs	alarme_6	0	M 21.0
Alarme Chan 1 (TRIP)	10	Erreurs	alarme_5	0	M 20.0
Alarme WARN 2	4	Erreurs	alarme_4	0	M 14.0
Alarme WARN 1	3	Erreurs	alarme_3	0	M 13.0
Alarme Arret TG (TRIP)	2	Erreurs	alarme_2	0	M 12.0
Alarme défaut vitesse (REG ACTV)	1	Erreurs	alarme_1	0	M 11.0

Figure III.18. Alarme TOR.

Texte	Numéro	Classe	Variable survei...	Valeur limite s...	Déclencheur
HHL	1	Erreurs	capteur de vit...	53	Si front montant
LLL	2	Erreurs	capteur de vitess...	46	Si front descendant

Figure III.19. Alarme analogique

### III.5. Conclusion

Dans ce chapitre, nous avons donné le logigramme fonctionnel de la turbine à gaz, et la simulation de son fonctionnement sur SIMATIC STEP 7. Ensuite, nous avons réalisé une plateforme de supervision pour notre système à l'aide de logiciel SIMATIC WIN CC flexible. Cette supervision permet de montrer toutes les étapes de fonctionnement du système.

## Partie II :

---

# Réalisation d'un prototype de surveillance intelligente de défauts de vitesse au niveau de la turbine à gaz.

---

- ➡ Chapitre IV : Notions sur le Machine Learning (ML) et le TinyML.
- ➡ Chapitre V : Réalisation d'un prototype de surveillance intelligente.
  - Position du problème
  - Construction d'une base de données d'apprentissage
  - Déploiement du modèle d'apprentissage sur un microcontrôleur ATmega328P d'une carte ArduinoUno
  - Réalisation d'un Prototype (banc d'essai)
  - Résultats de test

## CHAPITRE IV

# Notions sur le Machine Learning et le Tiny ML.

---

## IV.1. Introduction

Dans ce chapitre, nous présentons un état de l'art de Machine Learning ou ML (en français, *apprentissage automatique*) en général et celui de Tiny Machine Learning, ou TinyML (en français, *apprentissage automatique minuscule*) en particulier. Tout d'abord, nous donnons un bref aperçu historique allant des prémisses à l'essor de ce domaine. Ensuite, nous donnons ses différents types et certains algorithmes utilisés.

## IV.2. Définitions du Machine Learning (ML)

Le machine Learning (ML) est une branche de l'intelligence artificielle qui permet aux ordinateurs d'apprendre sans être explicitement programmés, et cela en utilisant un des algorithmes qu'on appellera modèles et de données. Le ML est constitué d'un ensemble d'algorithmes avancés par lesquels les résultats peuvent être prédits sur la base d'un modèle préparé et formé sur des données d'entrée historiques et son comportement de sortie [11]. Cependant, selon l'approche machine Learning choisie, les performances et les avantages peuvent varier. De nos jours, les techniques du ML sont appliquées dans plusieurs domaines de fabrication tels que la maintenance, l'optimisation, le dépannage et le contrôle [12]. Durant ces dernières années, le Machine Learning est devenu de plus en plus important dans le domaine informatique car les données peuvent être collectées et stockées beaucoup plus facilement.

### La renaissance de l'apprentissage automatique :

En 2006, « Geoffrey Hinton et al » ont publié un article [13] montrant comment former un réseau neuronal profond capable de reconnaître des chiffres manuscrits avec une précision de pointe ( $>98\%$ ). Ils ont baptisé cette technique « Deep Learning » (apprentissage profond). La formation d'un réseau neuronal profond était largement considérée comme impossible à l'époque et la plupart des chercheurs avaient abandonné l'idée depuis les années 1990. Cet article a ravivé l'intérêt de la communauté scientifique et, très vite, de nombreux nouveaux articles ont démontré que l'apprentissage profond était non seulement possible, mais qu'il permettait d'obtenir des résultats époustouflants qu'aucune autre technique d'apprentissage automatique ne pouvait espérer égaler (grâce à une puissance de calcul énorme et à de grandes quantités de données). Cet enthousiasme s'est rapidement étendu à de nombreux autres domaines de l'apprentissage automatique. En 10 ans, l'apprentissage automatique a conquis l'industrie : il est aujourd'hui au cœur de la magie des produits de haute technologie, classant

les résultats des recherches sur le web, alimentant la reconnaissance vocale des smartphones et recommandant des vidéos, battant le champion du monde au jeu de Go etc.

### **IV.3. Utilisation et domaine d'application de Machine Learning**

#### **IV.3.1. Problématique du ML**

Le ML peut servir à résoudre des problèmes:

- Que l'on ne sait pas résoudre (comme dans l'exemple de la prédition d'achats) ;
- que l'on sait résoudre, mais dont on ne sait formaliser en termes algorithmiques comment nous les résolvons (c'est le cas par exemple de la reconnaissance d'images ou de la compréhension du langage naturel) ;
- que l'on sait résoudre, mais avec des procédures beaucoup trop gourmandes en ressources informatiques (c'est le cas par exemple de la prédition d'interactions entre molécules de grande taille, pour lesquelles les simulations sont très lourdes).

Le ML est donc utilisé quand les données sont abondantes (relativement), mais les connaissances peu accessibles ou peu développées.

#### **IV.3.2. Domaine d'application du ML**

De la détection de fraude à la reconnaissance vocale, en passant par la prévision des prix, les applications du Machine Learning sont nombreuses et peuvent améliorer considérablement divers domaines. En fonction de la nature des données, de la masse à traiter et de l'utilisation des informations obtenues, le choix d'appliquer un certain type d'algorithme peut varier. Quoi qu'il en soit le ML dispose d'un réel potentiel et peut permettre d'améliorer de nombreux domaines [11]. De même à partir de données, il permet de prédire, de diagnostiquer ou de recommander automatiquement tout ce qui intéresse une entreprise. C'est donc une technique transversale, qui s'applique à de très nombreux domaines, et dont tous les cas d'usage n'ont pas encore été trouvés, loin de là. En un mot, le Machine Learning peut emmener l'entreprise qui l'adopte au plus haut degré de l'analytique et remplir précisément les missions qui lui sont assignées. On présente ici quelques cas d'application :

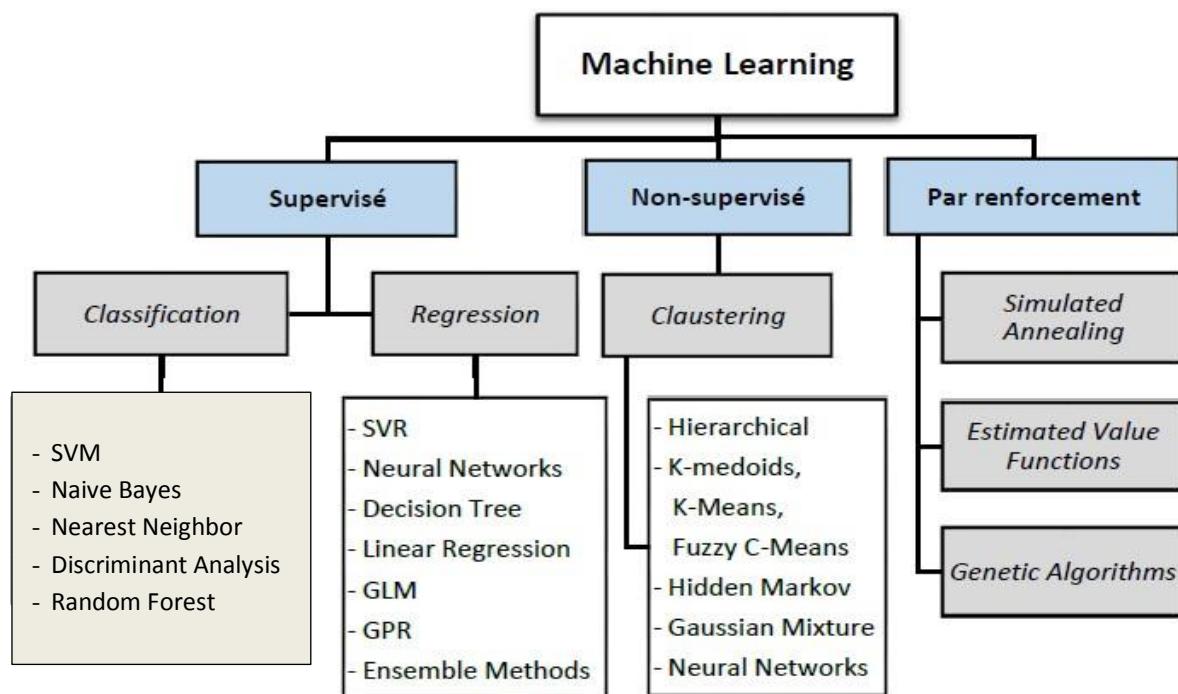
- **Finance – Assurance :**

- ✓ Modélisation d'indicateurs économiques,
- ✓ Personnalisation de l'expérience client,
- ✓ Evaluation de la solvabilité de l'emprunteur,

- ✓ Détection de fraudes,
  - ✓ Analyse de marché et analyse comparative.
- **Marketing – digital :**
- ✓ Segmentation clients au sein de DMP (data management Platform),
  - ✓ Analyse multi-canal, et Notation des prospects,
  - ✓ Optimisation de stratégies SEA (Google AdWords).
- **Industrie**
- ✓ Détermination et ajustement de prix, et Prévision des stocks,
  - ✓ Maintenance prédictive sur les données d'objets connectés.

#### IV.4. Approches du Machine Learning

Il existe deux grandes approches en apprentissage automatique : l'apprentissage non supervisé et l'apprentissage supervisé chacun procédant par ces propres méthodes comme sur la figure IV.1. L'apprentissage supervisé est ce qui nous intéresse pour la prédiction. Elle est constituée de plusieurs algorithmes que ce soit pour la régression ou la classification. Pour bien choisir un algorithme il faut comprendre les fondements des algorithmes existants et de ce qui permet de les distinguer ce qui permet de créer les modèles qui traiteraient au mieux un problème particulier.

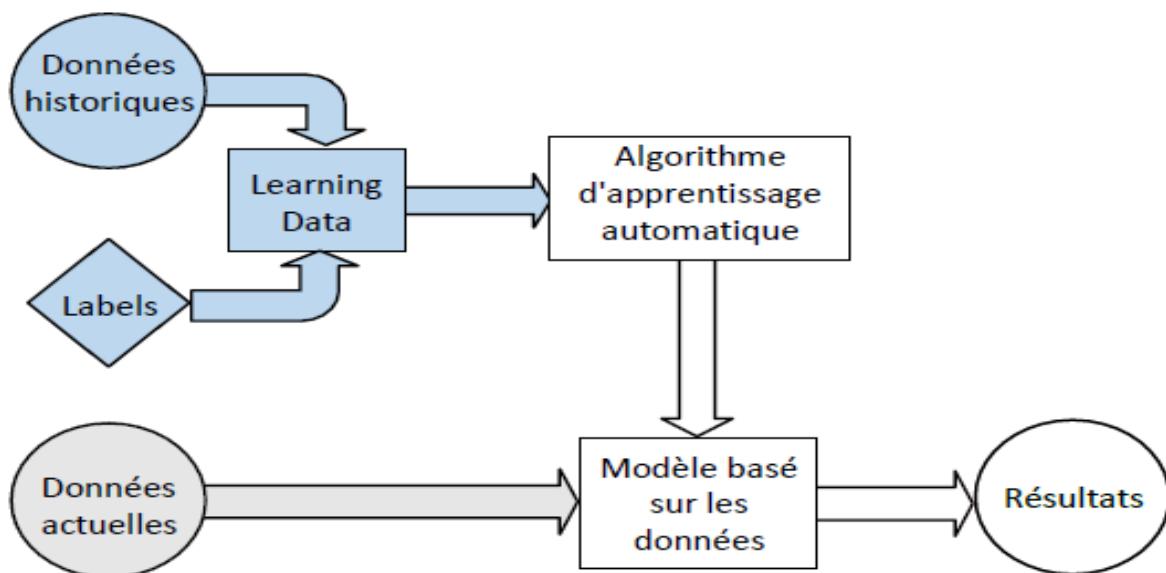


**Figure IV.1. Taxonomie des différentes techniques issues du ML.**

#### IV.4.1. Apprentissage automatique supervisé

L'apprentissage supervisé est un type d'apprentissage automatique qui utilise un ensemble de données connu pour effectuer des prédictions. L'ensemble de données d'apprentissage se compose de données d'entrée et de valeurs de réponse. Les algorithmes d'apprentissage supervisé cherchent à créer un modèle capable de prédire les valeurs de réponse d'un nouvel ensemble de données. L'utilisation d'ensemble de données d'apprentissage (ou donnée d'entraînement) plus volumineux et l'optimisation des hyper paramètres du modèle peuvent souvent augmenter le pouvoir prédictif du modèle et lui permettre de bien généraliser pour les nouveaux jeux de données (ensembles de données). Un ensemble de données de test est souvent utilisé pour valider le modèle [12].

Dans le domaine de la détection des types de défaillance et de la maintenance prédictive, la technique d'apprentissage supervisé est la technique d'apprentissage la plus couramment utilisée, car le système réel est surveillé et les données historiques sont disponibles. De plus, les données historiques peuvent être liées à l'état réel du système. Vu que tous les enregistrements de données collectées ne peuvent pas être utilisés comme exemples d'apprentissage, les données doivent être filtrées. Chaque enregistrement de données d'entraînement est étiqueté en fonction de son résultat attendu. L'ensemble de données résultant est appelé données d'entraînement. Il est traité par une technique d'apprentissage automatique qui examine la relation entre un enregistrement de données et la sortie étiquetée, puis crée un modèle basé sur les données. Pour toute nouvelle donnée, ce modèle essaie de donner le meilleur résultat basé sur les données apprises [12]. Les étapes d'apprentissage supervisé peuvent être représentées par la figure IV.2.



**Figure IV.2. Schéma synoptique qui illustre l'Apprentissage automatique supervisé.**

Du point de vue du Machine Learning, en fonction de la sortie de l'ensemble de données, deux classes de problèmes supervisés sont possibles : un problème de régression (si la sortie suppose des valeurs continues) et problème de classification (si la sortie prend des valeurs catégorielles) [11]. Ainsi, nous avons les algorithmes de classification et les algorithmes de régression.

### IV.4.1.1. Problème de classification

Dans l'apprentissage automatique et les statistiques, la classification est le problème qui consiste à identifier à quel groupe de catégories (sous-populations) une nouvelle observation appartient, à partir d'un ensemble d'apprentissages contenant des données dont l'appartenance à une catégorie est connue. Par exemple, attribuer un courrier électronique donné au classe « spam » ou « non-spam » et attribuer un diagnostic à un patient donné en fonction des caractéristiques observées du patient (sexe, pression artérielle, présence ou non de certains symptômes, ...etc.).

