

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITÉ HADJ LAKHDAR « BATNA »
FACULTÉ DE TECHNOLOGIE
DÉPARTEMENT DE GÉNIE INDUSTRIEL

MEMOIRE DE MAGISTER

PRESENTE AU

LABORATOIRE D'AUTOMATIQUE ET PRODUCTIQUE

En vue de l'obtention de diplôme de

MAGISTER

Spécialité : Génie Industriel et Productique

Par

BENAICHA SONIA

Ingénieur d'état en Informatique

Thème

LE PRONOSTIC INDUSTRIEL PAR LES SYSTÈMES

NEURO-FLOUS :

APPLICATION À UN SYSTÈME DE PRODUCTION

Directrice de Thèse : *Dr. L. Hayet MOUSS*

Devant le JURY :

Chaabane Hamouda

Okba Kazar

Hayet Mouss

Samir Abdelhamid

Rafik Mehdaoui

Said Achi

MC-A

MC-A

MC-A

MC-A

MAT-A

Ingénieur

Université de Batna

Université de Biskra

Université de Batna

Université de Batna

Centre Univ-Khenchela

Chef de Zone SCIMAT

Président

Examineur

Rapporteur

Examineur

Co-Rapporteur

Invité

ANNÉE UNIVERSITAIRE : 2011-2012

REMERCIEMENTS

Nos remerciements vont tout premièrement à dieu tout puissant pour la volonté, la santé et la patience, qu'il nous a données durant ces années d'étude afin que nous puissions arriver à ce stade.

Mes remerciements s'adressent ensuite aux membres du jury qui me fait l'honneur de participer à ma soutenance.

Je remercie sincèrement Dr. L. Hayet Mouss pour avoir accepté d'évaluer mes travaux en qualité de rapporteur et aussi pour son encadrement et ses précieux conseils.

Mes remerciements s'adressent également à Dr. Chaabane Hamouda, Dr. Okba Kazar et Dr Samir Abdelhamid pour leur participation au jury en tant qu'examineurs. Aussi, Mr Mahdaoui Rafik pour avoir accepté d'évaluer mes travaux en qualité de Co-encadreur. A Mr Said Achi pour ses explications et ses conseils.

Je remercie également, et sans exception, l'ensemble des opérateurs et des ingénieurs de la SCIMAT de Batna.

*Je ne peux terminer sans avoir une pensée pour ma famille. Sans elle, je n'aurais jamais pu arriver jusque là et atteindre mes objectifs. Un grand merci à mes parents (**Khadija** et **Meziane**) pour leur soutien indéterminé.*

Je remercie énormément mes amies, pour l'encouragement incroyable de leur part.

Un grand merci pour tous et cette thèse leurs est dédiée. Tous ce qu'ont loin de moi, mais m'aident avec les prières et les souhaits de succès dans cette thèse et dans toute ma vie.

Dédicaces

A mes parents, à qui je dois tous et qui ont tout donné, que Dieu les guérisse,

A mon frère,

A mes sœurs,

A mes professeurs,

A toute ma famille,

A mes amies pour ses encouragements,

A mon amie Nora qui m'a beaucoup soutenu,

A ceux et celles qui m'ont aidé

SONIA.

SOMMAIRE

Liste des figures	- 6 -
Liste des tableaux	- 9 -
Abréviation.....	- 10 -
Introduction Générale.....	-11-
Chapitre 01: Diagnostic industriel d'un système de production	-14-
1.1 Introduction	- 15 -
1.2 Système de production	- 15 -
1.2.1 Définition	- 15 -
1.2.2 Modes de fonctionnement d'un système	- 15 -
1.3 Théorie générale du diagnostic et de la surveillance	- 16 -
1.4 Méthodes de diagnostic	- 18 -
1.5 Diagnostic par Reconnaissance des Formes	- 19 -
1.5.1 Phase d'analyse	- 20 -
1.5.2 Phase de choix d'une méthode de décision	- 20 -
1.5.3 Phase d'exploitation	- 20 -
1.6 Du Diagnostic vers Pronostic	- 20 -
1.7 Le Pronostic industriel	- 21 -
1.7.1 Concept de pronostic	- 21 -
1.7.2 Classification des approches de pronostic	- 22 -
1.7.2.1 Pronostic basé sur le modèle physique	- 23 -
1.7.2.2 Pronostic guidé par les données	- 24 -
1.7.2.3 Pronostic basé sur l'expérience	- 25 -
1.8 Conclusion	- 25 -
Chapitre 02 : Outils de la Reconnaissance des Formes pour le diagnostic industriel	- 26 -
2.1 Introduction	- 27 -
2.2 La logique floue	- 27 -
2.2.1 Introduction	- 27 -
2.2.2 Définitions	- 27 -
2.2.3 Structure d'une commande Floue.....	- 27 -
2.2.3.1 Bases de règles et définitions	- 28 -
2.2.3.2 Interface de fuzzification	-28-
2.2.3.3 Mécanismes d'inférence	- 29 -
2.2.3.4 Interface de défuzzification	- 30 -
2.3 Les réseaux de neurones	- 30 -
2.3.1 Généralités.....	- 30 -

2.3.2 Les architectures neuronales	- 31 -
2.3.2.1 Les réseaux de neurones non bouclés « feedforward »	- 31 -
2.3.2.2 Les réseaux de neurones bouclés (récurrents)	- 32 -
2.3.3 Les types d'apprentissage des réseaux de neurones	- 32 -
2.3.3.1 Apprentissage non supervisé	- 32 -
2.3.3.2 Apprentissage supervisé	- 32 -
2.3.4 Les réseaux de neurones les plus utilisés	- 33 -
2.3.4.1 Perceptron Multicouches (PMC)	- 33 -
2.3.4.2 Les Réseau de neurones à Fonctions de Base Radiales	- 34 -
2.4 Réseaux Temporels	- 35 -
2.4.1 Représentation du temps	- 35 -
2.4.1.1 Représentation externe (spatiale) du temps	- 35 -
2.4.1.2 Représentation dynamique du temps	- 36 -
2.4.2 Modèle de Jordan	- 36 -
2.4.2.1 Architecture	- 37 -
2.4.2.2 Comportement	- 37 -
2.4.2.3 Algorithme d'apprentissage	- 37 -
2.4.3 Modèle d'Elman	- 37 -
2.4.3.1 Architecture	- 37 -
2.4.3.2 Comportement	- 38 -
2.4.3.3 Algorithme d'apprentissage	- 38 -
2.5 Réseaux Neuro-Flous	- 38 -
2.5.1 Introduction	- 38 -
2.5.2 Objectif	- 38 -
2.5.3 Définitions	- 39 -
2.5.4 Le système neuro-flou hybride	- 39 -
2.5.5 Architectures neuro-floues	- 40 -
2.5 Conclusion	- 42 -
Chapitre 03 : Analyse et spécification du nouveau système	- 43 -
3.1 Introduction	- 44 -
3.2 Etude de l'existant	- 44 -
3.2.1 Présentation de l'entreprise	- 44 -
3.2.1.1 Identité	- 44 -
3.2.1.2 Situation Géographique	- 44 -
3.2.2 Etapes de Fabrication du ciment	- 44 -
3.2.2.1 Matières premières	- 46 -
3.2.2.2 Exploitation des carrières	- 46 -
3.2.2.3 Stockage et reprise des matières premières	- 46 -
3.2.2.4 Broyage cru	- 46 -