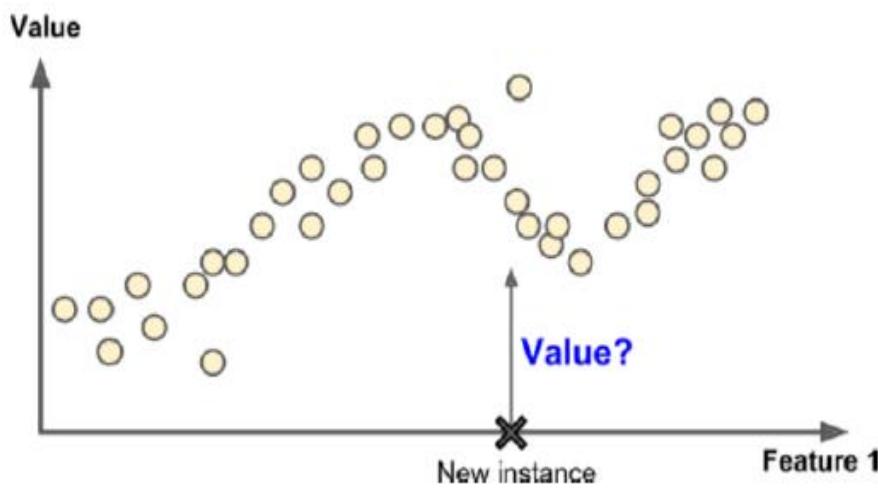
L'exemple donné dans la figure IV.3 exprime Le filtre anti-spam il est formé avec de nombreux exemples d'e-mails avec leur classe (spam ou ham), et il doit apprendre à classer les nouveaux e-mails.



Figure IV.3. Exemple de classification de spam

#### IV.4.1.2. Problème de régression

Une autre tâche typique consiste à prédire une valeur numérique cible, tel que le prix d'une voiture, en fonction d'un ensemble de caractéristiques (kilométrage, âge, marque, etc.) appelées prédicteurs. Ce type de tâche est appelé régression. Pour entraîner le système, nous devons lui donner de nombreux exemples de voitures, y compris leurs prédicteurs et leurs étiquettes (c'est-à-dire leurs prix). La figure IV.4 décrit une tâche de régression [14].



**Figure IV.4. Problème de Régression**

Noter que certains algorithmes de régression peuvent également être utilisés pour la classification, et vice-versa. Par exemple, la régression logistique est couramment utilisée pour la classification, car elle peut produire une valeur qui correspond à la probabilité d'appartenir à une classe donnée (par exemple, 20% de chances d'être un spam).

Voici quelques-uns des algorithmes d'apprentissage supervisé les plus importants :

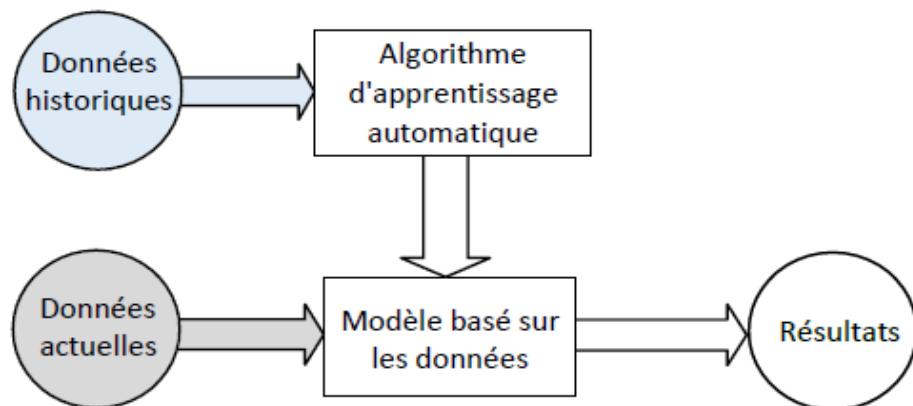
- Réseaux de neurones artificiels (RNA)
- Arbres de décision et forêts aléatoires (Radom Forest).
- Régression linéaire ;
- Régression logistique ;
- Machines à vecteurs de support (SVM);

#### IV.4.2. Apprentissage automatique non supervisé

Dans l'apprentissage non supervisé, les données d'apprentissage ne sont pas étiquetées. Le système tente d'apprendre sans professeur.

Le modèle n'a pas de « réponses » dont il peut tirer des enseignements ; il doit donner un sens aux données en fonction des observations elles-mêmes.

Les tâches non supervisées les plus courantes sont le clustering, dues au fait que les clusters (labels) ne sont pas connues à l'avance [13]. Les ordinateurs apprennent à identifier par eux-mêmes les similitudes et les distinctions entre les données, et à regrouper ceux qui partagent des caractéristiques communes sans aucune orientation humaine stricte et rigoureuse [11] (voir figure IV.5).



**Figure IV.5. Schéma synoptique qui illustre l'Apprentissage automatique non supervisé.**

L'apprentissage non supervisé est fréquemment appliqué dans l'analyse de grappes, à savoir les systèmes de recommandation, le regroupement de clients et la réduction de la dimensionnalité des facteurs [11]. L'apprentissage automatique non supervisé peut être comparé à un enfant qui apprend à identifier le type de fruit en observant le motif et la couleur, au lieu de mémoriser les noms avec l'aide d'une autre personne. Il recherche des similitudes entre les images, les séparant ainsi en groupes, tout en attribuant à chaque groupe son propre label [12].

#### IV.4.3. Apprentissage semi-supervisé

Nous avons préalablement vue l'apprentissage supervisé et non supervisé, dont la majeur différence réside dans le fait que les données soient étiquetées ou non, et à cela s'ajoute les méthodes adéquates utilisées pour traiter ses données.

L'apprentissage semi-supervisé regroupe ses deux principes, il prend un ensemble réduit de données étiquetées avec un autre ensemble de données non étiquetées du même types.

L'avantage de ce type d'apprentissage réside principalement dans le processus d'étiquetage des données prend beaucoup de temps et souvent coûteux.

Donc paradoxalement le non étiquetage devient bénéfique pour le processus d'apprentissage, et la construction du modèle est moins coûteuse. [14]

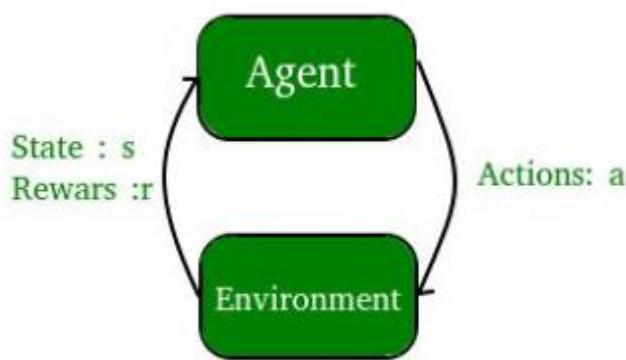
#### IV.4.4. Apprentissage par Renforcement

Entraînement par renforcement est une technique dans laquelle on immerge un agent dans un environnement où celui-ci interagit avec son environnement dans le but d'apprendre.

Un agent dans un environnement est muni de **sensors** pour capter les informations de son environnement, ainsi que d'**actionneurs** qui lui permettront d'agir dans son environnement.

Les agents observent l'entrée, puis il effectue une action en prenant des décisions.

Une fois l'action réalisées, l'agent reçoit des récompenses en conséquence, ce qui renforce le modèle en stockant ses informations dans une base de données. La récompense peut être positive, négative ou nulle selon les actions effectuées.



**Figure IV.6. Apprentissage par renforcement**

## IV.5. Quelques Algorithmes d'apprentissage supervisé

### IV.5.1. Arbre de décision DT (Decision Tree)

Un arbre de décision *DT* (*Decision Tree*) est un enchaînement hiérarchique de règles logiques ou de production construite de manière automatique à partir d'un ensemble de données. La construction de l'arbre de décision consiste à utiliser les descripteurs, pour subdiviser progressivement l'ensemble en sous-ensembles de plus en plus fins. L'algorithme estime la probabilité qu'une observation se trouve dans le nœud  $t$  en utilisant la formule :

$$P(T) = \sum_{j \in T} W_j ; \text{ Où :}$$

$W_j$ : Poids des observations,

$T$  : Ensemble de tous les indices d'observation du noeud  $t$ ,

$n$  : Égale au nombre d'observations.

Les arbres de décision sont des algorithmes de classification de données reposant sur un apprentissage supervisé. Comme son nom l'indique, la structure de ces algorithmes ressemble à des arbres constitués de noeuds, de branches et de feuilles (voir figure IV.7). La construction de ces arbres est réalisée à l'aide d'une base de données brute (vecteur des caractéristiques et classes) et de lois qui permettent de déterminer les variables discriminantes pour la classification efficace des futures données. Chacun des noeuds constituant l'arbre représente une règle de classification préalablement déterminée de manière récursive.

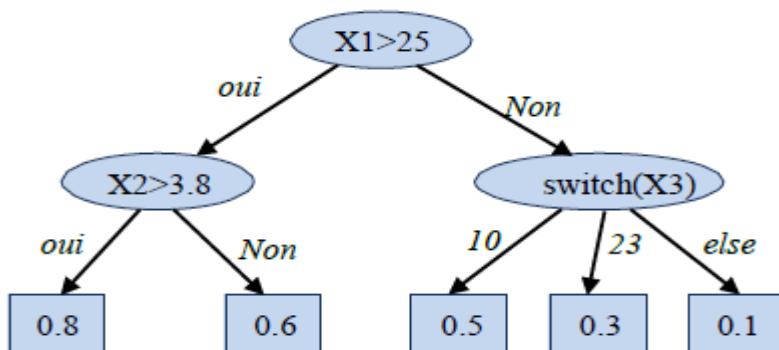


Figure IV.7. Exemple de l'arbre de décision (DT).

Le principal inconvénient des arbres de décision est qu'ils ont tendance à se surajuster et à fournir de mauvaises performances de généralisation. C'est pourquoi, dans la plupart des applications, les méthodes d'ensemble comme les forêts aléatoires dont nous parlons ci-après sont généralement utilisées à la place d'un arbre de décision unique.

#### IV.5.2. Les forêts aléatoires (Random forest )

Radom Forest (RF) est un algorithme d'apprentissage supervisé pour les tâches de classification et de régression. Comme son nom l'indique (forêt aléatoire), un RF crée un ensemble (une forêt) avec plusieurs arbres de décision aléatoires. Le RF entraîne le système plusieurs fois avec un échantillon de jeu de données arbitraire pour fournir un modèle de prédiction extraordinaire. Il donne une prédiction définitive en utilisant les résultats DT comme dans la méthode d'apprentissage d'ensemble. S'il s'agit d'une prédiction qui se produit plusieurs fois dans l'arbre de décision, il s'agit de la prédiction de haut niveau de la forêt aléatoire.

La puissance centrale de l'algorithme de forêt aléatoire réside dans sa commodité pour résoudre à la fois la classification et la régression qui permettent de bons calculs de celles-ci. Il gère les grands ensembles de données très doucement sans perdre la dimensionnalité. Il fournit des méthodes efficaces pour analyser les données manquantes. Le but de la méthode RF est d'obtenir des résultats plus efficaces avec plus d'un décideur comme dans d'autres méthodes. La différence de cette méthode par rapport aux autres méthodes est que les variables sont sélectionnées au hasard lorsque les branches se ramifient [11].

#### ➤ Bagging et Boosting dans le Machine Learning

Comme nous le savons, l'apprentissage d'Ensemble permet d'améliorer les résultats de l'apprentissage automatique en combinant plusieurs modèles. Cette approche permet de produire de meilleures performances prédictives par rapport à un modèle unique. L'idée de base est d'apprendre un ensemble de classificateurs (experts) et de leur permettre de voter. **Le Bagging et le Boosting** sont deux types d'**Ensemble Learning**. Ces deux éléments diminuent la variance d'une seule estimation car ils combinent plusieurs estimations de différents modèles.

1. **Bagging** : C'est un modèle homogène d'apprenants faibles qui apprend les uns des autres indépendamment en parallèle et les combine pour déterminer la moyenne du modèle.
2. **Boosting** : C'est aussi un modèle homogène d'apprenants faibles mais fonctionne différemment du Bagging. Dans ce modèle, les apprenants apprennent de manière séquentielle et adaptative pour améliorer les prédictions du modèle d'un algorithme d'apprentissage.

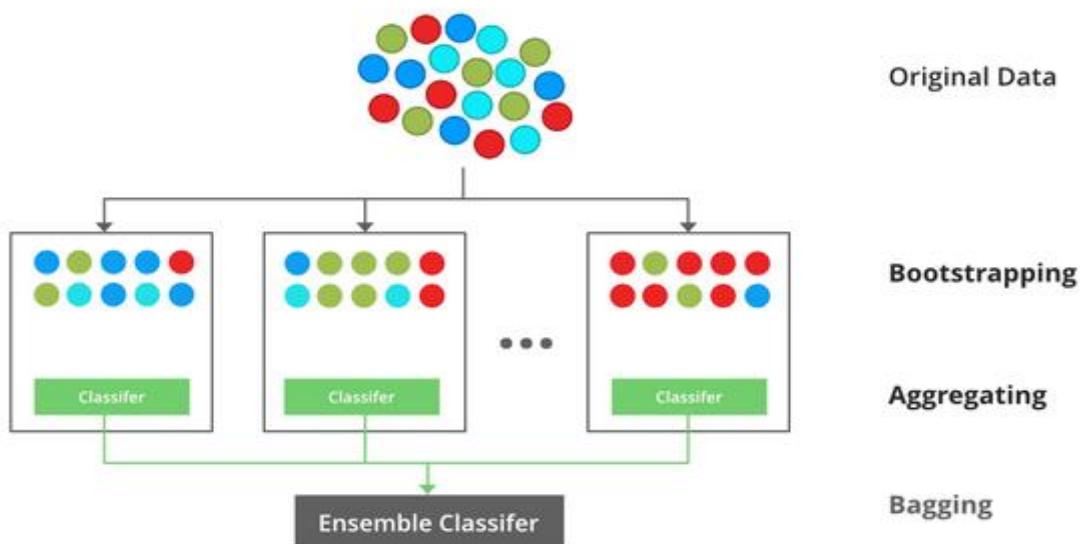
### ➤ Description de la technique

Supposons un ensemble  $D$  de  $d$ -tuples, à chaque itération  $i$ , un ensemble  $D_i$  d'apprentissage de  $d$  tuples est échantillonné avec remise à partir de  $D$  (c'est-à-dire bootstrap). Ensuite, un modèle de classificateur  $M_i$  est appris pour chaque ensemble d'apprentissage  $D < i$ . Chaque classifieur  $M_i$  renvoie sa prédiction de classe.

Le classificateur ensaché  $M^*$  compte les votes et attribue la classe avec le plus de votes à  $X$  (échantillon inconnu).

### ➤ Étapes de mise en œuvre de la technique :

- ⊕ **Étape 1:** plusieurs sous-ensembles sont créés à partir de l'ensemble de données d'origine avec des tuples égaux, en sélectionnant les observations avec remplacement.
- ⊕ **Étape 2 :** Un modèle de base est créé sur chacun de ces sous-ensembles.
- ⊕ **Étape 3:** Chaque modèle est appris en parallèle à partir de chaque ensemble de formation et indépendamment les uns des autres.
- ⊕ **Étape 4:** Les prédictions finales sont déterminées en combinant les prédictions de tous les modèles.

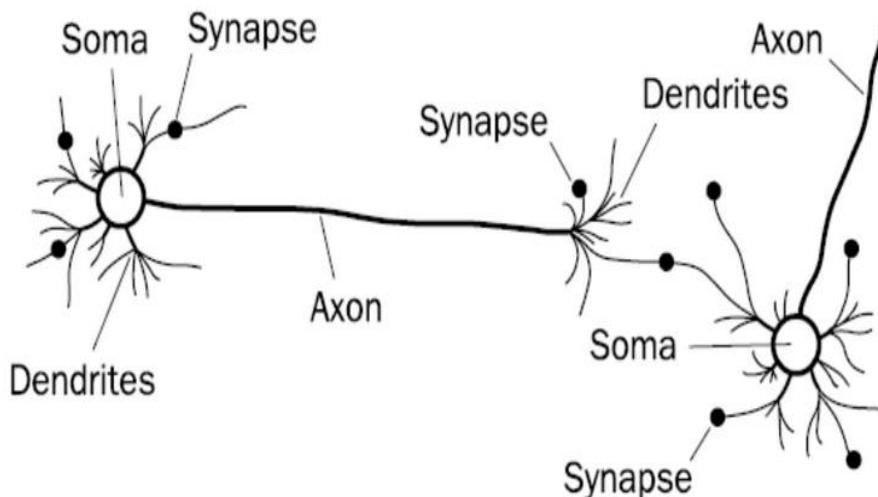


**Figure IV. 8. Schéma d'illustration du concept d'agrégation bootstrap (Bagging)**

### IV.5.3. Les réseaux de neurones artificiels

Les réseaux de neurones ont été développés pour simuler le système nerveux humain pour des tâches d'apprentissage automatique en traitant les unités de calcul dans un modèle d'apprentissage d'une manière similaire aux neurones humains. La grande vision des réseaux de neurones est de créer une intelligence artificielle en construisant des machines dont l'architecture simule les calculs dans le système nerveux humain.

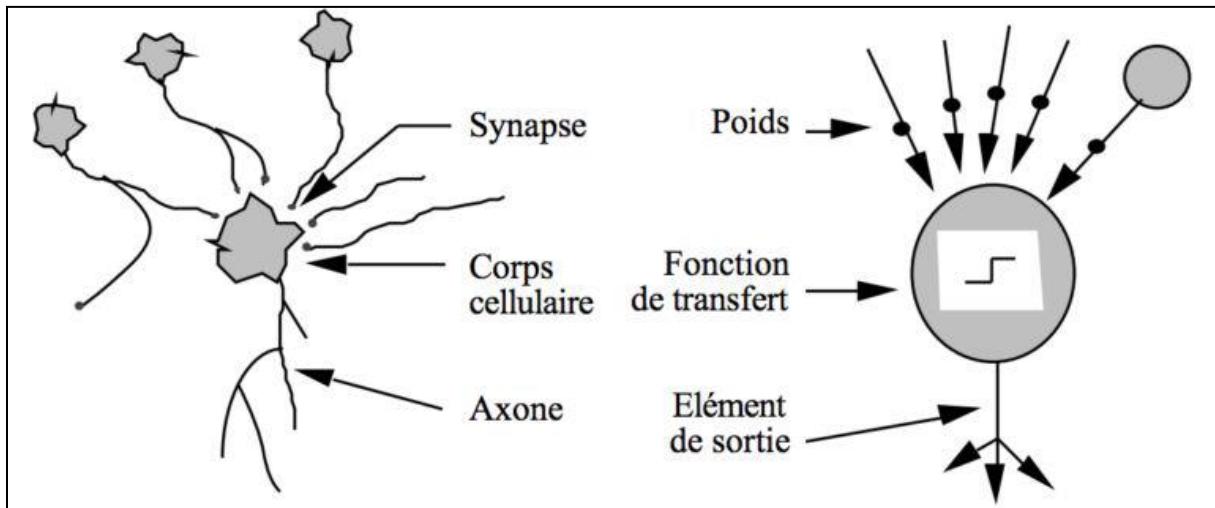
Les réseaux de neurones sont théoriquement capables d'apprendre n'importe quelle fonction mathématique avec suffisamment de données d'entraînement et ils sont devenus des techniques d'apprentissage automatique populaires qui simulent le mécanisme d'apprentissage dans les organismes biologiques. Le système nerveux humain contient des cellules, appelées neurones. Les neurones sont connectés les uns aux autres grâce à l'utilisation d'axones et de dendrites, et les régions de connexion entre les axones et les dendrites sont appelées synapses [14]. Ces connexions sont illustrées à la figure IV.9.



**Figure IV.9. Réseau de neurones biologique [14]**

Les forces des connexions synaptiques changent souvent en réponse à des stimuli externes. Ce changement est la façon dont l'apprentissage prend place dans les organismes vivants. Ce mécanisme biologique est simulé dans des réseaux de neurones artificiels, qui contiennent des unités de calcul appelées neurones. Les unités de calcul sont reliées les unes aux autres par des poids, qui jouent le même rôle que les forces des connexions synaptiques dans les organismes biologiques. Chaque entrée d'un neurone est mise à l'échelle avec un poids, ce qui affecte la fonction calculée à cette unité (voir figure IV.10).

On retrouve **Poids Synaptiques**, **fonction d'activation** ainsi que **éléments de sortie** qui sont similaires aux fonctions occupées par (Synapses, corps cellulaire et Axone) chez le neurone biologique. Cette architecture est illustrée à la figure IV.10.



**Figure IV.10. Neurone biologique VS Neurone artificiel**

#### IV.6. Machine Learning vs Deep Learning (DL)

Deep Learning ou apprentissage en profondeur ou **DL** est une branche du Machine Learning entièrement basée sur des **réseaux de neurones artificiels**. [14]

La majeure différence qu'on note entre ses 2 concepts (Machine Learning et Deep Learning ) provient de la manière dont les données sont présentées au système (modèle).

- ✚ Les algorithmes de ML nécessitent presque toujours des données structurées, alors que les réseaux d'apprentissage approfondis reposent sur des couches de réseaux de neurones artificiels (RNA).
- ✚ On voit aussi une différence au sein de l'architecture des modèles qui les composent, on note que les modèles type DL sont plus profond que les modèles type ML.
- ✚ Deep learning n'utilise que les réseaux de neurones, alors que pour le ML les réseaux de neurones sont qu'une approche de conception des modèles.

En considérant le fait que le DL est la prochaine étape de l'évolution du ML inculquant aux machines la manière de prendre leurs décisions de façon précise sans l'intervention de l'expert humain.

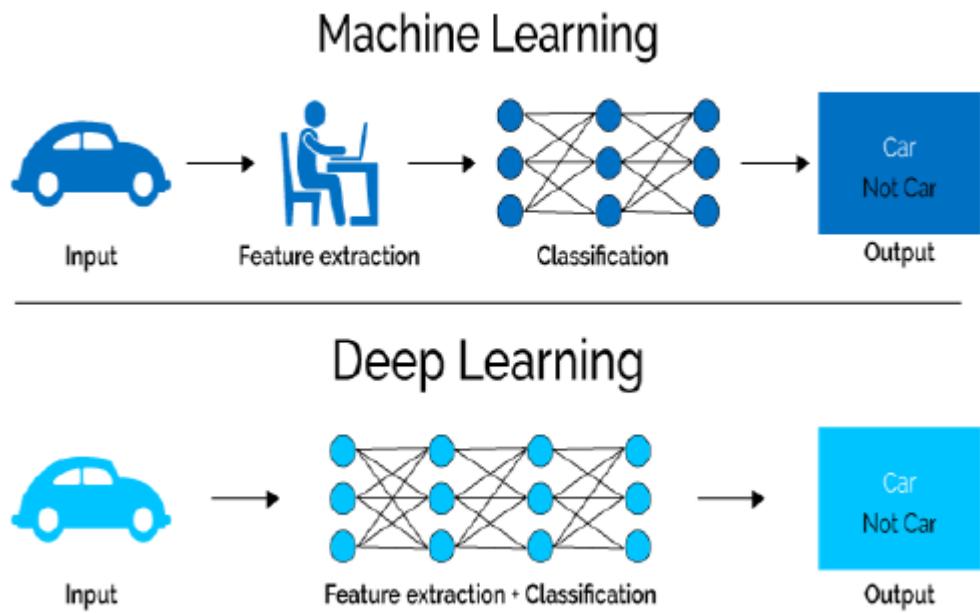


Figure IV.11. ML vs DL.

## IV.7. Internet des Objets, ou Internet of Things (IoT) :

L'Internet des Objets, ou Internet of Things (IoT), a fait le sujet de plusieurs innovations depuis son apparition. Aujourd'hui, l'IoT est un domaine d'investissement rentable car l'IoT est la technologie du futur monde.

### IV.7.1. Définitions

➤ **Définition d'un objet connecté :**

« Un **objet connecté** est un objet physique équipé de capteurs ou d'une puce qui lui permettent de transcender son usage initial pour proposer de nouveaux services. Il s'agit d'un matériel électronique capable de communiquer avec un ordinateur, un smartphone ou une tablette via un réseau sans fil (**Wi-Fi**, **Bluetooth**, réseaux de téléphonie mobile, réseau radio à longue portée de type **Sigfox** ou **LoRa**, etc.), qui le relie à Internet ou à un réseau local ».

➤ **Définition de l'IoT**

D'après l'Union Internationale des Télécommunications « *L'Internet des Objets (IdO) est une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution* ».

## IV.7.2. Composants d'un objet connecté

Les composants de base d'un objet connecté sont les suivants : [12]

### IV.7.2.1. Le microcontrôleur

Un microcontrôleur est un circuit intégré composé des éléments de base suivants :

- ✚ **Microprocesseur** qui se charge des calculs.
- ✚ **Mémoire permanente (ROM)** qui contient le programme à exécuter.
- ✚ **Mémoire temporaire (RAM)** qui contient les données temporaires.
- ✚ **Des ports d'entrée/sortie.**

Il en existe de différents types de microcontrôleurs pour le développement des projets d'IoT, nous citons les suivants :

#### 1- Le microcontrôleur Arduino

On cite par exemple La carte Arduino Uno est le produit le plus populaire parmi les cartes Arduino. La carte Arduino Uno, comme son nom l'indique, a été la première à utiliser la version de programmation Arduino 1.0, et elle est devenue le symbole de l'univers Arduino.

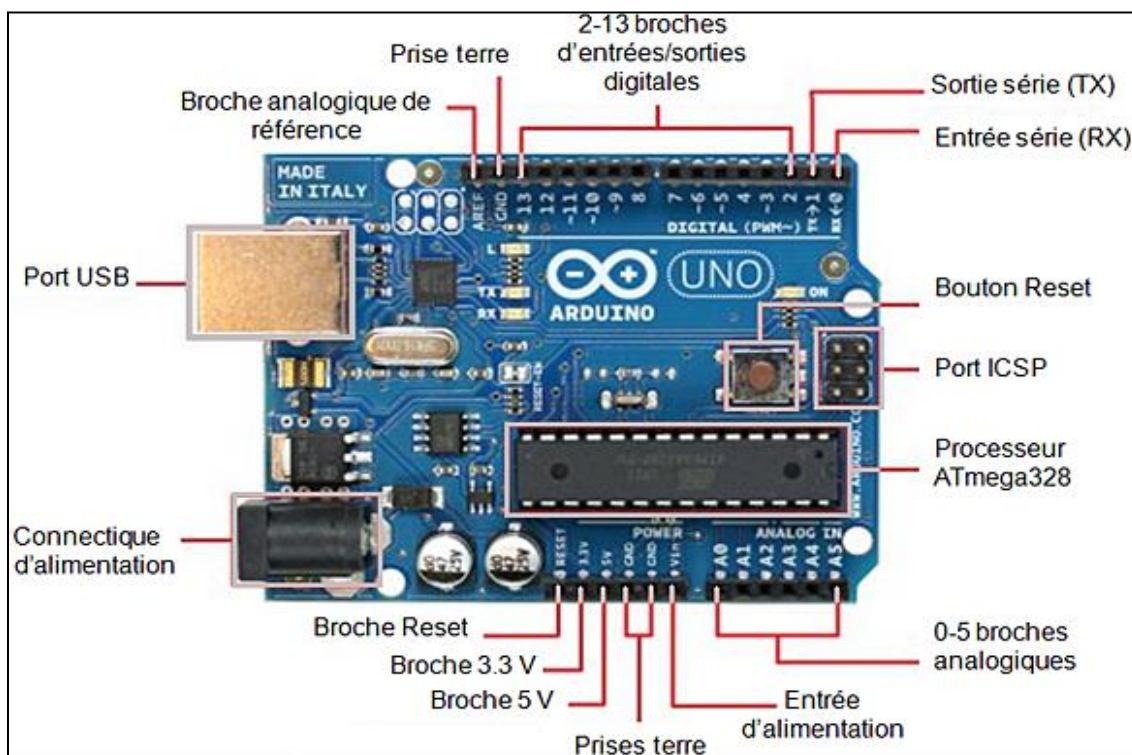


Figure IV.12. Carte Arduino UNO

## 2- Le microcontrôleur Raspberry Pi

Raspberry Pi 3 est un microordinateur avec un module Bluetooth et Wifi intégré. Très simple d'utilisation il est léger et possède une capacité de mémoire et de calculs supérieurs aux Arduino (Figure IV.13).

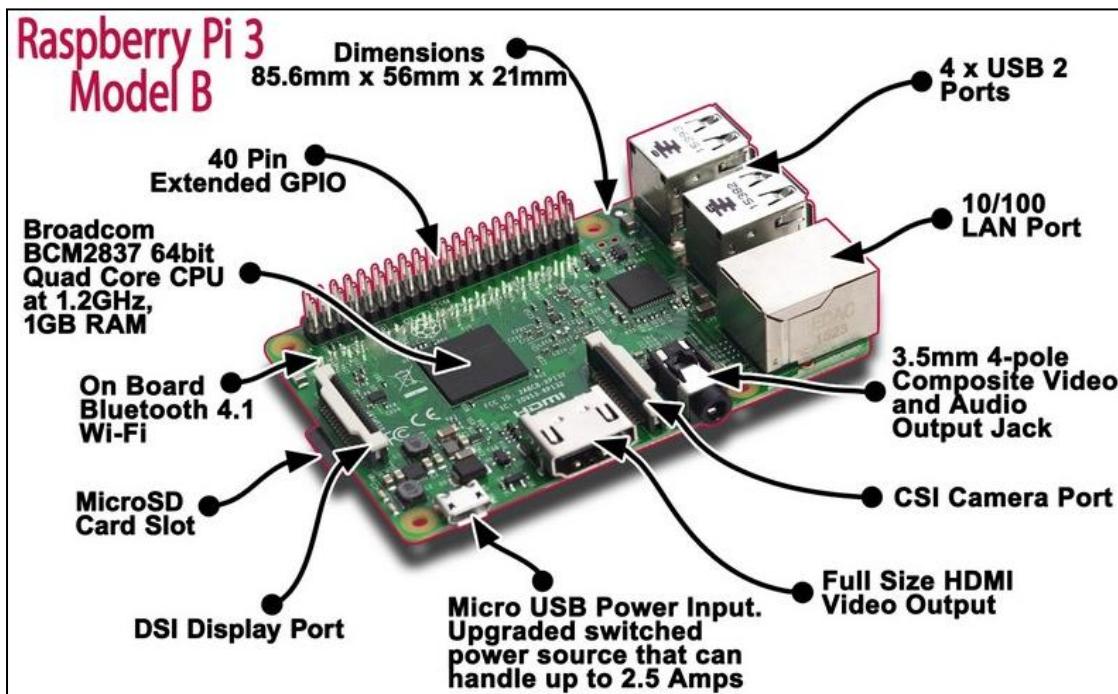


Figure IV.13. Le microcontrôleur Raspberry Pi 3 Modèle B

### IV.7.2.2. Les capteurs et les actionneurs

On cite ci-dessous quelques capteurs :

- ✓ Capteur de température LM35 : fournit la température de son entourage en Celsius avec un large intervalle.
- ✓ Capteur de gaz MQ135 : le capteur fournit deux valeurs en sortie de la quantité de gaz (CO<sub>2</sub>, fumée, etc...) dans l'air.
- ✓ Capteur de mouvements HC-SR501 PIR : le capteur peut détecter le mouvement d'un être-humain ou d'un objet, avec une couverture de 120° comme angle et 7 mètres comme distance.