3.2.2.5 Cuisson	- 47 -
3.2.2.6 Broyage Clinker	- 47 -
3.2.2.7 Expéditions	- 47 -
3.2.3 Système de contrôle, contexte existant.....	- 48 -
3.2.3.1 Système FLS-ACE-625	- 48 -
3.2.3.2 Système de régulation FLS 470.....	- 48 -
3.2.3.3 Système de contrôle FLS-325	- 48 -
3.2.3.4 Système de conduite S.D.R	- 48 -
3.2.3.5 CEM-SCANNER (Système de Contrôle des Réfractaires).....	- 48 -
3.2.3.6 Système de Contrôle et de Conduite (QCX)	- 49 -
3.2.3.7 SDR/FUZZY LOGIC	- 49 -
3.3 Cahier des Charges Fonctionnel	- 50 -
3.3.1 Définition	- 50 -
3.3.2 Description du projet	- 50 -
3.3.3 Positionnement	- 50 -
3.3.4 Caractéristiques de l'installation	- 51 -
3.3.5 Composition chimique des produits	- 52 -
3.3.6 Expression des besoins	- 52 -
3.3.7 Exigences fonctionnelles.....	- 52 -
3.3.8 Exigences non fonctionnelles	- 53 -
3.3.9 Présentation de la Conduite Future du Four	- 53 -
3.4 Conclusion.....	- 55 -
Chapitre 04: Environnement de travail.....	-56-
4.1 Introduction	- 57 -
4.2 Automate Programmable Industriel (API)	- 57 -
4.2.1 Définition	- 57 -
4.2.2 Constitution de l'automate	- 57 -
4.2.3 Automate Programmable Industriel S7-400.....	- 57 -
4.2.4 Caractéristiques	- 58 -
4.3 Progiciel PCS7	- 58 -
4.3.1 Description du projet PCS7	- 58 -
4.3.1.1 Qu'est ce qu'un projet PCS 7 ?	- 58 -
4.3.1.2 Applications PCS 7	- 59 -
4.3.2 SIMATIC Manager	- 59 -
4.3.2.1 Qu'est-ce que SIMATIC Manager ?.....	- 59 -
4.3.2.2 Systèmes d'automatisation	- 59 -
4.3.3 Programmer avec STEP 7	- 59 -
4.3.3.1 Qu'est-ce que STEP 7 ?	- 59 -

4.3.3.2 Applications disponibles	- 60 -
4.3.3.3 Types de blocs	- 60 -
4.3.3.4 Mnémoniques STEP7	- 60 -
4.3.4 S7-PLCSIM	- 61 -
4.3.5 WinCC (Windows Control Center)	- 61 -
4.4 Add-Ons PCS7	- 62 -
4.4.1 FuzzyControl++	- 62 -
4.4.1.1 Qu'est ce que Fuzzy Control++?	- 62 -
4.4.1.2 Caractéristiques Fuzzy Control++	- 62 -
4.4.1.3 Utilisation Fuzzy Control++	- 63 -
4.4.2 NeuroSystems	- 63 -
4.4.2.1 Qu'est ce que NeuroSystems ?	- 63 -
4.4.2.2 Caractéristiques NeuroSystems	- 63 -
4.4.2.3 Utilisation NeuroSystems	- 64 -
4.4.3 Fuzzy-NeuroSystems	- 64 -
4.4.4 GRAFCET	- 64 -
4.5 AMDEC	- 65 -
4.6 MATLAB	- 65 -
4.5.1 Qu'est ce que NeuroSystems ?	- 65 -
4.5.2 Caractéristiques Matlab	- 65 -
4.5.3 Utilisation Matlab	- 65 -
4.5.4 Principales fonctionnalités	- 66 -
4.5.5 Modules complémentaires à Matlab	- 66 -
4.7 Conclusion	- 66 -
Chapitre 05: Vers une approche neuro-floue pour le pronostic industriel	- 67 -
5.1 Introduction	- 68 -
5.2 Atelier de Clinkérisation	- 68 -
5.2.1 Conception du four rotatif	- 68 -
5.2.2 Zones de réaction du Four Rotatif	- 69 -
5.2.3 Problématique dans la conduite du four	- 69 -
5.3 Pronostic de l'atelier	- 69 -
5.3.1 Démarche proposée	- 70 -
5.3.2 Pronostic industriel basé sur la <i>RdF Neuro-floue</i>	- 70 -
5.3.3 Phase d'analyse	- 71 -
5.3.2.1 <i>AMDEC</i>	- 71 -
5.3.2.2 Contexte de l'étude	- 74 -
5.3.3.3 Architecture du système Neuro-Flou	- 74 -
5.3.4 Phase de choix	- 75 -

5.3.5 Phase d'exploitation	- 77 -
5.4 Conclusion	- 79 -
Chapitre 06: Implémentation NEFPROG	-80-
6.1 Introduction	-81-
6.2 Composants du programme	-81-
6.2.1 Station Opérateur	-81-
6.2.2 Table Mnémonique.....	-82-
6.2.3 Les blocs.....	-83-
6.2.4 Besoin de dialogue entre les opérateurs et la partie commande	-84-
6.3 Processus de clinkérisation	-85-
6.3.1 Alimentation Four	-85-
6.3.2 Four	-85-
6.3.2.1 Conditions de mise en marche.....	-86-
6.3.2.2 Séquence de démarrage	-88-
6.3.2.3 Mise en Arrêt (Refroidissement du four)	-89-
6.3.2.4 Les alarmes.....	-89-
6.3.3 Boucles de régulation	-90-
6.3.3.1 Régulation classique.....	-90-
6.3.3.2 Régulation Floue	-91-
6.3.3.3 Régulation neuro floue	-94-
6.3.3.4 Régulation par ANFIS.....	-98-
6.4 Présentation NEFPROG	-100-
6.4.1 Interface générale	-100-
6.4.2 Sélection des équipements.....	-100-
6.4.3 Alimentation Four	-101-
6.4.4 Préchauffement du four	-102-
6.4.5 Cuisson.....	-102-
6.4.6 Refroidissement et stockage Clinker	-103-
6.4.7 Tirage Gaz	-104-
6.5 Conclusion	-105-
Conclusion Générale	-107-

Bibliographie

Annexes

Annexe A: Principaux termes du process de Clinkérisation

Annexe B: Proportion des matières premières

Annexe C: Les algorithmes d'apprentissage du système neuro-flou

Annexe D: Système SDR/FUZZY LOGIC

Liste des figures

Figure 1.1: Structure Globale d'un système de production.....	- 15 -
Figure 1.2 : Classification des modes de fonctionnement.....	- 16 -
Figure 1.3: Composantes de la surveillance industrielle.....	- 17 -
Figure 1.4: Surveillance Classique et Surveillance Dynamique.....	- 18 -
Figure 1.5: Composantes de la Surveillance Dynamique.....	- 18 -
Figure 1.6: Classification de méthodes de diagnostic.....	- 19 -
Figure 1.7: Structure d'un système de diagnostic par Reconnaissance des Formes.....	- 20 -
Figure 1.8: a- Détection, Diagnostic et Pronostic b- Détection, Diagnostic et Pronostic dans les systèmes de maintenance intelligente phénoménologiques.....	- 21 -
Figure 1.9: Classification des approches de Pronostic.....	- 23 -
Figure 2.1: Structure d'une commande floue.....	- 27 -
Figure 2.2: Méthode de fuzzification pour une mesure exacte.....	- 29 -
Figure 2.3: Méthode de fuzzification pour une mesure incertaine.....	- 29 -
Figure 2.4: Modèle de neurone.....	- 31 -
Figure 2.5: Réseau de neurone non bouclé.....	- 31 -
Figure 2.6: Illustration d'apprentissage non supervisé.....	- 32 -
Figure 2.7: Illustration d'apprentissage supervisé.....	- 33 -
Figure 2.8: Exemple de réseau de type perceptron Multi-couches.....	- 33 -
Figure 2.9: Fonction d'activation d'un neurone caché possédant une seule entrée.....	- 34 -
Figure 2.10: Classification des réseaux de neurones temporels.....	- 35 -
Figure 2.11: Réseau récurrent de Jordan.....	- 37 -
Figure 2.12: Réseau récurrent d'Elman.....	- 38 -
Figure 2.13: le système Neuro-Flou.....	- 39 -
Figure 2.14: Architecture NEFCLASS.....	- 40 -
Figure 2.15: Architecture ANFIS.....	- 41 -
Figure 2.16: Architecture NEFCON.....	- 41 -
Figure 2.17: Architecture NEFPROX.....	- 42 -
Figure 2.18: Architecture NEFDIAG.....	- 42 -
Figure 3.1 : Schéma synoptique du process de fabrication du ciment.....	- 45 -
Figure 4.1: Automate S7-400.....	- 58 -
Figure 4.2: Applications disponibles du STEP7.....	- 60 -
Figure 4.3 : Fuzzy Control++.....	- 62 -
Figure 4.4: Neuro Systems.....	- 63 -

Liste des figures

Figure	4.5: Neuro-Fuzzy Systems	- 64 -	
Figure	5.1: Four Rotatif de la SCIMAT	- 68 -	
Figure	5.2: Zones de réaction du Four Rotatif	- 69 -	
Figure	5.3: Pronostic industriel par RdF Neuro-Floue	- 70 -	
Figure	5.4: Architecture Neuro-Floue d'Elman	- 75 -	
Figure	5.5: Système Flou.....	- 76 -	
Figure	5.6: Fonctions d'appartenance de %O ₂ et NO _x	- 76 -	
Figure	5.7: Fonctions d'appartenance de clinker et gaz.....	- 76 -	
Figure	6.1: Composants du programme	-81-	
Figure	6.2: Interface OS du système	-82-	
Figure	6.3: Blocs du programme S7	-83-	
Figure	6.4: Forçage.....	-83-	
Figure	6.5: Temporisateur	-84-	
Figure	6.6: Comparaison	-84-	
Figure	6.7: Commande de marche alimentateur Schenck	-85-	
Figure	6.8: Commande de marche de ventilateur de tirage.....	-87-	
Figure	6.9: Commande de marche de bruleur à gaz.....	-87-	
Figure	6.10: Commande de marche de transporteur à vis	-87-	
Figure	6.11: Séquence de démarrage de l'atelier de clinkérisation.....	-88-	
Figure	6.12: Arrêt du programme PRG13	-89-	
Figure	6.13: Alarmes et Pré-alarmes	-89-	
Figure	6.14: Traitement des pré-alarmes.....	-90-	
Figure	6.15: Template régulateur classique.....	-90-	
Figure	6.16: Stock de variables de régulateur	-90-	
Figure	6.17: Bloc d'un régulateur classique.....	-91-	
Figure	6.18: Template régulateur flou.....	-92-	
Figure	6.19: Bloc d'un régulateur Flou	-92-	
Figure	6.20: Connexion Fuzzy Control++ et DB31	-93-	
Figure	6.21: Ensemble flou Tirage	Figure 6.22: Ensemble flou Vitesse.....	-93-
Figure	6.23: Table Règles Floues.....		-93-
Figure	6.24: Courbe de régulation.....		-94-
Figure	6.25: Surface générée par la boucle de régulation		-94-
Figure	6.26: Template régulateur Neuro-flou		-94-
Figure	6.27: Bloc d'un régulateur Neuro Flou		-95-

Liste des figures

Figure	6.28: Connexion NeuroSystems et DB32	-95-
Figure	2.29: Connexion Fuzzy Control++ et NeuroSystems	-96-
Figure	6.30: Propriétés de l'entrée du réseau	-96-
Figure	6.31: Temps et Nombre d'itérations de l'apprentissage.....	-96-
Figure	6.32: Trace de l'apprentissage	-97-
Figure	6.33: Surface générée.....	-97-
Figure	6.34: Courbe de régulation.....	-97-
Figure	6.35: phase de test	-97-
Figure	6.36: Les ensembles flous	-98-
Figure	3.37: les fonctions d'appartenance	-98-
Figure	6.38: Table de règles floues	-99-
Figure	3.39: Chargement du fichier Fuzzy sur ANFIS edit	-99-
Figure	3.40: Apprentissage par ANFIS	-99-
Figure	6.41: Interface générale du NEFPROG.....	-100-
Figure	6.42: Sélection des équipements	-101-
Figure	6.43: Alimentation four.....	-101-
Figure	6.44: Interface régulateur	-102-
Figure	6.45: Tour de préchauffement.....	-102-
Figure	6.46: Zone de cuisson	-103-
Figure	6.47: Refroidissement et stockage Clinker	-104-
Figure	6.48: Tirage gaz	-104-

Liste des tableaux

Tableau 2.1: Comparaison entre la logique floue et les réseaux de neurones -----	39 -
Tableau 3.1: Taux de pannes des équipements de la chaîne technologique-----	51 -
Tableau 3.2: Programme d'échantillonnage -----	54 -
Tableau 3.3: Maîtrise de la proportionnalité entre Vitesse de rotation, Alimentation Four et Consommation Gaz & Tirage-----	54 -
Tableau 3.4: Paramètres de contrôle du four -----	55 -
Tableau 4.1: Types de blocs utilisés en STEP7-----	60 -
Tableau 5.1: L'AMDEC du procédé du clinkérisation -----	73 -
Tableau 5.2: Les entrées et les sorties du système -----	74 -
Tableau 5.3: Apprentissage des règles floues -----	78 -
Tableau 5.4: Apprentissage de fonctions d'appartenance-----	79 -
Tableau 6.1: Les Mnémoniques du Système -----	82

Abréviation

- *AMDEC* : Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité
- *ANFIS* : Adaptive-Network-based Fuzzy Inference System
- *IA* : Intelligence Artificielle
- *LAP* : Laboratoire d'Automatique et de Productique
- *LSF* : Lime Saturation Factor
- *MA* : Module Alumino-Ferique
- *MATLAB* : MATrix LABoratory
- *MS* : Module Silicique
- *NEFPROG* : NEuro Fuzzy PROGnosis,
- *PID* : Proportionnelle-Intégrale-Dérivée
- *PMC* : Perceptron Multicouches
- *RBF* : Radial Basis Function
- *RdF* : Reconnaissance des Formes
- *RdFSNF* : RdF Statistique Neuro-Floue
- *RNA* : Réseau de Neurone Artificiel
- *SCIMAT* : Société des CIMents d'Ain-Touta
- *SDR* : Supervision -Dialogue Reports
- *SGBD* : Système de Gestion de la Base de Données
- *SIF* : Système d'Inférence Floue
- *SNF* : Système Neuro-Flou
- *SNMC* : Société Nationale des Matériaux de Construction
- *UML* : Unified Modeling Language
- *WinCC* : Windows Control Center

INTRODUCTION

GENERALE

Le diagnostic consiste à déterminer à chaque instant le mode de fonctionnement dans lequel le système se trouve. Il est basé sur des observations multiples. Ces observations sont regroupées pour former des classes qui définissent une situation ou un mode de fonctionnement du processus, auxquelles une nouvelle observation sera comparée pour être identifiée.

Pour cela, beaucoup de travaux ont été dédiés à l'élaboration des systèmes d'aide au diagnostic. La majorité de ces derniers se base sur des outils informatiques qui couplent des méthodologies de l'intelligence artificielle comme la logique floue, les systèmes experts, les réseaux de neurones et les algorithmes génétiques.