On cite ci-dessous quelques actionneurs :

- ✓ **LED** : C'est une composante électronique émettant un signal lumineux, généralement blanc, rouge, vert ou bleu.
- ✓ **Buzzer** : C'est une composante électronique émettant un signal sonore, l'intensité du son dépend de l'intensité du courant.

### IV.7.3. Protocoles de communication et Plateformes de développement

Il en existe plusieurs protocoles de communication, assurant l'échange de données entre les différentes parties d'un système d'IoT, nous citons à titre d'exemples :

- ✓ **HTTP** (Hypertext Transfer Protocol) : protocole de transfert hypertexte. Ce protocole définit la communication entre un client (exemple : navigateur) et un serveur sur le World Wide Web (WWW).

De plus, il en existe plusieurs environnements et plateformes de développement pour la programmation des systèmes d'IoT, pouvant être utilisé par les concepteurs pour le développement des projets IoT, nous citons les suivantes :

- ✓ **L'environnement Arduino** : destiné aux microcontrôleurs Arduino, avec le langage pseudo C.
- ✓ **L'environnement Cloud9** : pour la programmation des microcontrôleurs BeagleBone.

### IV.7.4. Domaines d'application

Plusieurs domaines d'application sont touchés par l'IoT. Parmi ces principaux domaines nous citons:

- Les villes intelligentes (Smart Cities).
- La domotique (Smart Home / Home Automation).
- Agriculture intelligente (Smart Agriculture).
- Infographie E-Santé

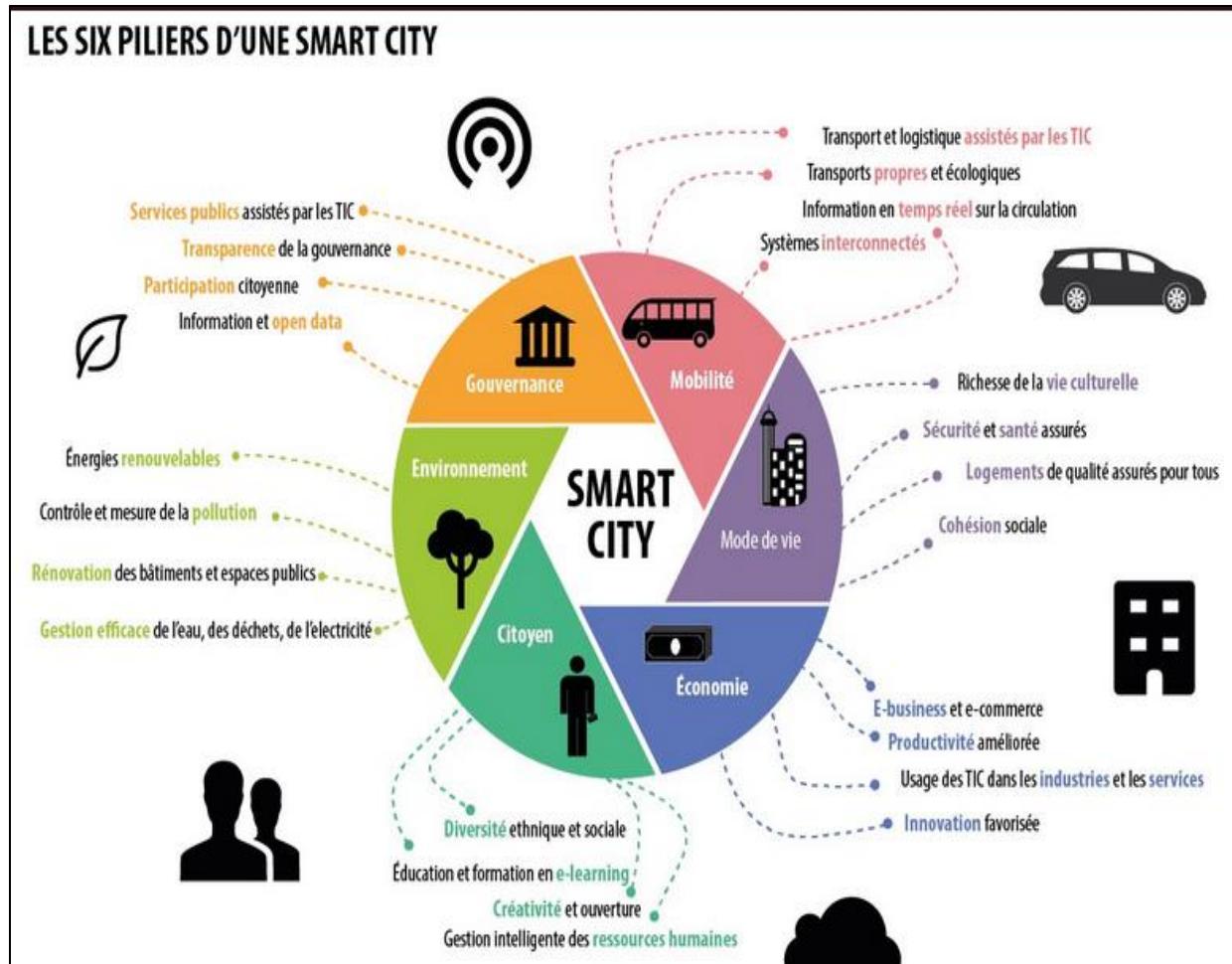


Figure IV.14. Les six piliers d'une Smart City

## IV.8. TinyML : Machine Learning sur microcontrôleur

### IV.8.1. Définition

Tiny Machine Learning (ou TinyML) est une méthode d'apprentissage automatique qui combine des options d'apprentissage automatique légères et optimisées qui nécessitent **des** solutions complètes (matériel, système, logiciels et applications), y compris des architectures d'apprentissage automatique, des techniques, des outils et des approches capables d'effectuer analytique sur un appareil en périphérie du cloud.

TinyML peut être implémenté dans des systèmes à faible consommation tels que des capteurs ou des microcontrôleurs afin d'effectuer des tâches automatisées.

Avec TinyML, nous pouvons faire plus avec moins. Il s'agit toujours d'apprentissage automatique, mais avec moins de puissance, de coût et aucune connectivité réseau.

#### IV.8.2. Les opportunités et les défis pour TinyML

C'est ainsi que Tiny Machine Learning (ou TinyML) peut être brièvement décrit sans exagération, une percée dans le domaine de l'intelligence artificielle. Il convient de garder à l'esprit que, selon les prévisions d'ABI Research, d'ici 2030, environ 2,5 milliards d'appareils entreront sur le marché en utilisant les technologies TinyML, dont le principal avantage sera la création d'appareils IoT intelligents et, non moins important, leur vulgarisation, en raison de la réduction probable des coûts.

De plus, une étude du cabinet de conseil "**The Silent Intelligence**" confirme une prévision précédente selon laquelle la valorisation économique du marché TinyML pourrait dépasser le cap des 70 milliards de dollars dans les cinq prochaines années. Ces chiffres ne peuvent pas passer inaperçus. Plusieurs entreprises travaillent déjà sur la construction de puces qui seront utilisées dans les implémentations TinyML.

Si l'on y réfléchit, nous sommes déjà entourés d'appareils alimentés par batterie qui utilisent ML sous le capot. Par exemple, les dispositifs portables, comme les smartwatches et les bracelets de suivi de la condition physique, peuvent reconnaître les activités humaines pour suivre nos objectifs de santé ou détecter des situations dangereuses, comme une chute au sol. Ces objets de tous les jours sont des applications TinyML à toutes fins utiles, car ils sont alimentés par des piles et ont besoin d'un ML sur l'appareil pour donner un sens aux données acquises par les capteurs.

Cependant, les solutions alimentées par batterie ne se limitent pas aux seuls dispositifs portables. Il existe des scénarios dans lesquels nous pouvons avoir besoin de dispositifs pour surveiller des environnements. Par exemple, nous pourrions envisager de déployer des dispositifs alimentés par batterie et exécutant une analyse ML dans une forêt pour détecter les incendies et empêcher qu'ils ne se propagent sur une grande surface.

Les cas d'utilisation potentiels de TinyML sont illimités, et ceux que nous venons de présenter brièvement ne sont que quelques-uns des domaines d'application probables. Cependant, parallèlement aux opportunités, il y a quelques défis critiques à relever. Les défis sont d'ordre informatique car nos appareils sont limités en mémoire et en puissance de traitement. Nous travaillons sur des systèmes dotés de quelques kilo-octets de RAM et, dans certains cas, de processeurs sans accélération arithmétique à virgule flottante.

#### IV.8.3. TensorFlow Lite pour les microcontrôleurs

TinyML utilise la même architecture et la même approche de l'apprentissage automatique, mais sur de petits appareils capables d'effectuer une variété de fonctions, de la réponse à des commandes sonores à l'exécution d'actions via des interactions chimiques. Mais comment implémenter TinyML ? Des outils spéciaux peuvent nous aider à exécuter des modèles d'apprentissage automatique sur des appareils IoT. Le plus célèbre d'entre eux est Tensorflow Lite.

TensorFlow Lite est un ensemble d'outil conçus pour exécuter des modèles de machine Learning sur des microcontrôleurs et d'autres appareils ne disposant que de quelques kilo-octets de mémoire. Le composant d'exécution principal tient sur 16 Ko sur un processeur Arm Cortex M3 et peut exécuter de nombreux modèles de base. Aucune compatibilité avec le système d'exploitation n'est nécessaire. Il ne nécessite, en outre, aucune bibliothèque C ou C++ standard, ni d'allocation de mémoire dynamique.

Le langage préféré pour créer des modèles d'apprentissage automatique est généralement Python, mais avec TensorFlow Lite, vous pouvez utiliser C, C ++ ou Java pour créer des modèles.

La connexion au réseau est une opération énergivore. À l'aide de Tensorflow Lite, vous pouvez déployer des modèles de Machine Learning en ayant besoin d'une connexion Internet. Il résout également les problèmes de sécurité car il est relativement plus facile d'exploiter les vulnérabilités des systèmes embarqués.

Tensorflow Lite propose déjà un ensemble de modèles d'apprentissage automatique pré-entraînés pour une utilisation quotidienne. Ceux-ci inclus:

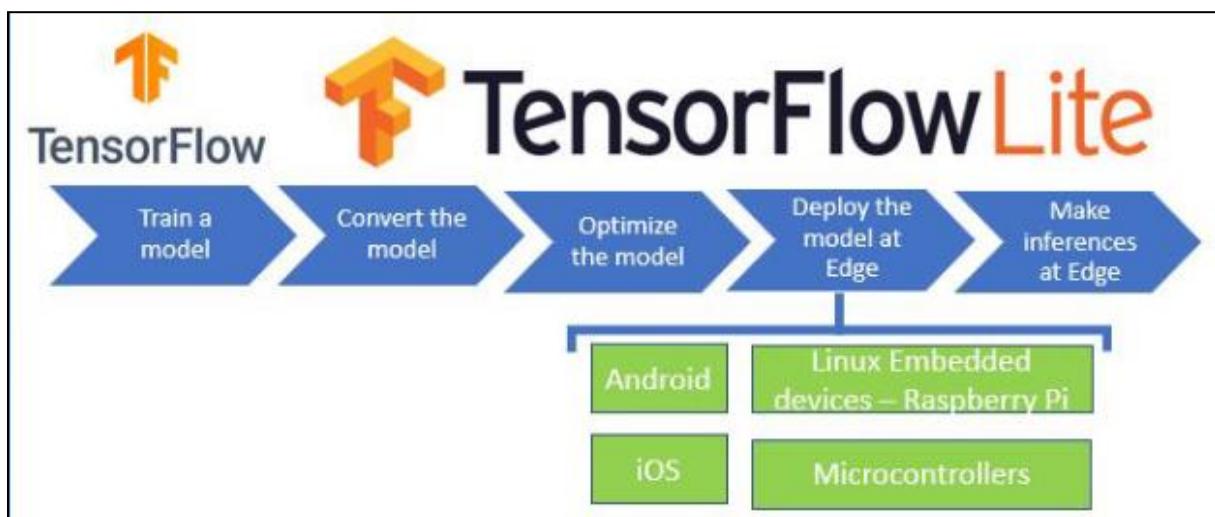
- Détection d'objets - utilisée pour reconnaître plusieurs objets dans une image, avec prise en charge de jusqu'à 80 éléments différents.
- Réponses intelligentes - génère des réponses intelligentes similaires à celles que vous obtenez lorsque vous interagissez avec une IA conversationnelle.
- Recommandations - propose des systèmes de recommandations individuelles basées sur le comportement des utilisateurs.

Les étapes suivantes sont requises pour déployer et exécuter un modèle TensorFlow sur un microcontrôleur :

**1. Entraîner un modèle :**

- Générez un petit modèle *TensorFlow* adapté à votre appareil cible et contenant des opérations compatibles.
- Convertissez un modèle au format *TensorFlow Lite* à l'aide du *convertisseur TensorFlow Lite*.
- Effectuez une conversion en tableau d'octets *C* à l'aide d'outils standards en vue du stockage dans une mémoire de programme en lecture seule sur l'appareil.

**2. Exécutez une inférence** sur l'appareil à l'aide de la bibliothèque C++ et traitez les résultats.



**Figure IV.15. Conception du TensorFlow.**

## IV.9. Conclusion

Dans ce premier chapitre, nous avons présenté de manière globale les principaux axes de l'apprentissage automatique ML et TinyML, certaines définitions, ses types, quelques algorithmes utilisés en particulier apprentissage supervisé, avant de finir par décrire l'outil TensorFlow Lite qui est conçu pour exécuter des modèles de machine Learning sur des microcontrôleurs.

Le chapitre suivant sera consacré à l'application des techniques TinyML et TensorFlow pour la réalisation d'un système intelligent de détection des défauts de notre système étudié dans la partie I de ce mémoire.

## CHAPITRE V

Réalisation d'un prototype de surveillance intelligente à l'aide de TinyML à base d'une carte Arduino.

---

## V. 1. Introduction

Dans de nombreux secteurs industriels et pétroliers, l'activité de surveillance dans les machines tournantes est une tâche très complexe et nécessite une grande quantité d'informations et des données intrinsèques, concernant le fonctionnement de ces processus.

A l'origine, la surveillance était destinée à assurer la protection d'une machine en déclenchant automatiquement son arrêt, ou en générant des alarmes avant que ne surviennent des dégradations graves. En effet, le développement actuel des nouvelles technologies, a permis l'amélioration des performances qui s'expriment à travers différents dispositifs, et qui contribuent dans l'évolution des systèmes de surveillance et de contrôle des installations industrielles. Dans ce contexte, de nombreuses approches de surveillance des systèmes industriels sont développées, en vue de la détection des défaillances, ainsi que pour le diagnostic de défauts, par les différentes communautés de recherche en automatique, productique et en intelligence artificielle [2].

Ce chapitre consiste à confirmer l'efficacité des techniques de Machine Learning à savoir TinyML et TensorFlow Lite sur un microcontrôleur conçus pour la surveillance intelligente en détectant un défaut de sous-vitesse ou de survitesse d'une turbine à gaz au niveau de la centrale RAS-DJINET qui s'avère être très complexe par son seuil qui varie en fonction de la température d'air d'entrée du compresseur axial.