En effet, le problème du diagnostic peut être vu comme un problème de reconnaissance des formes où les classes correspondent aux différents modes de défaillance du système et les formes représentent l'ensemble des observations ou mesures du système (données qualitatives ou quantitatives).

Si un procédé industriel est supervisé par un opérateur humain, on peut comprendre aisément qu'il lui est difficile de surveiller en même temps un grand nombre de paramètres «variables». Une technique de diagnostic est alors nécessaire pour faire l'analyse des variables pertinentes et donner une information plus claire, concise et utile pour que l'opérateur puisse prendre des décisions. Une solution développée pour traiter les problèmes de diagnostic, c'est la combinaison des outils informatiques (la logique floue et les réseaux de neurones).

Alors, les systèmes neuro-flous sont utilisés pour résoudre les problèmes de classification des défaillances. Mais, lorsqu'une méthode de surveillance estime les conditions futures d'un système à partir de son état actuel, on parle de diagnostic prédictif (pronostic). Cette prise en compte du passé du signal n'est possible que des architectures de réseaux de neurones temporelles. Les réseaux de neurones statiques sont incapables d'assurer ce genre de traitement. Une détection précoce d'un palier de dégradation ou même l'apprentissage d'une séquence d'évolution d'un système nécessite une architecture neuronale temporelle (dynamique).

L'objectif de ce travail est le développement d'un module de pronostic industriel basé sur la reconnaissance de formes statistique neuro-floue temporelle, qui s'appuie sur une représentation numérique et au même temps symbolique des formes. La réalisation pratique de notre travail concerne le domaine de l'industrie du ciment et, plus précisément, le processus de clinkérisation de la Société de fabrication des CIMENTS d'Ain-Toute-BATNA.

Au delà des perspectives des travaux des collègues, « Dr. Dj.Mouss et Mr. R.Mahdaoui » du LAP-Batna, qui sont appliqués à l'atelier de clinkérisation de la cimenterie pour résoudre le problème de diagnostic. Notre étude permet de suivre le comportement d'un processus de clinkérisation, une estimation de la durée de fonctionnement avant défaillance et du risque d'apparition ultérieure d'un ou de plusieurs modes de défaillance, et de classification des pannes. Cette solution donne naissance à un outil neuronal temporel de pronostic industriel « **NEFPROG** » (**NEuro Fuzzy PROG**nosis).

Afin d'atteindre cet objectif, notre mémoire est organisée en six chapitres qui peuvent être ainsi résumés :

Le **premier chapitre** est dédié au diagnostic et au pronostic d'un système de production. Les méthodologies de diagnostic à partir des données historiques sont généralement divisées en deux groupes : méthodes qualitatives et quantitatives. Puisque la fonction de diagnostic est vue comme une application de Reconnaissance des Formes (RdF), nous avons présenté le fonctionnement d'un système de diagnostic par RdF. Ensuite, nous présentons un état de l'art sur les approches de **pronostic** existantes dans la littérature en se basant sur les principales catégories d'outils généralement admises : pronostic basé sur l'expérience, pronostic basé sur les données et pronostic basé sur les modèles.

Le **deuxième chapitre** est consacré à la présentation de la logique floue et les réseaux de neurones artificiels statiques et dynamiques dans le domaine du diagnostic des systèmes de production. La représentation du temps dans les réseaux de neurones représente une caractéristique essentielle dans la perspective d'une surveillance industrielle dynamique.

La combinaison de la logique floue et de réseaux de neurones donne lieu la naissance des systèmes neuro-flous. Ces derniers permettent donc d'exploiter les capacités d'apprentissage des réseaux de neurones d'un part et les capacités de raisonnement de la logique floue d'autre part. Ainsi les architectures les plus connues et les plus utilisés dans l'industrie et dans les domaines de classification sont présentés.

Le **troisième chapitre**, est consacré à une étude générale de la « Cimenterie », lieu de stage et plus précisément le champ d'étude « l'atelier de clinkérisation ». Une présentation de la cimenterie d'Ain-Touta Batna ainsi qu'une explication du processus de fabrication du ciment est alors faites. Après, une proposition d'un cahier des charges du progiciel proposé. C'est un cahier des charges fonctionnel « cahier des charges technique » qui a été mis au point.

Le **quatrième chapitre** donne lieu l'environnement de travail de notre étude. Puisque la SCIMAT de Ain-Touta utilise dans son installation les automates programmables industriels, nous avons présenté l'automate *Siemens S7-400* et ses caractéristiques, qui est réellement le type installé dans la cimenterie. Une présentation d'un ensemble de langages de programmation et les outils nécessaires utilisés pour implémenter notre approche. Ensuite, nous avons utilisé la méthode *AMDEC* (Analyse des Modes de Défaillances de leurs Effets et de leur Criticité) pour identifier toutes les défaillances du système à étudier.

Le **cinquième chapitre** est dédié à la présentation de l'approche neuro-floue temporelle pour le pronostic industriel. En effet, nous proposons une stratégie pour le suivi du comportement d'un processus et la détection des défaillances. Une approche de pronostic industriel basée sur la reconnaissance de formes statistique neuro-floue qui s'appuie sur une représentation numérique et au même temps symbolique des formes est alors mise au point.

Le **sixième chapitre** est réservé à la présentation d'une exploitation industrielle innovante sur laquelle a débouché notre étude. Afin d'implémenter cette approche et l'exploiter, pour diagnostiquer un système de production en temps réel, nous avons développé un *pro-logiciel* informatique NEFPROG (*NEuro Fuzzy PROG*nosis).

Enfin, la dernière partie conclue ce mémoire. Nous y dresserons un bilan final de notre travail et nous y donnerons quelques perspectives de recherche prometteuses.

CHAPITRE 01

DIAGNOSTIC INDUSTRIEL D'UN SYSTÈME DE PRODUCTION

La surveillance se compose de deux fonctions principales qui sont la détection et le diagnostic. Le diagnostic des pannes dans les installations industrielles est le processus qui, à partir des symptômes observés, permet l'identification des causes à l'origine des dysfonctionnements et d'incrémenter le composant en panne du système. Le diagnostic prédictif des pannes est l'estimation de la durée de fonctionnement avant défaillance et du risque d'existence d'un ou plusieurs modes de défaillance.

1.1 Introduction

La surveillance industrielle est une partie intégrante de la sûreté de fonctionnement et en constituant une fonction de plus en plus importante en pilotage des systèmes. Les besoins dans ces deux domaines évoluent constamment.

L'un des problèmes d'une grande importance dans la surveillance industrielle est constitué par la possibilité d'effectuer une détection des dégradations et éventuellement de prédire une défaillance. Ces problématiques rentrent dans le cadre de la surveillance dynamique permettant à terme de s'orienter vers le concept de diagnostic prédictif (*pronostic*).

Ce chapitre débute par un aperçu sur le système de production, une vue générique sur la surveillance industrielle. Ensuite une vision globale sur le diagnostic industriel et une classification des différentes techniques du diagnostic en spécifiant la méthode de Reconnaissance des Formes (RdF) et pour terminer par une vue globale sur le pronostic industriel.

1.2 Système de production

1.2.1 Définition

Tout système de production est composé d'une partie opérative et d'une partie commande. Pour une séquence de tâches à réaliser, la partie commande donne l'ordre d'exécution à la partie opérative[21].