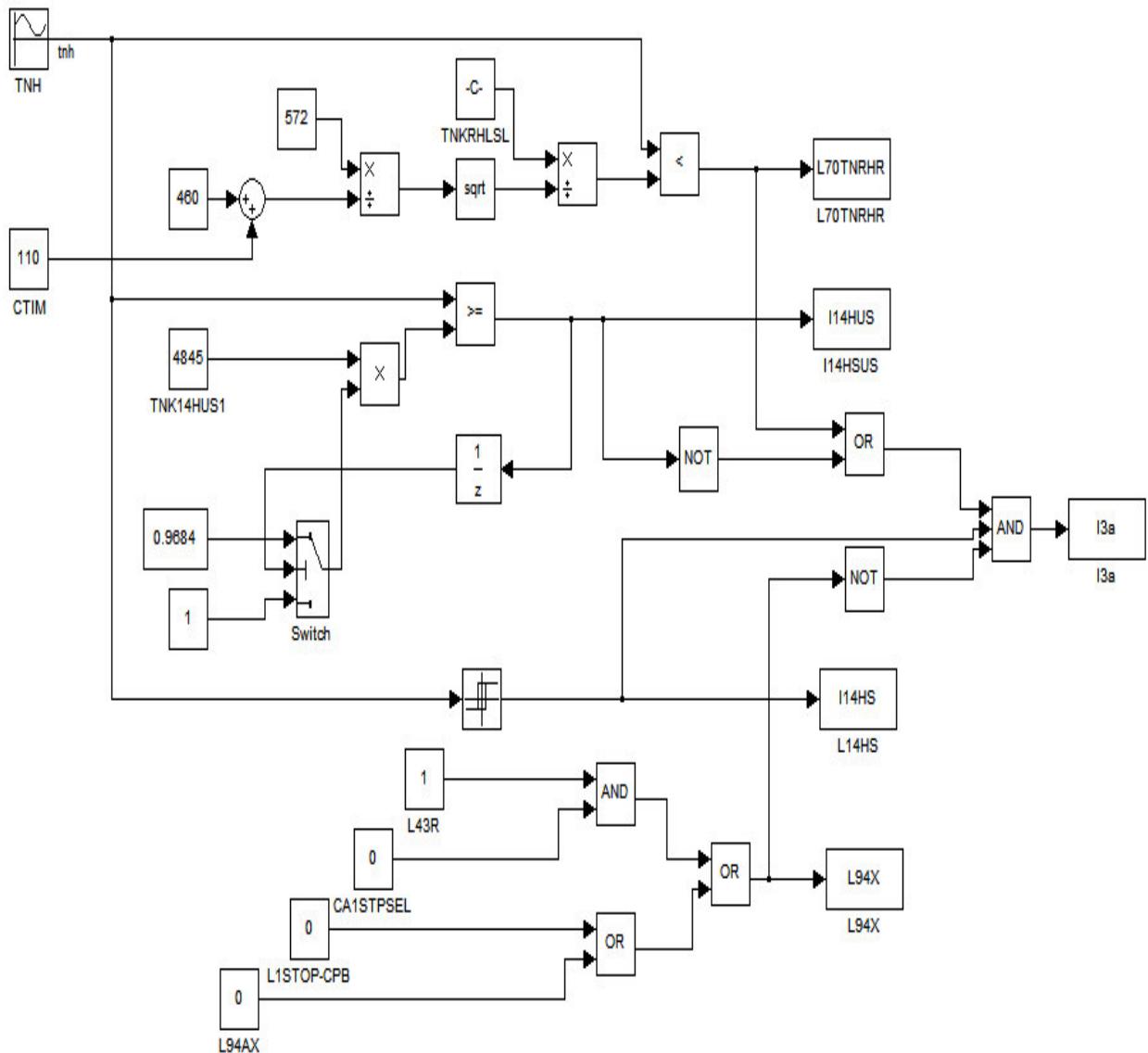
## V.2. Position du problème

Souvent, pour la surveillance de la turbine à gaz, le seuil du déclenchement d'alarme est fixé à une valeur constante [3]. Cependant, pour le cas de sous vitesse le seuil du déclanchement d'alarme est variable (en fonction de la température d'air d'entrée du compresseur CTIM), ce phénomène peut causer des fausses alarmes. Et c'est dans ce cas où les méthodes de surveillance conventionnelle trouvent ses limites.

La centrale de RAS-DJINET utilise la commande MARK V SPEEDTRONIC est un système de régulation dédié aux turbines à gaz ou à vapeur. Il contient un certain nombre de systèmes, de commandes, protections et séquentiel conçus pour assurer un fonctionnement fiable et sûr de la turbine à gaz. [6].

## CHAPITRE V : Réalisation d'un prototype de surveillance intelligente à l'aide de TinyML à base d'une carte Arduino.

La figure (V.1) montre une simulation d'alarme de défaut de sous-vitesse (L3A) sous MARK V SPEEDTRONIC converti en Matlab Simulink :



**Figure V.1. Alarme de sous-vitesse en simulink**

Le but de cette conversion de mark V à Matlab simulink est d'avoir un programme qu'on peut manipuler et simuler les cas où l'alarme de sous-vitesse est activée, et avoir une base de données d'apprentissage automatique ML.

L'objectif de notre travail est de proposer un système de surveillance intiligente à l'aide des technique ML et TinyML qui permet d'estimer le seuil de déclenchement d'alarme pour chaque point de fonctionnement, la détection d'un problème de sous vitesse en comparant le seuil avec la vitesse HP de la turbine en temps réel et le déclenchement d'alarme.

## CHAPITRE V : Réalisation d'un prototype de surveillance intelligente à l'aide de TinyML à base d'une carte Arduino.

Pour ce faire, nous avons réalisé les étapes suivante :

- Etape 1 : Construction d'une base de données d'apprentissage
- Etape 1 : Installation de MicroML
- Etape 2 : Configuration de l'environnement Python
- Etape 3 : Création d'une fonction pour gérer le jeu de données
- Etape 4 : Téléchargement des données
- Etape 5 : Entraînement du classificateur d'apprentissage automatique
- Etape 6 : Transférer le modèle en code C/C++ simple
- Etape 7 : Déploiement du classificateur de ML sur Arduino Uno

### IV.3. Application et résultats

#### IV.3.1. Construction d'une base de données d'apprentissage

Nous avons vu dans la section précédente, que le seuil d'alarme change en fonction de CTIM (la température à l'entrée du compresseur). En changeant cette dernière, nous avons pu identifier les données dont nous avons besoin (voir le tableau V.1) pour entraîner notre modèle.

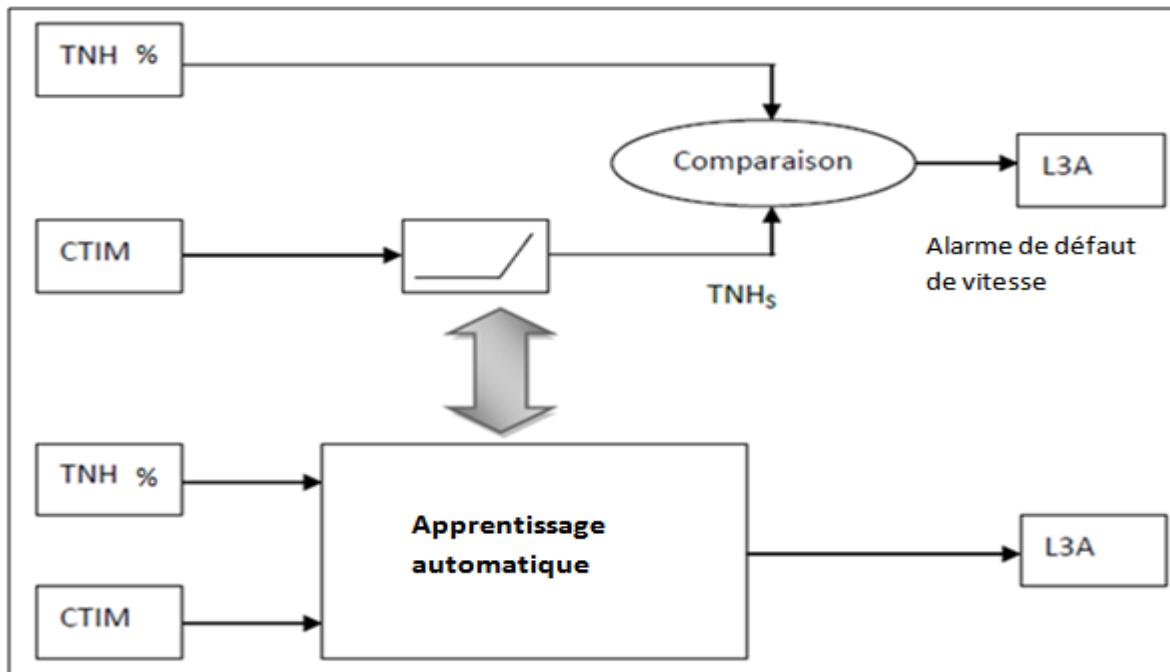


Figure V.2. Schéma synoptique pour l'apprentissage automatique de défaut de vitesse.

## CHAPITRE V : Réalisation d'un prototype de surveillance intelligente à l'aide de TinyML à base d'une carte Arduino.

---

**Tableau V.1. Tableau de base d'apprentissage pour le défaut de sous-vitesse**

Défaut de sous-Vitesse			
CTIM °F	Seuil (%)	TNH (%)	Alarme
-40	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
-20	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
0	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
20	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
40	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
60	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
80	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
100	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
105	91.998	[80 : 91.998 [ 91.998 : 110]	1 0
110	92.3375	[80 : 92.3375 [ 92.3375 : 110]	1 0
115	92.744	[80 : 92.744 [ 92.744 : 110]	1 0
120	93.1437	[80 : 93.1437 [ 93.1437 : 110]	1 0
125	93.544	[80 : 93.544 [ 93.544 : 110]	1 0
130	93.944	[80 : 93.944 [ 93.944 : 110]	1 0
135	94.3419	[80 : 94.3419 [ 94.3419 : 110]	1 0
140	94.7372	[80 : 94.7372 [ 94.7372 : 110]	1 0

### **V.3.2. Déploiement du modèle d'apprentissage sur un microcontrôleur ATmega328P d'une carte ArduinoUno**

#### **➤ Etape 1 : Installation de *MicroML* :**

*MicroML* est un projet visant à apporter des algorithmes d'apprentissage automatique aux microcontrôleurs. Il est né comme une alternative à *Tensorflow for Microcontrollers*, qui est uniquement dédié aux réseaux neuronaux artificiels (RNA).

Dans notre projet, nous trouverons des alternatives plus légères aux réseaux neuronaux pour exécuter l'inférence même sur des microcontrôleurs à 8 bits.

Pour installer *MicroML*, nous devons tout d'abord utiliser la bibliothèque « scikit-learn » de Python. Ensuite, nous devons installer le paquet *MicroML* [15].

Pour ce faire, nous allons utiliser « *Colabou Colaboratory* » qui est un service *cloud*, offert par Google (gratuit), basé sur *Jupyter Notebook* et qui est destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de ML directement dans le *cloud* sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.

```
[ ] pip install micromlgen

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting micromlgen
  Downloading micromlgen-1.1.28.tar.gz (12 kB)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from micromlgen) (2.11.3)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from jinja2->micromlgen) (2.0.1)
Building wheels for collected packages: micromlgen
  Building wheel for micromlgen (setup.py) ... done
    Created wheel for micromlgen: filename=micromlgen-1.1.28-py3-none-any.whl size=32172 sha256=d4eee05463c8cdffc8b9dd897ebe9
    Stored in directory: /root/.cache/pip/wheels/6f/98/43/0860b692737af546b9262a8b60ec3690c8bdc51c25e545ae09
Successfully built micromlgen
Installing collected packages: micromlgen
Successfully installed micromlgen-1.1.28
```

**Figure V.3. Un aperçu sur le code d'Installation de MicroML**

➤ **Etape 2 : Configuration de l'environnement Python :**

*Python* est un langage de programmation interprété, de haut niveau et d'usage général. Créé par « Guido van Rossum » et publié pour la première fois en 1991, la philosophie de conception de Python met l'accent sur la lisibilité du code avec son utilisation notable d'un espace blanc important. Ses constructions de langage et son approche orientée objet visent à aider les programmeurs à écrire un code clair et logique pour des projets de petite et de grande envergure.

**Pourquoi Python est-il le langage de programmation le mieux adapté pour l'apprentissage automatique ?**

Python est actuellement le langage de programmation le plus populaire pour la recherche et le développement en Machine Learning. On cite ci-dessous quelques de ces raisons :

- Python a plusieurs bibliothèques et frameworks qui peuvent être utilisés par les développeurs.
- Python bénéficie d'un support communautaire et d'entreprise
- Python est portable et extensible

Pour les besoins d'entraînement du modèle d'apprentissage, nous aurons besoin d'utiliser :

- La bibliothèque *NumPy* qui permet d'effectuer des calculs numériques avec Python et d'introduire une gestion facilitée des tableaux de nombres ;
- la bibliothèque *Scikit-Learn* de Python pour mettre en œuvre l'algorithme de la technique Random Forest (*en français, forêts aléatoires*) afin de résoudre les problèmes de classification ;
- le package *micromlgen* pour générer du code C simple pour les microcontrôleurs à partir des classificateurs *Scikit-Learnde* Python ;
- le module *glob* qui permet de rechercher tous les chemins correspondant à un motif particulier selon les règles utilisées par le shell Unix ;
- le module *os.path* qui est un sous-module du module OS en Python utilisé pour la manipulation commune des noms de chemin.

## CHAPITRE V : Réalisation d'un prototype de surveillance intelligente à l'aide de TinyML à base d'une carte Arduino.

Dans cette étape, nous allons mettre en place les dépendances requises pour notre *notebook Colab* :

```
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from micromlgen import port
from glob import glob
from os.path import basename
```

**Figure V.4. Un aperçu sur le code de configuration de l'environnement Python**

### ➤ Etape 3 : Création d'une fonction pour gérer le jeu de données :

La fonction suivante (Figure V. 5) permet de charger des jeux de données à partir des fichiers dédiés (.csv) pour des classes à classer. Elle reçoit en entrée le chemin du dossier contenant les fichiers et en sortie, elle retourne un jeu de donnée avec sa classe correspondante.

```
def load_features(folder):
    dataset = None
    classmap = {}
    for class_idx, filename in enumerate(glob('%s/*.csv' % folder)):
        class_name = basename(filename)[:-4]
        classmap[class_idx] = class_name
        samples = np.loadtxt(filename, dtype=float, delimiter = ';')
        labels = np.ones((len(samples), 1)) * class_idx
        samples = np.hstack((samples, labels))
        dataset = samples if dataset is None else np.vstack((dataset, samples))
    return dataset, classmap
```

**Figure V.5. Programme d'une fonction pour gérer le jeu de données**

➤ **Etape 4 : Chargement des données d'apprentissage**

Pour entraîner un modèle, on aura besoin de données. Pour cela, on doit tout d'abord créer une structure de dossiers, un dossier qui contiendra les données que nous avons collecté (base d'apprentissage, voir tableau V.1).

A l'intérieur de ce dossier, on crée un fichier dédié (.csv) pour chacune des classes que nous voulons classer, en mettant un échantillon sur chaque ligne. Par la suite, nous aurons qu'à exécuter la commande suivante (Figure V.6) pour charger le jeu de données :

The screenshot shows a Jupyter Notebook interface. On the left, there is a file tree with the following structure:

- {x}
- ..
- dataset
  - 0.csv
  - 1.csv
- sample\_data

On the right, there is a code cell containing the following Python code:

```
[ ] features, classmap = load_features('/content/dataset')
x, y = features[:, :-1], features[:, -1]
```

**Figure V.6. Commande de chargement des données d'apprentissage**

➤ **Etape 5 : Entraînement du modèle d'apprentissage :**

Les données de notre projet sont labellisées, c'est-à-dire, des données qui ont déjà été étiquetées avec le bon label (classe 0 et 1}). Par conséquent, nous pouvons aisément dire qu'on est face à un problème d'apprentissage supervisé. Cet apprentissage sur des données ayant déjà la « réponse correcte » permet de prédire par la suite le label de données nouvelles non étiquetées.

Ainsi, afin d'entrainer notre modèle, nous avons utilisé la technique « **Random Forest** » (voir chapitre IV). Cette technique est simplement constituée de nombreux arbres de décision réunis dans un schéma de vote. L'idée centrale est celle de la « sagesse des arbres », de sorte que si de nombreux arbres votent pour une classe donnée (après avoir été formés sur différents sous-ensembles de l'ensemble de formation), cette classe est probablement la vraie classe [14].

```
classifier = RandomForestClassifier(n_estimators=30, max_depth=10).fit(x, y)
```

**Figure V.7. Code d'entraînement du modèle d'apprentissage**

➤ **Etape 6 : Transfert du modèle en code C/C++ simple :**

Une fois que le modèle a été entraîné, nous devons par la suite le convertir en code C simple en utilisant le package *micromlgen*. A son tour, le code C généré sera enregistré dans un fichier appelé « *model.h* » pour l'importer dans un projet Arduino.

```
c_code = port(classifier, classmap=classmap)
print(c_code)
```

**Figure V.8. Fonction de transfert du modèle en code C/C++ simple**

➤ **Etape 7 : Déploiement du modèle sur le microcontrôleur ATmega328P d'une carte ArduinoUno :**

Avant de transférer le code source du projet Arduino au microcontrôleur de carte ArduinoUno, il doit être « compilé », c'est à dire le transformer en un fichier binaire (suite de 1 et de 0) compréhensible par le microcontrôleur.

Une fois la compilation terminée, si tout s'est bien déroulé correctement, on peut lancer le transfert de ce code au microcontrôleur.

Ensuite, une fois ce transfert terminé, le système de surveillance sera prêt pour être mis en œuvre.

## CHAPITRE V : Réalisation d'un prototype de surveillance intelligente à l'aide de TinyML à base d'une carte Arduino.

```
// inclure le modèle entraîné de l'étape 5
#include "model.h"

// spécification de la technique d'apprentissage utilisée
Eloquent::ML::Port::RandomForest classifier;

void classify() {
    float x_sample[] = { /* sample values */ };

    Serial.print("Predicted class: ");
    Serial.println(classifier.predictLabel(x_sample));
}
```

Figure V.9. Déploiement du modèle sur le microcontrôleur ATmega328P

### V.3.3. Le matériel utilisé dans notre projet

Tableau V.2. Les différents composants utilisés dans notre projet.

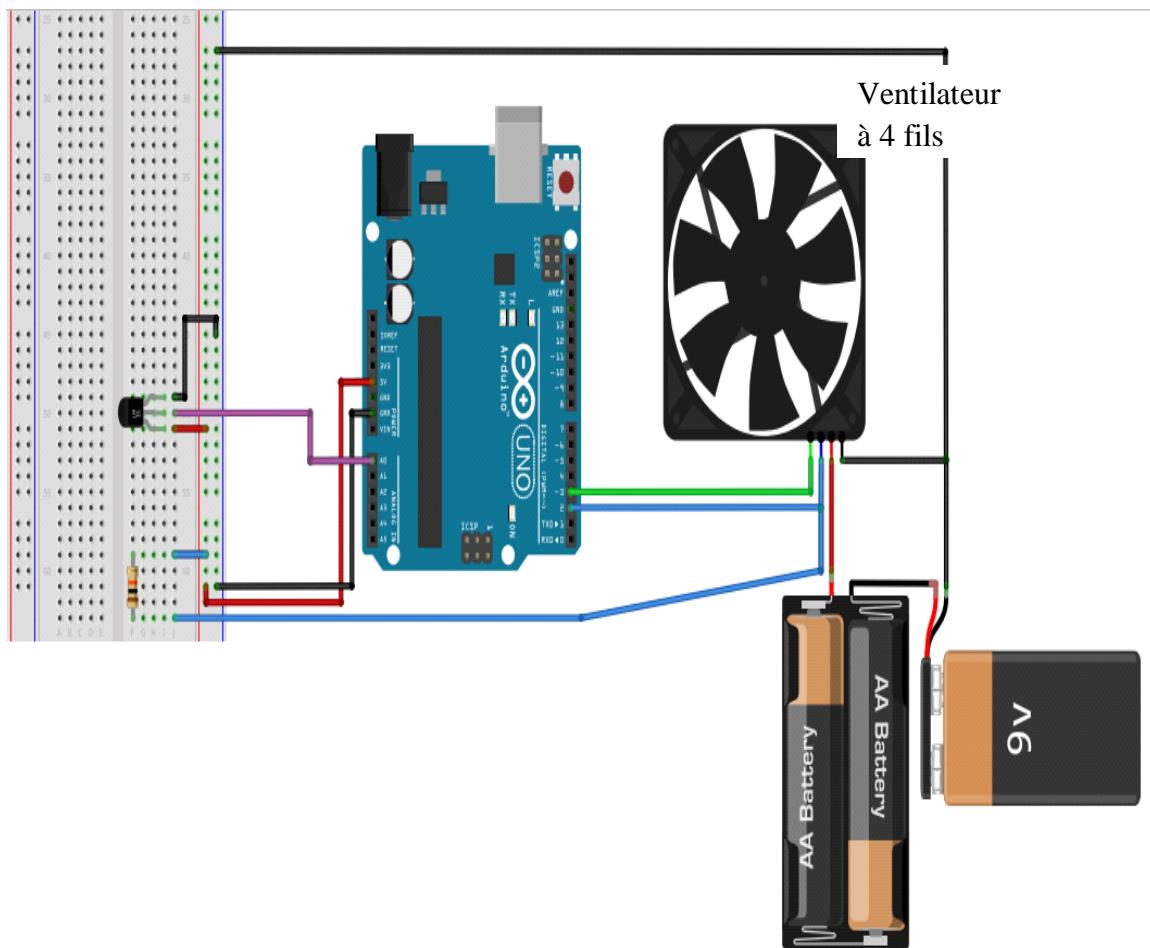
Composants	Rôle
Arduino UNO	traitement des données
Potentiomètre	varier les valeurs de la vitesse du ventilateur
capteur LM35	calculer la valeur de la température
ventilateur à 4 fils	joue le rôle de la turbine et récupérer la valeur de vitesse de rotation
Résistance R4 et R5 de 20K hom	Protection et connexion avec Arduino
Pile 12 V	alimenter le ventilateur

### V.3.4. Réalisation d'un Prototype (banc d'essai)

Nous avons réalisé le schéma global du système et nous avons connecté tous les éléments avec les câbles nécessaires, comme indique la figure V.10.

Le schéma général du banc d'essai réalisé est composé de deux parties essentielles :

- ✓ La *partie hardware* qui contient tous les composants utilisés à savoir :  
Ventilateur (qui joue le rôle de la turbine), les capteurs de température, piles d'alimentation et microcontrôleurs,...etc.
- ✓ La deuxième partie est la *partie software* elle contient deux programmes principaux : le programme Arduino et le déploiement du modèle d'apprentissage sur un microcontrôleur ATmega328P d'une carte ArduinoUno.



**Figure V.10. Schéma synoptique global d'Interconnexion et câblage avec L'arduino**

## CHAPITRE V : Réalisation d'un prototype de surveillance intelligente à l'aide de TinyML à base d'une carte Arduino.

La figure V.11. présente le banc d'essai de notre projet que nous avons réalisé.

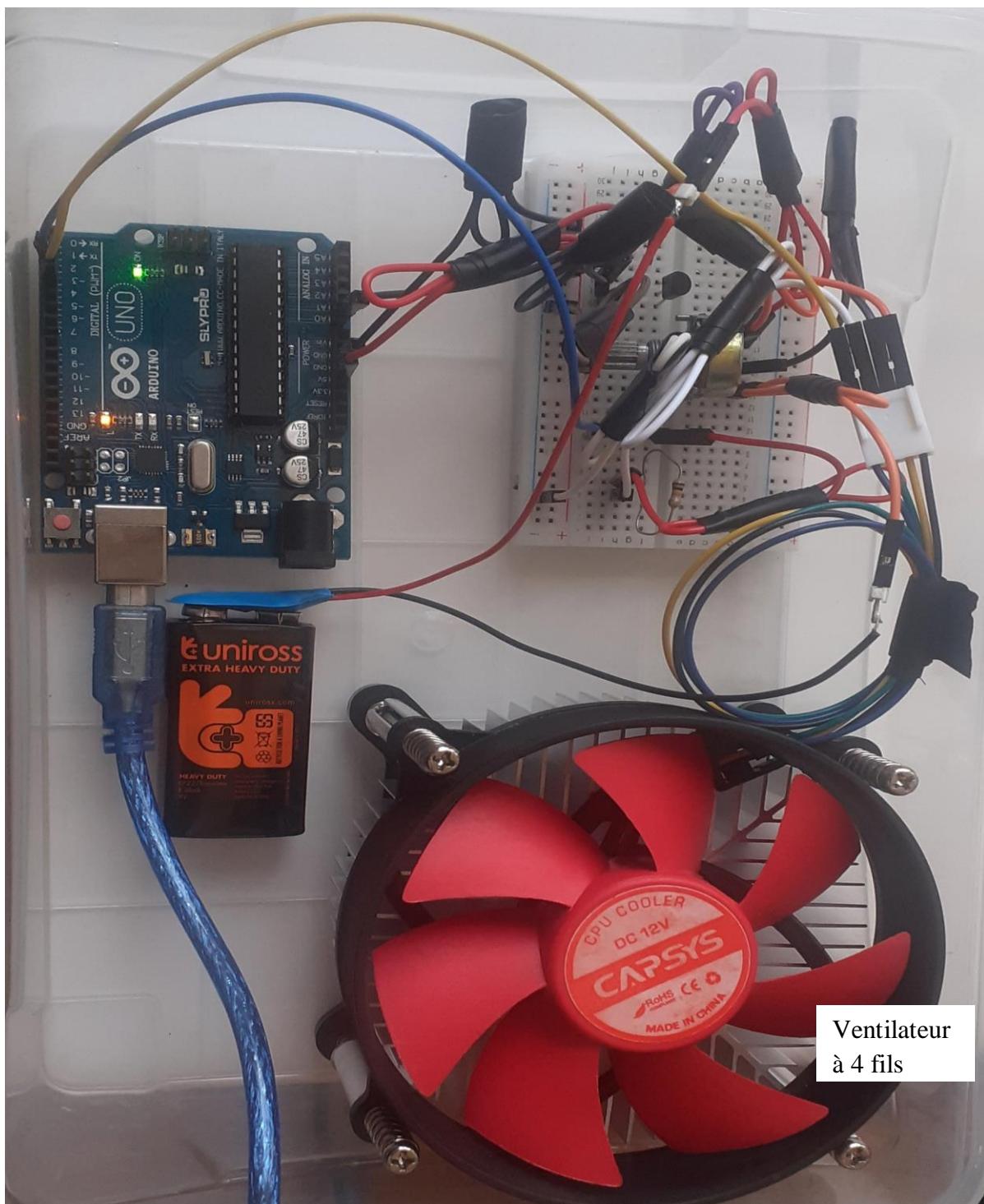


Figure V.11. Le banc d'essai de notre projet

Une fois le projet est construit, et que son apprentissage à atteint des performances satisfaisantes, on passe à l'étape de test en changeant la vitesse de la turbine et la température calculée en Fahrenheit à l'aide du logiciel ARDUINO sur l'ordinateur et noter le résultat.

## CHAPITRE V : Réalisation d'un prototype de surveillance intelligente à l'aide de TinyML à base d'une carte Arduino.

Les résultats de simulation par des valeurs qui n'appartiennent pas à la base d'apprentissage sont affichés sur le serial monitor du logiciel (Tableau V.3) :

```
Output Serial Monitor X

Message (Ctrl + Enter to send message to 'Arduino Uno' on 'COM3')

13:13:50.702 -> 68.07F , Speed = 91%, Prediction = 1
13:13:58.745 -> 68.07F , Speed = 90%, Prediction = 1
13:14:06.737 -> 68.07F , Speed = 91%, Prediction = 1
13:14:14.765 -> 67.19F , Speed = 90%, Prediction = 1
13:14:22.772 -> 68.07F , Speed = 91%, Prediction = 1
13:14:30.800 -> 68.07F , Speed = 91%, Prediction = 1
13:14:38.836 -> 68.07F , Speed = 94%, Prediction = 0
13:14:46.830 -> 74.23F , Speed = 93%, Prediction = 0
13:14:54.846 -> 76.87F , Speed = 94%, Prediction = 0
13:15:02.866 -> 78.63F , Speed = 94%, Prediction = 0
13:15:10.885 -> 78.63F , Speed = 93%, Prediction = 0
13:15:18.905 -> 79.51F , Speed = 96%, Prediction = 0
13:15:26.932 -> 80.39F , Speed = 94%, Prediction = 0
13:15:34.961 -> 80.39F , Speed = 93%, Prediction = 0
13:15:53.410 -> 81.27F , Speed = 95%, Prediction = 0
13:16:01.387 -> 82.15F , Speed = 87%, Prediction = 1
13:16:09.450 -> 82.15F , Speed = 97%, Prediction = 0
13:16:17.455 -> 82.15F , Speed = 91%, Prediction = 1
13:16:25.472 -> 82.15F , Speed = 92%, Prediction = 0
13:16:33.490 -> 83.03F , Speed = 95%, Prediction = 0
13:16:41.510 -> 82.15F , Speed = 95%, Prediction = 0
13:16:49.497 -> 83.03F , Speed = 95%, Prediction = 0
```

**Tableau V.3. Résultats de la simulation de notre prototype**

Les résultats de la simulation obtenus confirment l'efficacité de la technique d'apprentissage automatique utilisé à savoir TinyML.

### V.4. Conclusion

Dans ce chapitre nous avons réalisé un prototype de surveillance intelligente afin de détecter les défauts de vitesse d'une turbine (moteur) avec un déploiement du modèle d'apprentissage sur un microcontrôleur ATmega328P d'une carte ArduinoUno. Les résultats obtenus montrent l'efficacité et la grande précision de calcul.

## Conclusion générale

---

Le travail présenté dans ce mémoire porte sur une étude d'automatisations et supervision de système de commande de vitesse d'une turbine à gaz au niveau de la centrale thermique à cycle combiné de RAS-DJENET au sein de l'entreprise SONELGAZ, avec la réalisation d'un prototype de surveillance intelligente à l'aide des technique d'intelligences artificiels à savoir le Machine Learning et TinyML à base d'une carte Arduino.

Notre travail, nous a permis d'approfondir nos connaissances dans les domaines de contrôle, surveillance intelligente et diagnostic des systèmes industriels.

L'automatisation et la surveillance d'un système de protection de la turbine à vapeur sont des tâches qui demandent une bonne connaissance des aspects de l'automatique en général et des règles à respecter pour le fonctionnement de l'équipement en particulier. Nous avons essayé de prendre en considération tous ces aspects pour réaliser notre travail.

Après avoir présenté une description générale de la centrale thermique à cycle combiné de RAS-DJENET, dans laquelle nous avons montré son principe de fonctionnement et ses caractéristiques. Deux parties sont étudiées :

### ***Partie I : Automatisation et supervision de système de commande de vitesse d'une turbine à gaz.***

L'objectif de cette première partie de ce mémoire est de remplacer le système de la logique câblée de la commande de vitesse d'une turbine à gaz qui existe dans la centrale de Ras-Djanet par un système automatisé en utilisant un automate programmable industriel SIEMENS S7-300.