Dès que la tâche est effectuée, celle ci informe la partie commande de la réalisation de chaque tâche. Ces échanges représentés sur la *figure 1.1*, sont mis en évidence par les ordres (actions) et les comptes-rendus (capteurs)

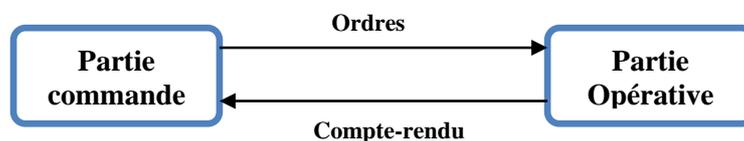


Figure 1.1: Structure Globale d'un système de production

1.2.2 Modes de fonctionnement d'un système

Un système est un ensemble susceptible évolué dans le temps. Il est composé d'un ensemble d'éléments interagissant entre eux et avec l'extérieur, réunis pour assurer une mission.

Suivant le mode de fonctionnement sous lequel le système évolue, la mission pour laquelle le système a été conçu peut être totalement remplie, partiellement remplie ou non remplie [21].

Les modes de fonctionnement sont de différents types, comme le montre la *figure 1.2*:

- Les modes de fonctionnement « *normaux* » : ils regroupent le mode de fonctionnement nominal qui doit être, en principe, en parfaite adéquation avec la qualité de la mission.
- Les modes de fonctionnement « *anormaux* » : lorsque le système est dans ce mode, la mission peut être partiellement remplie ou non remplie ; en tout état de cause, le service n'est pas satisfaisant.

Ces modes peuvent être décomposés en :

- *Modes interdits* : ce sont des modes sous lesquels le système ne doit absolument pas fonctionner pour des raisons de sécurité.
 - *Modes défailants* : ils correspondent à de mauvais fonctionnement du système. Un sous-ensemble de ces modes est constitué des modes de pannes où un sous-ensemble du système voire le système complet ne fonctionne plus.
 - *Modes dégradés* : ils correspondent à l'accomplissement de la mission soit de façon partielle, soit avec des performances moindres.
 - *Modes critiques* : ce sont des modes pour lesquels le système présente des caractéristiques de fonctionnement très particulières et souvent non souhaitées.
- Les modes de fonctionnement «d'exception» : ce sont des modes qui peuvent être normaux ou anormaux mais qui ont la caractéristique d'être peu tolérés ou peu fréquents.

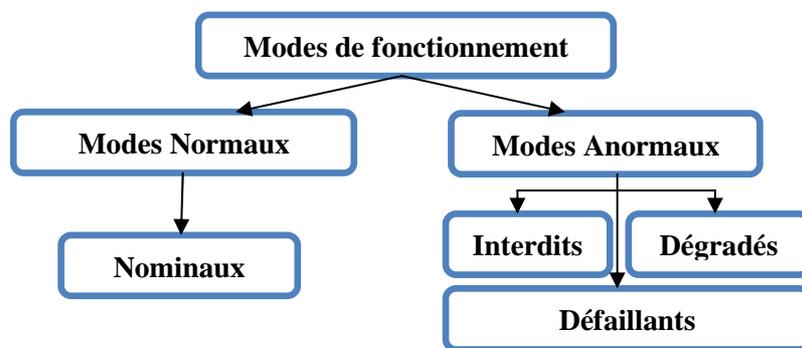


Figure 1.2 : Classification des modes de fonctionnement

1.3 Théorie générale du diagnostic et de la surveillance

La diversité des définitions trouvées dans différents travaux [6], [10], [14], [16], [17] et [22] fait que nous avons jugé important d'établir un lexique sur les termes qui seront utiles pour la compréhension du présent mémoire.

Dans la littérature associée au domaine de la surveillance industrielle, on peut trouver plusieurs définitions quelquefois divergentes. C'est pour cette raison que nous considérons important d'affirmer notre point de vue, en donnant des définitions des notions de base indispensables au positionnement de notre recherche.

*La **surveillance** est un dispositif passif, informationnel, qui analyse l'état du système et fournit des indicateurs. La surveillance consiste notamment à détecter et classer les défaillances en observant l'évolution du système, puis à les diagnostiquer en localisant les éléments défailants et en identifiant les causes premières.*

La surveillance se compose donc de deux fonctions principales qui sont la détection et le diagnostic (voir la figure 1.3).

*Pour **détecter les défaillances** du système, il faut être capable de classer les situations observables comme étant normales ou anormales.*

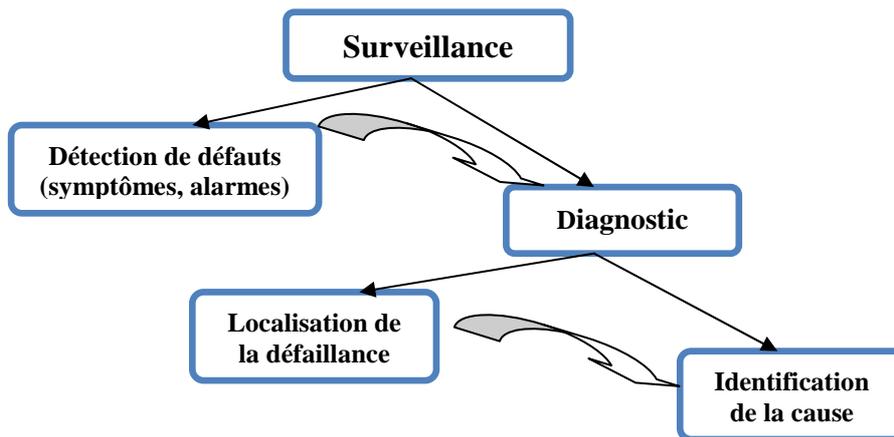


Figure 1.3: Composantes de la surveillance industrielle

L'objectif de la fonction **diagnostic** est de rechercher les causes et de localiser les organes qui ont entraîné une observation particulière. Sa fonction se décompose en deux fonctions élémentaires : la localisation permet de déterminer le sous-ensemble fonctionnel défaillant tandis que l'identification de la cause consiste à déterminer les causes qui ont mené à une situation anormale.

A partir de l'observation d'un état de panne, la fonction diagnostic est chargée de retrouver la faute qui en est à l'origine. Ce problème est difficile à résoudre.

En effet si, pour une faute donnée, il est facile de prédire la panne résultante, la démarche inverse qui consiste à identifier la faute à partir de ses effets, est beaucoup plus ardue.

Une **dégradation** représente une perte de performances d'une des fonctions assurées par un équipement.

Si les performances sont au-dessous du seuil d'arrêt défini dans les spécifications fonctionnelles de cet équipement, il n'y a plus dégradation mais défaillance.

Une **défaillance** est l'altération ou la cessation de l'aptitude d'un ensemble à accomplir sa ou ses fonctions requises avec les performances définies dans les spécifications techniques.

Une **panne** est l'inaptitude d'une entité (composant ou système) à assurer une fonction requise.

A partir du lexique des différents termes donnés, on peut constater qu'il y a une différence entre la surveillance classique et la surveillance dynamique. Cette différence se trouve dans la figure 1.4.

Comme pour la surveillance classique, la surveillance prédictive est un dispositif passif, informationnel, qui analyse l'état présent et passé du système et fournit des indicateurs sur les tendances d'évolution future du système.