Pour cela, nous avons entamé notre travail en premier lieu par une présentation logigramme fonctionnel séquentiel de la turbine à gaz. Ensuite, nous avons réalisé le programme correspondant au logigramme et nous avons testé notre application par simulation en utilisant le logiciel STEP 7. Enfin, nous avons proposé et développé un système de supervision sous le logiciel Windows contrôle centre (WinCC) de Siemens pour garantir une conduite adéquate de l'interface Homme Machine et assurer le contrôle et la surveillance du système.

## Conclusion générale

---

***Partie II : Réalisation d'un Prototype de surveillance intelligente par un système embarqué à base d'un microcontrôleur ATmega328P d'une carte Arduino Uno.***

Dans cette deuxième partie, nous avons proposé les techniques d'apprentissage automatique (Machine Learning) à savoir TinyML et TensorFlow Lite sur un microcontrôleur pour la surveillance intelligente en détectant un défaut de sous-vitesse ou de survitesse d'une turbine à gaz au niveau de la centrale RAS-DJINET qui s'avère être très complexe par son seuil qui varie en fonction de la température d'air d'entrée du compresseur axial.

Pour cela, nous avons réalisé un Prototype de surveillance par un système embarqué à base d'un microcontrôleur ATmega328P d'une carte Arduino Uno et nous avons connecté tous les éléments avec les câbles nécessaires. Ce prototype est composé de deux parties essentielles :

- ✓ La partie hardware qui contient tous les composants utilisés à savoir :  
Ventilateur (qui joue le rôle de la turbine), les capteurs de température, piles d'alimentation et microcontrôleurs,...etc.
- ✓ La deuxième partie est la partie software elle contient deux programmes principaux : le programme Arduino et le déploiement du modèle d'apprentissage sur un microcontrôleur ATmega328P d'une carte ArduinoUno.

Les résultats de test ont montré que la technique d'apprentissage utilisé était efficace pour la surveillance intelligente des systèmes.

Enfin, nous espérons que l'étude effectuée présente un intérêt pour l'entreprise, et apporte un éclairage aux étudiants qui auront à préparer leur projet de fin d'études ultérieurement.

# Références bibliographique

---

## ➤ Documents de l'entreprise :

- [1] Rapport SONALGEZ "centrale à Cycle Combiné de RAS DJINET 2, juin 2021.
- [2] Documentation de la centrale à cycle combiné de 'RAS-DJINET' ( Basic - 06 Aspectsgénéraux de la centrale électriques-Composants mécanique majeurs, 2020)
- [3] Documentation interne de la centrale RAS-DJENET « **Système de Régulation, Contrôle et Protection** », Service de formation professionnelle (N°, d'enregistrement : **5524**).
- [4] Catalogue Siemens ST 70 ; Rapport SONALGEZ 2018.
- [5] Documentation interne de la centrale RAS-DJENET « Description générale-Turbine a gazSGT5 4000F et ses composants », Basic 12 & 13, 2021.
- [6] Documentation de l'entreprise SONALGAZ intitulé « Société Algérienne de Production de l'Electricité » unité de production CC, 2020.

## ➤ Ouvrages et articles :

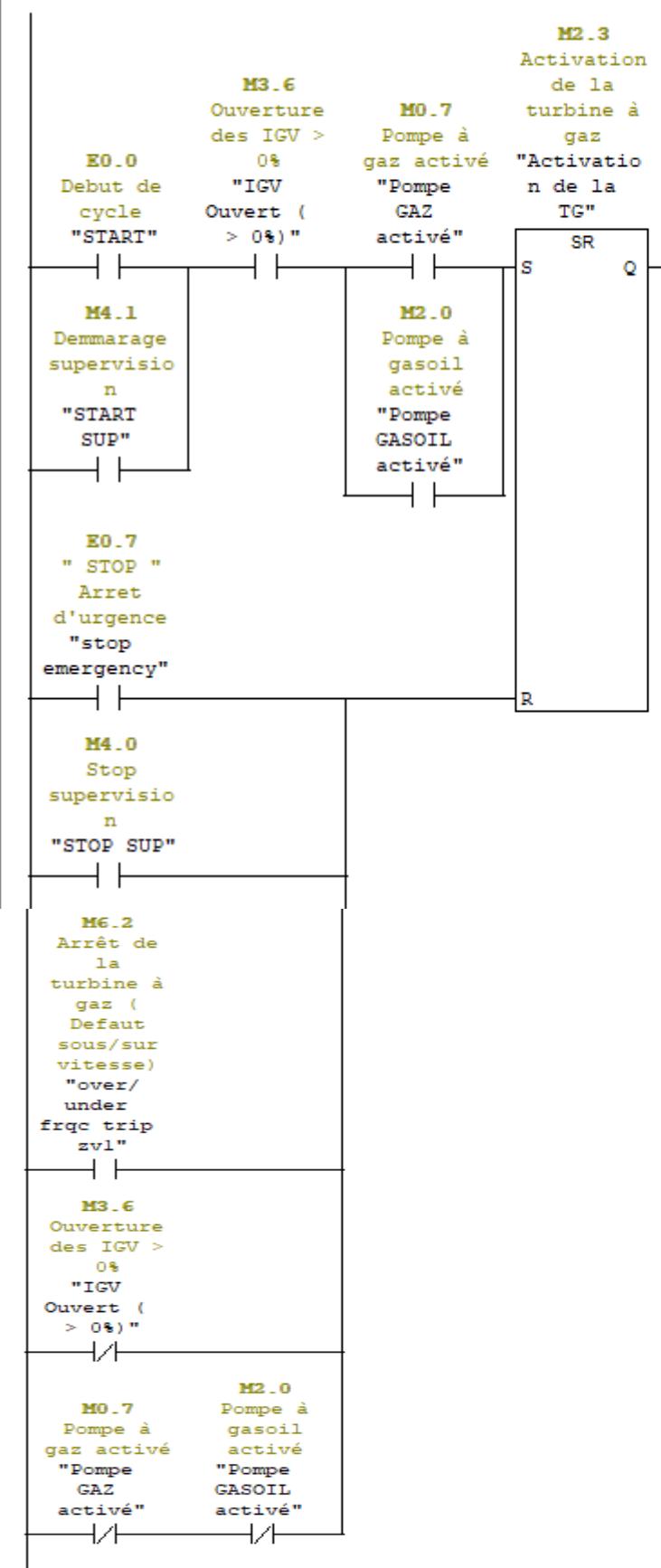
- [7] **G. MICHEL**, « *Les A.P.I architecteur et application des automates programmables industriels* », Edition DUNOD ,1987.
- [8] **J-M. BLEUX J-L. FANCHON** « Automatisme industrielle » Edition Nathan 2001
- [9] **C. MERLAUD. JACQUES PERRIN. JEAN-PAUL TRICHARD** « Automatique Informatique industrielle » Edition DUNOD ,1995.
- [10] **N. KANDI** «Architecture Matérielle des API », formation IAP (ALGERIANPETROLEUM INSTITUTE), 2015
- [11] **A. Wendbenedo**, « *Application des algorithmes d'apprentissage automatique pour la détection de défauts de roulements sur les machines tournantes dans le cadre de l'industrie* 4.0 », thèse du grade de maître ès sciences appliquée en ingénierie, profil recherche, université du Québec à Chicoutimi, 2020, 69p.
- [12] **P. Jahnke**, « *Machine Learning Approaches for Failure Type Detection and Predictive Maintenance* », June 19, 2015.
- [13] **C-A. Azencott**, « *Introduction au Machine Learning* », édition 1, 2015, 180 p.
- [14] **C. Faure, L. Miclet** ; « *Intelligence artificielle et reconnaissance des formes* », Technique de l'ingénieur H1900 , Juin 1990.
- [15] **G. M. IODICE**, *TinyML Cookbook : Combine artificial intelligence and ultra-low-power embedded devices to make the world smarter*. S.l.: Packt Publishing LTD, 2022.

# **ANNEXES**

## ANNEXES

---

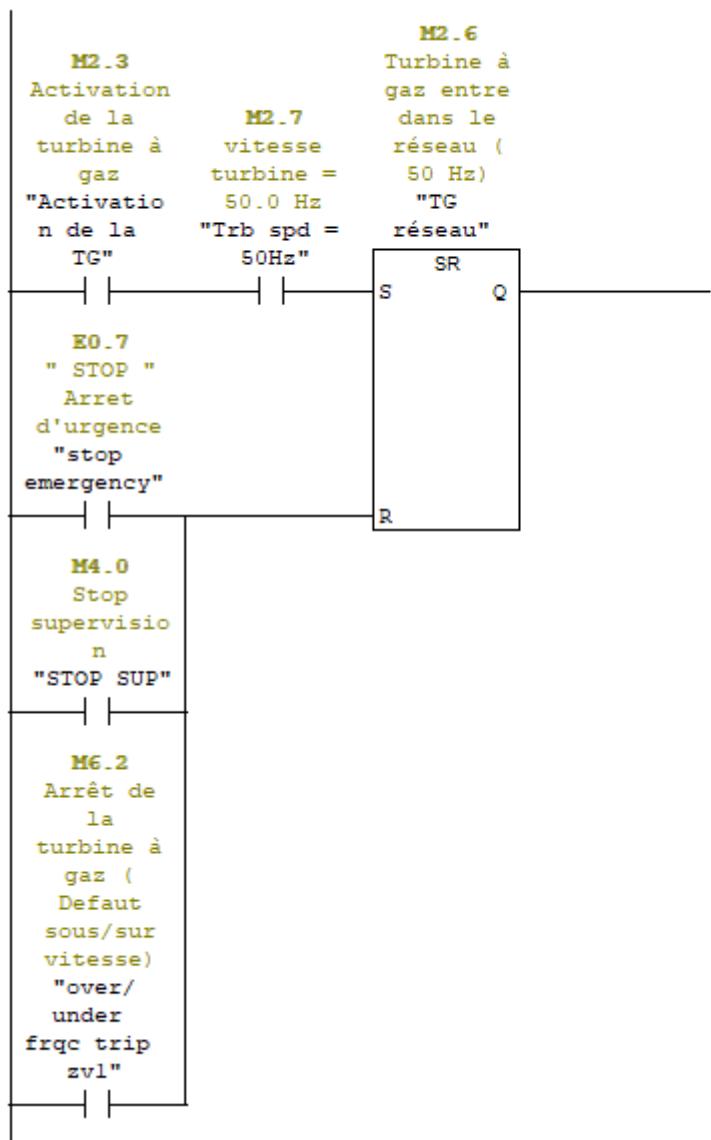
□ Réseau 1 : Réseau d'activation de la turbine à gaz



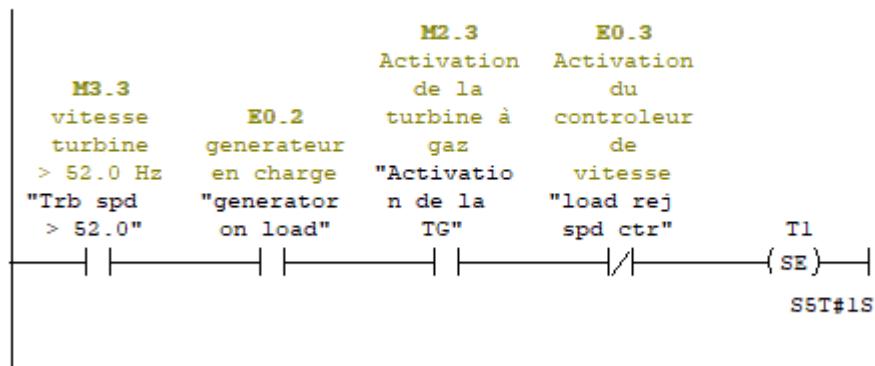
## ANNEXES

---

□ Réseau 2 : Turbine à gaz prête à intégrer le réseau

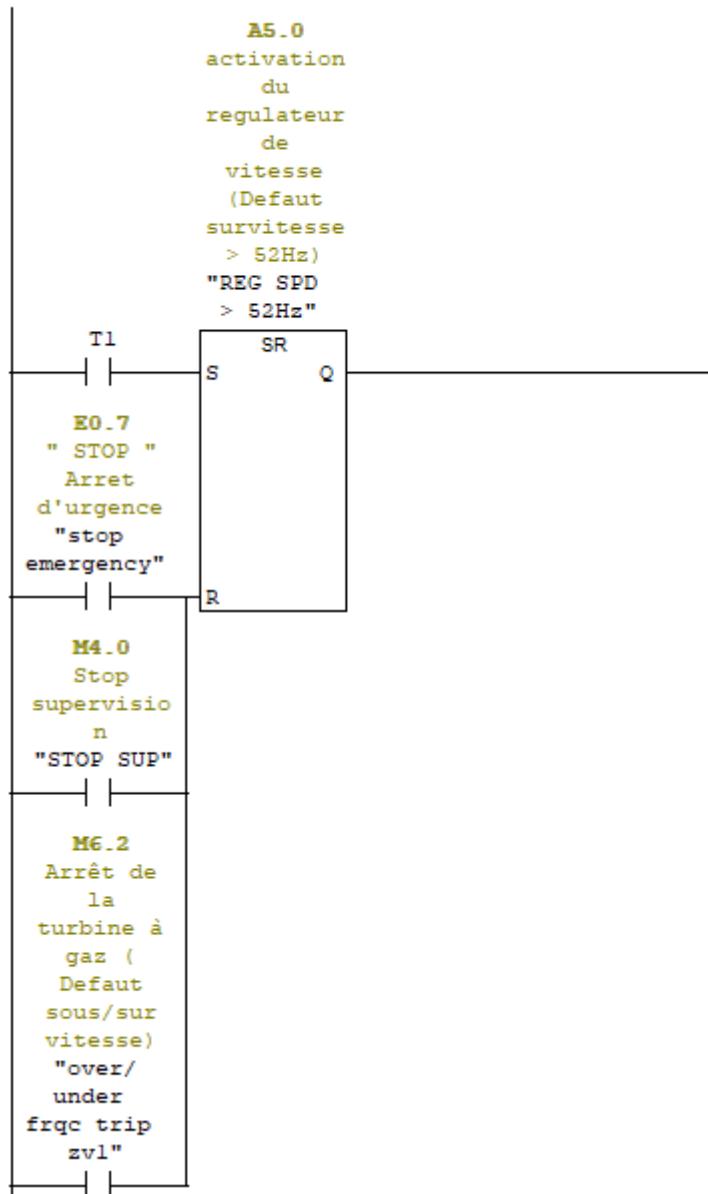


□ Réseau 3 : Réseaux d'activation de la temporisation T1

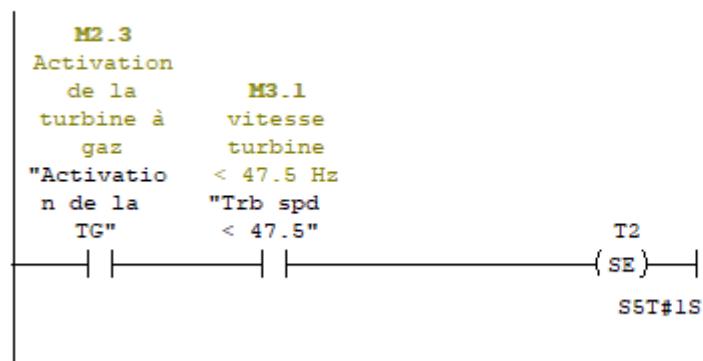


## ANNEXES

### ■ Réseau 4 : Réseau d'activation du régulateur de vitesse (> 52 Hz)



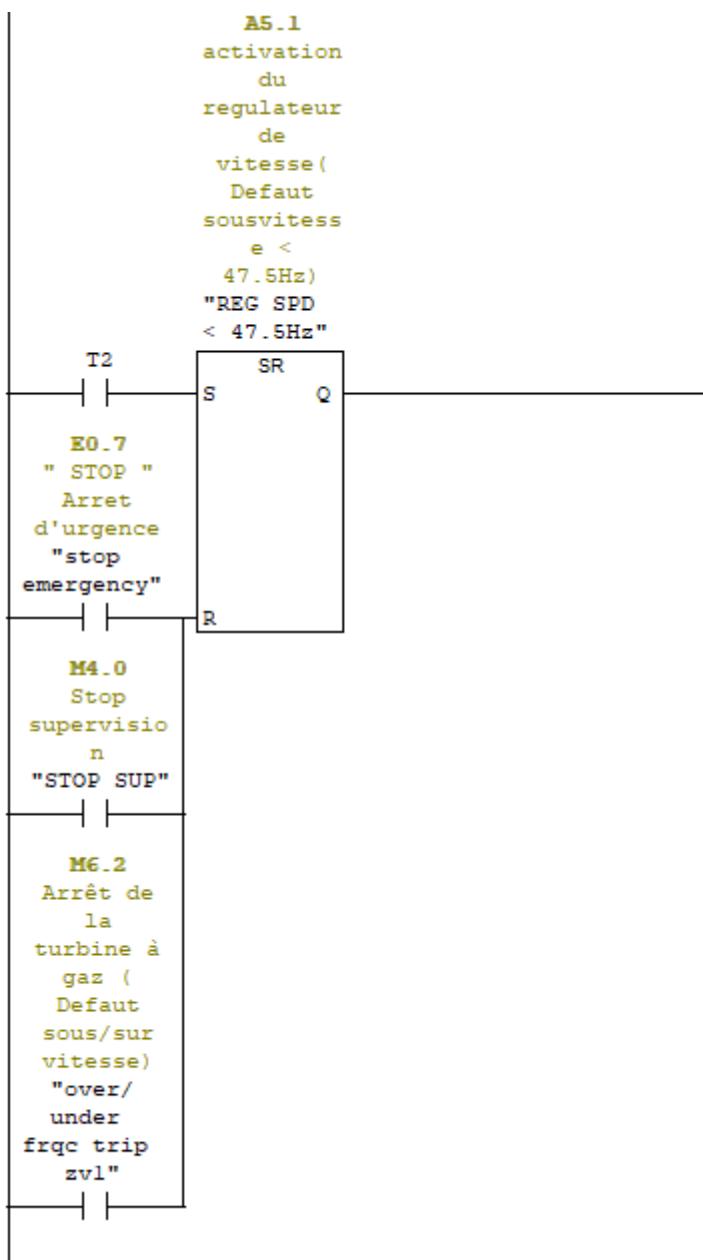
### ■ Réseau 5 : Réseaux d'activation de la temporisation T2



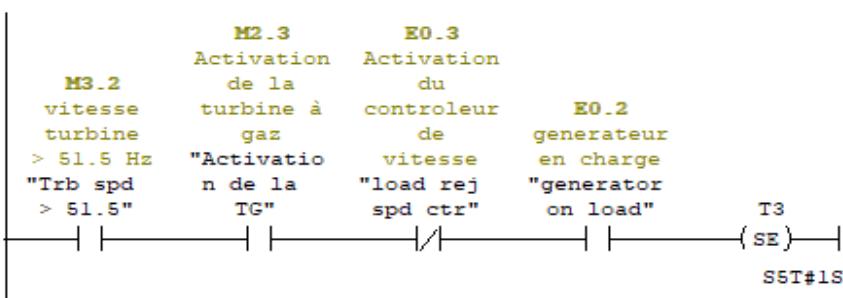
## ANNEXES

---

□ Réseau 6 : Activation du régulateur de vitesse( < 47.5 Hz )



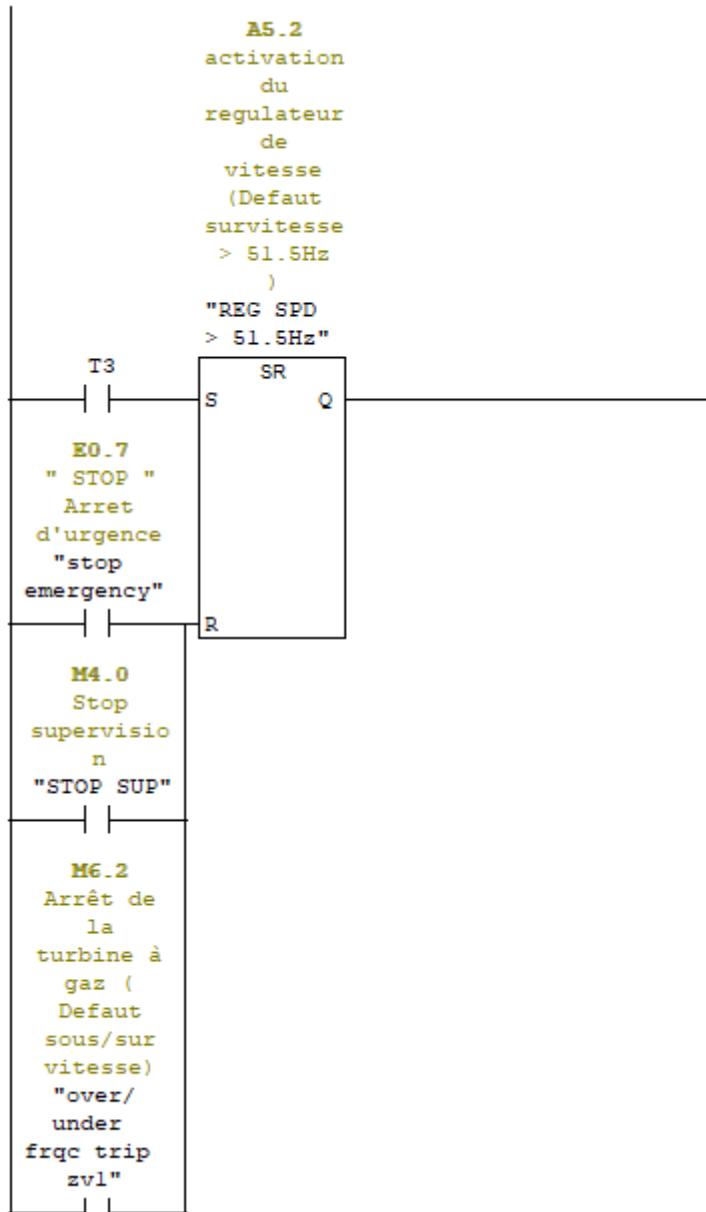
□ Réseau 7 : Réseaux d'activation de la temporisation T3



## ANNEXES

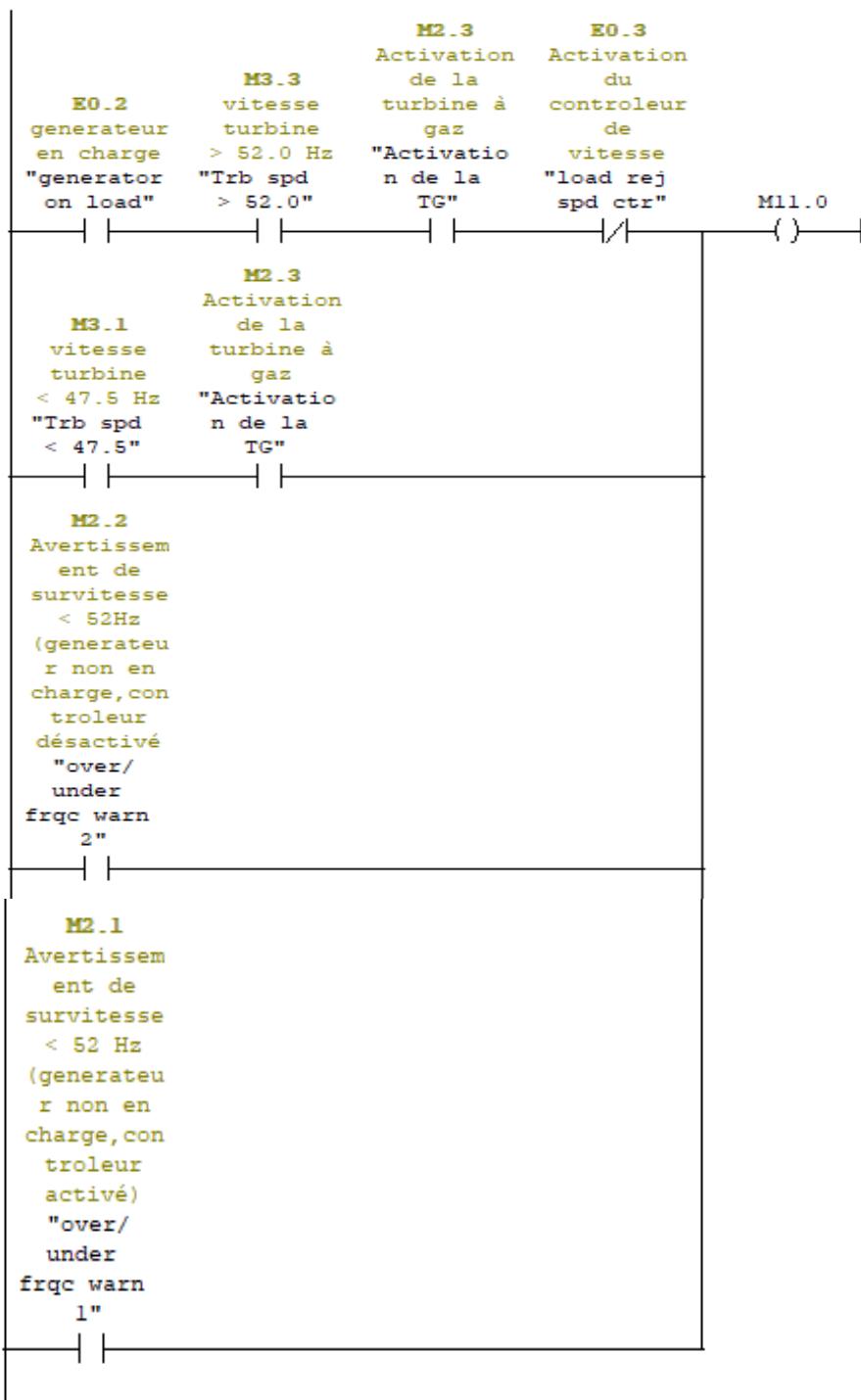
---

□ Réseau 8 : Réseau d'activation du régulateur de vitesse ( $> 51.5 \text{ Hz}$ )



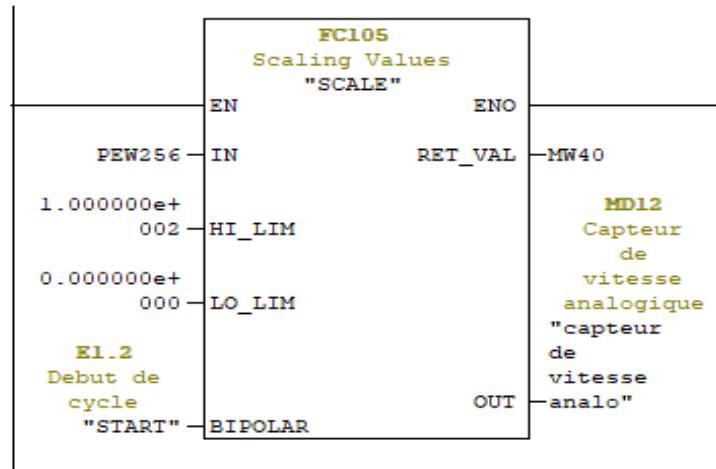
## ANNEXES

■ Réseau 9 : Alarme défaut de vitesse (Activation du régulateur de vitesse)

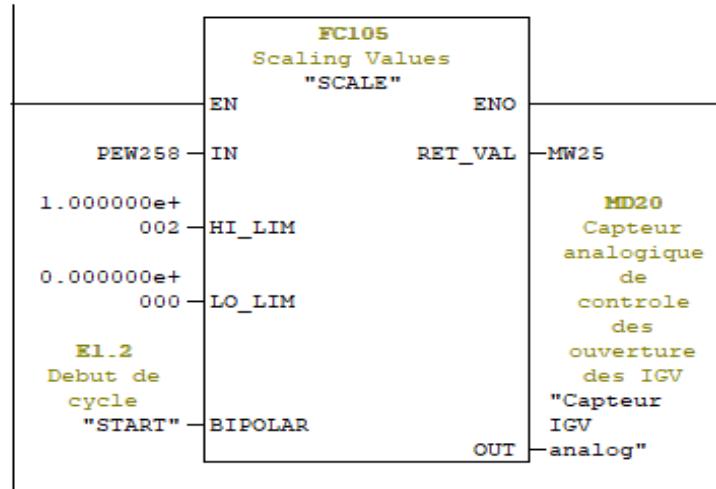


## ANNEXES

### □ Réseau 32 : Capteur de vitesse analogique



### □ Réseau 33 : Capteur analogique de contrôle d'ouverture des IGV analogique



## ANNEXES

---

■ Réseau 47 : Alarme de survitesse < 52Hz (GNEC , contrôleur activé)



■ Réseau 48 : Alarme de survitesse < 52Hz (GNEC , contrôleur désactivé)



## Résumé

On s'intéresse dans ce travail à étude de l'automatisations et supervision de système de commande de vitesse d'une turbine à gaz au niveau de la centrale thermique à cycle combiné de RAS-DJENET de l'entreprise SONELGAZ, avec la réalisation d'un prototype de surveillance intelligente à l'aide des technique d'intelligences artificiels à savoir le Machine Learning, TinyML et *Tensorflow for Microcontrollers* base d'une carte Arduino.

Ce prototype est composé essentiellement de deux parties à savoir :

- ✓ La *partie hardware* qui contient tous les composants utilisés à savoir :  
Ventilateur (qui joue le rôle de la turbine à gaz), capteurs, microcontrôleurs,...etc.
- ✓ La deuxième partie est la *partie software* elle contient deux programmes principaux : le programme Arduino et le déploiement du modèle d'apprentissage sur un microcontrôleur.

Les résultats de test de notre prototype ont montré que la technique d'apprentissage utilisé était efficace pour la surveillance intelligente des systèmes.

**Mots clés :** Automatisation, Supervision, Turbine à gaz, Surveillance intelligente, Machine Learning.

---

## Abstract

We are interested in this work in the study of the automation and supervision of the speed control system of a gas turbine at the level of the combined cycle thermal power plant of RAS-DJENET of the company SONELGAZ, with the realization of 'a prototype of intelligent monitoring using artificial intelligence techniques namely Machine Learning, TinyML and Tensorflow for Microcontrollers based on an Arduino board.

This prototype is essentially composed of two parts, namely:

- ✓ The hardware part which contains all the components used, namely:  
Fan (which acts as the gas turbine), temperature and speed sensors, power batteries and microcontrollers, etc.
- ✓ The second part is the software part it contains two main programs: the Arduino program and the deployment of the learning model on an ATmega328P microcontroller of an ArduinoUno board.

The test results of our prototype showed that the learning technique used was effective for intelligent system monitoring.

**Keywords:** Automation, Supervision, Gas turbine, Intelligent monitoring, Machine learning.