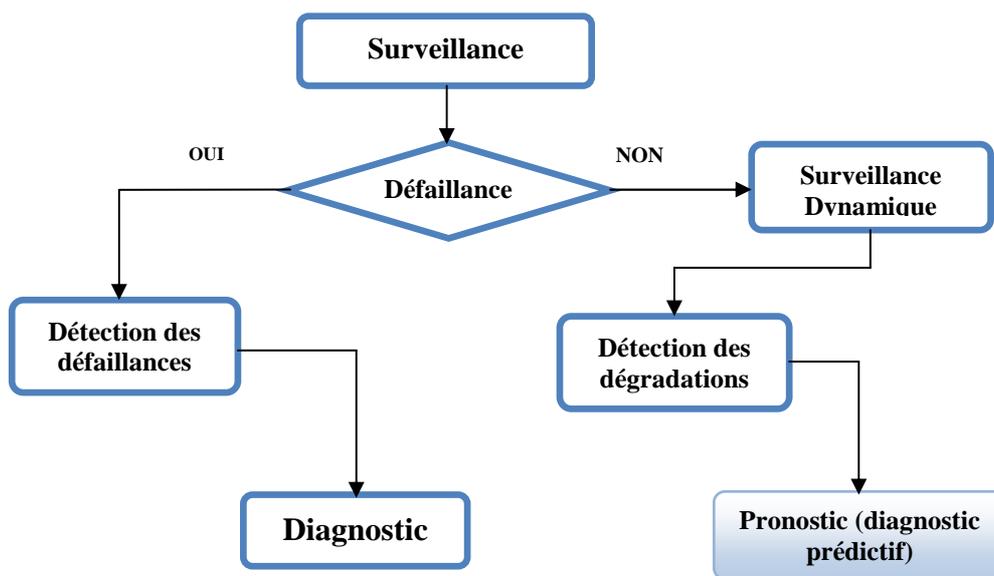


Figure 1.4: Surveillance Classique et Surveillance Dynamique

La surveillance prédictive se compose de : la *détection prédictive* (dynamique) et du *diagnostic prédictif*, appelé aussi *pronostic* (voir la figure 1.5).

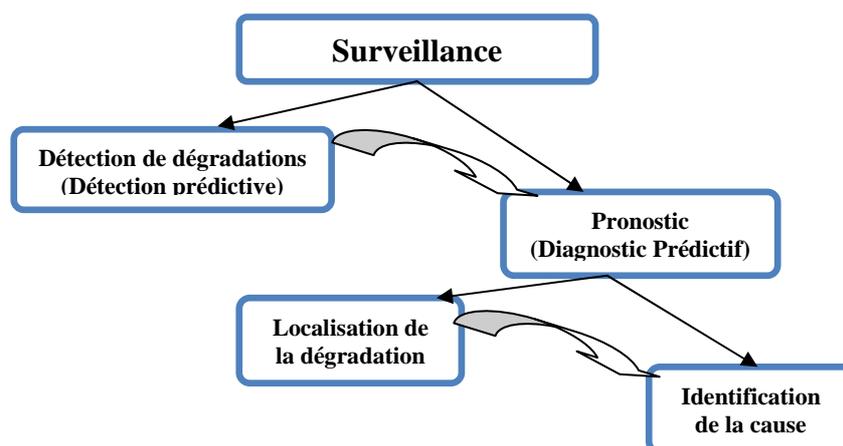


Figure 1.5: Composantes de la Surveillance Dynamique

La **détection prédictive** consiste à prédire une défaillance future. En d'autres termes, le but de la détection prédictive est de détecter une dégradation au lieu d'une défaillance. L'objectif du diagnostic prédictif (pronostic) est d'identifier les causes et de localiser les organes qui ont entraîné une dégradation particulière.

1.4 Méthodes de diagnostic

Les méthodes de diagnostic varient selon le type de représentation des modes de fonctionnement (figure 1.6), elles peuvent être classées comme suit :

- Les méthodes basées sur des modèles analytiques ou mathématiques,
- Les méthodes symboliques ou les systèmes experts,
- Les méthodes basées sur la RdF.

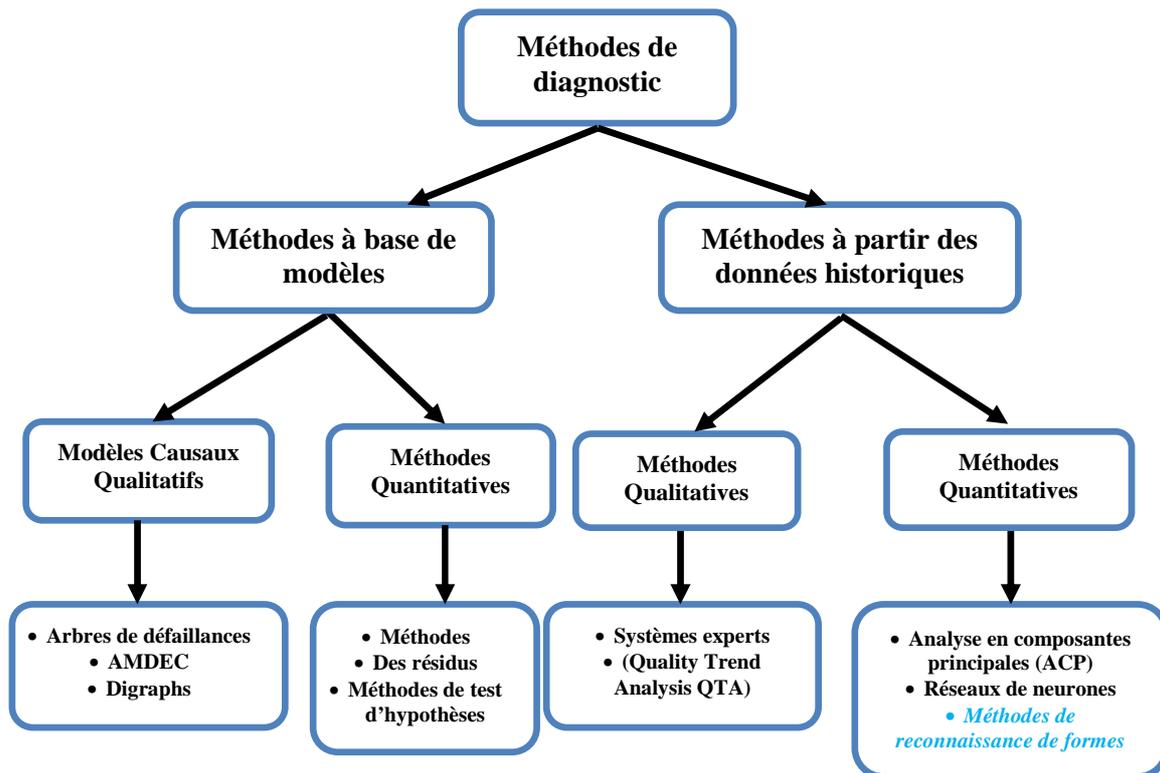


Figure 1.6: Classification de méthodes de diagnostic [23]

La connaissance a priori et la connaissance instantanée des modes de fonctionnement sont matérialisées par un ensemble d'observations. Le type de représentation de ces observations détermine le choix de la méthode de diagnostic.

Lorsque les observations sont de type symbolique, des effets, les systèmes experts s'imposent. Chaque système expert contient une base de connaissance, une base de faits et de règles, et un moteur d'inférence. Ce dernier choisit les règles en fonction des faits observés.

Quand les observations sont de types numériques, issus de capteurs, et que l'on dispose d'un modèle mathématique du système, les méthodes basées sur des modèles analytiques sont privilégiées.

En revanche, si les observations sont de type numérique et qu'on ne peut pas construire de modèle mathématique correspondant aux divers modes de fonctionnement du système ou si sa complexité le rend inexploitable, les méthodes basées sur la RdF sont les plus adaptées.

1.5 Diagnostic par Reconnaissance des Formes

La RdF est la science qui se base sur la définition d'algorithmes permettant de classer des objets ou des formes en les comparant à des formes types.

Le diagnostic par RdF peut être présenté comme une solution alternative à l'approche par modèle puisque les modes de fonctionnement sont modélisés, pas de manière analytique, mais en utilisant uniquement un ensemble des mesures de ces modes.

La figure 1.7 présente la structure simplifiée d'un système de diagnostic par reconnaissance des formes. La fonction d'observation a pour rôle d'élaborer le vecteur forme à partir des mesures et observations réalisées sur l'installation. La forme ainsi générée est appliquée au bloc de classification permettant de réaliser son affectation à l'une des classes connues et au mode de fonctionnement correspondant [11].

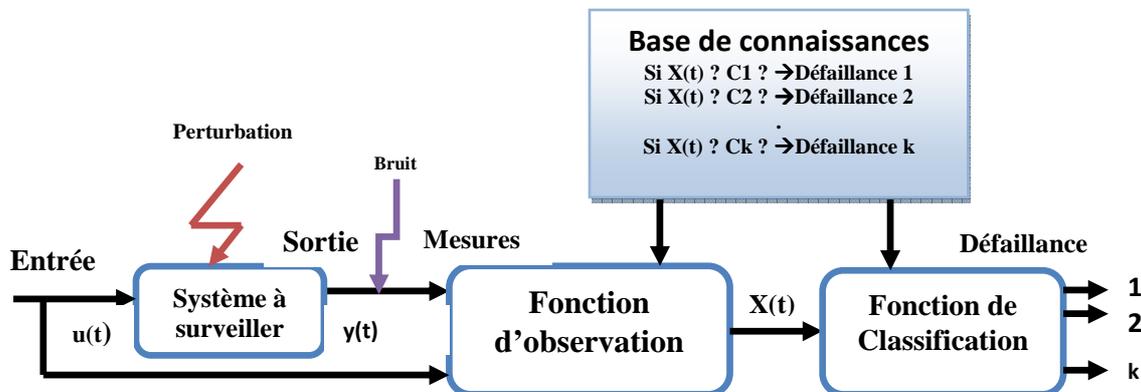


Figure 1.7: Structure d'un système de diagnostic par Reconnaissance des Formes

Le fonctionnement d'un système de diagnostic par RdF se déroule en trois phases: une phase d'analyse, une phase de choix d'une méthode de décision et une phase d'exploitation [8].

1.5.1 Phase d'analyse

Il s'agit de construire le vecteur forme x . Elle consiste à déterminer les paramètres et les techniques de prétraitement nécessaires pour permettre une meilleure reconnaissance ou classification. Elle est le fruit de l'expérience acquise sur le processus et conditionne le succès du système de diagnostic par RdF. Un ensemble de N observations : x_1, x_2, \dots, x_N , recueillies sur le processus à l'aide des capteurs, constitue l'ensemble d'apprentissage.

1.5.2 Phase de choix d'une méthode de décision

Les méthodes de décision définissent une règle de décision pour la classification des nouvelles observations aux différentes classes de l'ensemble d'apprentissage. La performance de ces méthodes est estimée en utilisant un indice de performance, en général on choisit la probabilité de mauvaise classification. La règle de décision peut être construite en utilisant une approche statistique ou une approche analytique.

1.5.3 Phase d'exploitation

Dans cette phase, le système de diagnostic par RdF peut être mis en service. Il permet de classifier chaque nouvelle observation recueillie sur le système dans une des classes connues, en appliquant la règle de décision élaborée dans la phase d'analyse. La détermination de cette classe permet de connaître le mode de fonctionnement actuel du système.

1.6 Du Diagnostic vers Pronostic

L'un des enjeux les plus importants de l'industrie concerne aujourd'hui la sûreté de fonctionnement des processus technologiques et la recherche de l'augmentation de leur fiabilité, de leur disponibilité. Pour les industriels, il est donc impératif d'appréhender l'état de gravité d'un défaut et de prévoir le moment optimal pour arrêter une machine et intervenir.

Alors, l'activité de surveillance fait traditionnellement appel à différents processus visant d'abord, à « percevoir » certains phénomènes (détection), ensuite, à les « comprendre » (diagnostic), et enfin, à « agir » en conséquence (choix d'actions de maîtrise). Aussi, plutôt que de comprendre à posteriori un phénomène qui vient de se manifester (défaillance), il peut être opportun d'en « anticiper » l'apparition afin de recourir à des actions protectrices en conséquence : c'est ce qui peut être entendu sous l'étiquette « pronostic de défaillances ».

Le positionnement relatif de ces processus de « détection », « diagnostic » et « pronostic » est schématisé sur la figure 1.8-a.

D'un point de vue phénoménologie, leur complémentarité peut être expliquée comme suit (figure 1.8-b) :

- La détection vise à identifier le mode de fonctionnement du système et son état.
- Lorsqu'une défaillance est apparue, le diagnostic permet d'isoler et d'identifier le composant qui a cessé de fonctionner (propagation arrière : des effets vers les causes),
- Le pronostic vise la prédiction des états futurs du système (propagation avant : des causes vers les effets).

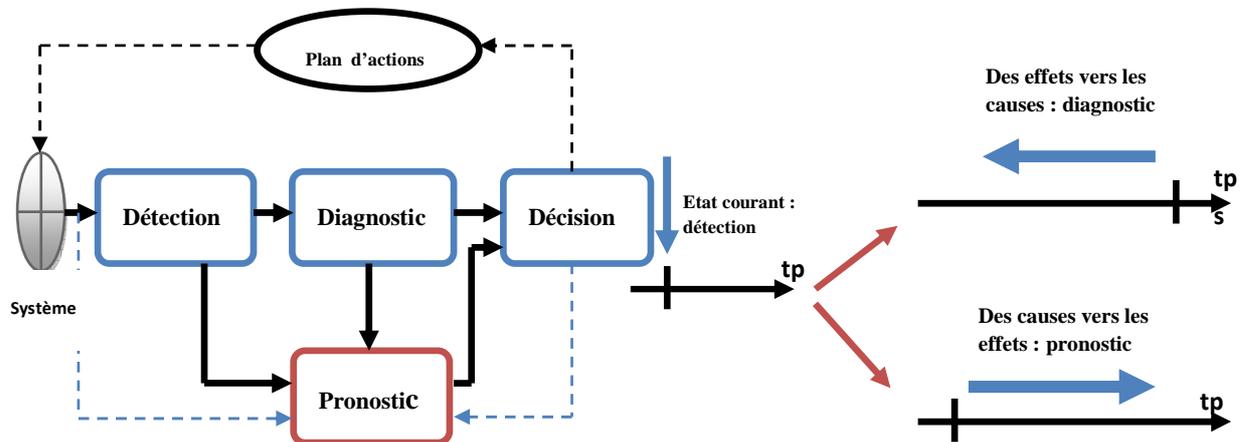


Figure 1.8: a- Détection, Diagnostic et Pronostic dans les systèmes de maintenance intelligente **b- Détection, Diagnostic et Pronostic phénoménologiques**

1.7 Le Pronostic industriel

1.7.1 Concept de pronostic

Différentes interprétations assignées au pronostic dans la littérature sont :

- "Après détection de la dégradation d'un composant ou d'un sous-système, le rôle du pronostic est de prédire l'évolution future des performances du système de production en tenant compte d'interventions de maintenance planifiées et éventuellement, de conditions opérationnelles ou environnementales changeantes" [22].

- "L'objectif du *pronostic* est d'identifier les causes et de localiser les organes qui ont entraîné une dégradation particulière".

Dans cette définition, le pronostic est assimilé à un diagnostic prédictif, qui s'exerce sur la dégradation et non pas sur la défaillance comme le diagnostic classique.

Le comité ISO définit le pronostic comme :

L'estimation de la durée de fonctionnement avant défaillance d'un système et du risque d'existence ou d'apparition ultérieure d'un ou de plusieurs modes de défaillance » (ISO 13381-1, 2004). [4]

La base conceptuelle générale d'un processus de pronostic est de (ISO, 2004) :

- Définir le point limite (généralement la mise à zéro),
- Etablir la gravité actuelle,
- Déterminer ou estimer les comportements des paramètres et la vitesse de détérioration escomptée,
- Déterminer la Durée estimée de Fonctionnement Avant Défaillance (DFAD).

Cependant, le pronostic est focalisé sur l'avenir et, de ce fait, doit prendre en compte les aspects suivants (ISO, 2004) [22] :

- les modes de défaillance existants et les taux de détérioration,
- les critères de déclenchement de futurs modes de défaillance,
- le rôle des modes de défaillance existants dans le déclenchement de futurs modes de défaillance,
- l'influence entre les modes de défaillance existants et les modes futurs et leurs taux de détérioration,
- la sensibilité à la détection et aux modifications des modes de défaillance existants et futurs du fait des techniques de surveillance actuelles,
- la conception et les changements de stratégies de surveillance afin de s'adapter à tous les éléments ci-dessus,
- L'effet des actions de maintenance et/ou des conditions d'exploitation,
- les conditions ou les hypothèses dans lesquelles les pronostics restent valables.

1.7.2 Classification des approches de pronostic

Il existe différents points de vue quant à la classification des méthodes de pronostic. Dans cette section, nous donnons un bref aperçu des approches utilisées (d'après les travaux [4], [15], [22]) pour résoudre des problématiques de pronostic.

Les méthodes de pronostic peuvent être associées à une ou plusieurs des trois approches (voir *figure 1.9*) suivantes :

- *Pronostic basé sur le modèle physique.*
- *Pronostic guidé par les données.*
- *pronostic basé sur l'expérience.*

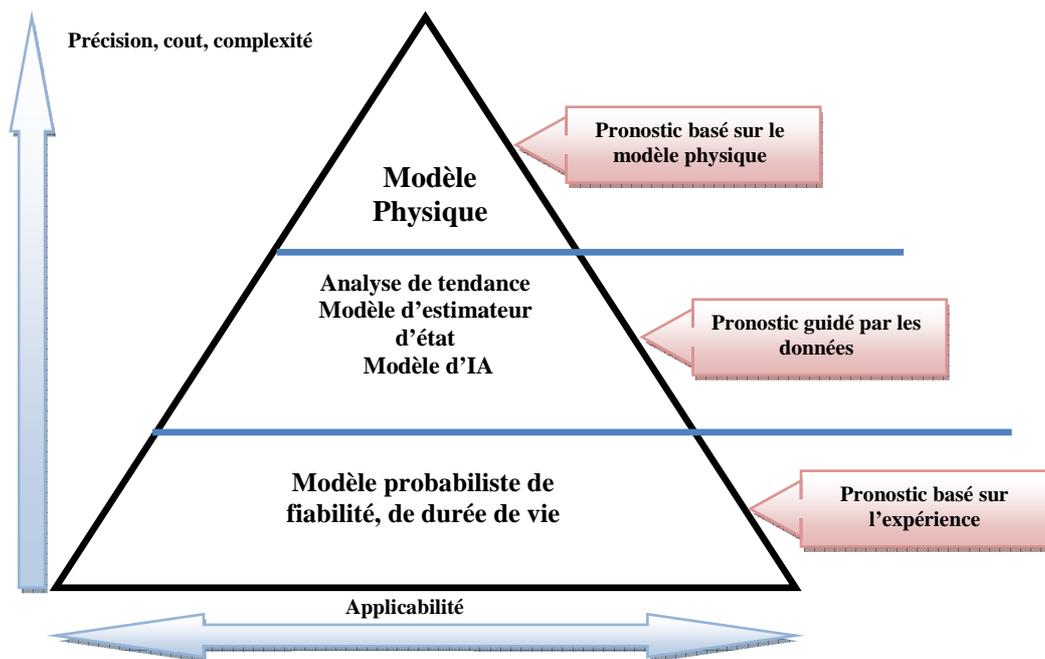


Figure 1.9: Classification des approches de Pronostic

1.7.2.1 Pronostic basé sur le modèle physique

La mise en œuvre de cette approche s'appuie généralement sur une représentation mathématique disponible du mécanisme de dégradation. Les relations causales issues des lois de la physique sont utilisées pour modéliser les interactions entre les entités du système. Ainsi, les résidus sont employés comme instruments mathématiques. De gros résidus dénotent la présence de défauts de fonctionnement, et des petits la présence de perturbations normales comme les bruits ou les erreurs de modélisation.

L'utilisation de méthodes de pronostic basées sur le modèle implique des connaissances spécifiques liées à la défaillance ainsi qu'une forte maîtrise du mode de fonctionnement du système analysé.

Principe de fonctionnement, avantages et inconvénients

Le principe du pronostic basé sur des modèles physiques est de déterminer le niveau de dégradation courant du système et d'évaluer le temps restant jusqu'à la défaillance en utilisant la courbe d'évolution de la dégradation en fonction de la sollicitation du système.

Les principaux avantages de cette approche sont :

- la flexibilité : si une des propriétés du système ou de la dégradation change, le modèle peut être réajusté pour prendre en compte cette modification.
- la connaissance de la structure du modèle permet de relier la variation d'indicateurs à une modification d'un paramètre ce qui n'est pas le cas des approches "guidées par les données".

Les principaux inconvénients de ce type de pronostic sont les suivants :

- Le développement des modèles est extrêmement coûteux. Il nécessite un niveau de qualification élevé et une expérience certaine de la part des développeurs,
- Il est difficile de construire un modèle global pour les systèmes complexes : complexité de modélisation des interactions entre les différents mécanismes, difficultés de calcul associées à la résolution d'un système d'équations différentielles,
- Il peut être impossible de généraliser une approche basée sur un modèle physique orientée "composant" à un niveau "système",
- Il est nécessaire de connaître les mécanismes de dégradation et des facteurs influents sur celles-ci.

1.7.2.2 Pronostic guidé par les données

Le pronostic guidé par les données s'appuie sur le constat suivant: les mesures (entrées/sorties) constituent souvent la plus forte et la plus sûre source d'information pour comprendre les phénomènes de dégradation. Sa force réside dans l'aptitude à apprendre et à capturer les relations subtiles entre données, même si ces relations sont inconnues ou difficile à décrire.

Principe de fonctionnement, avantages et inconvénients

Le pronostic guidé par les données exploite les indicateurs de dégradation ou d'interventions de maintenance délivrés respectivement par les processus de surveillance et d'aide à la décision (par exemple, données calorimétriques de calibrage, données spectrométriques, puissance, vibration et signal, température, pression, débris d'huile, tensions acoustiques de courants).

Le diagnostic situé en amont conditionne le succès du pronostic par sa capacité à fournir une estimation fiable et précise de l'état de santé courant du système et une mise à jour des paramètres des processus de détérioration. Ce type de pronostic se fonde sur l'hypothèse que les caractéristiques statistiques des données sont relativement inchangées à moins qu'un défaut de fonctionnement ne se produise dans le système.

La capacité d'adaptation à tout type d'application disposant de données suffisantes en quantité et en qualité représente un point fort pour cette approche de pronostic. En même temps, la mise en œuvre d'une approche "guidée par les données" est relativement simple car elle ne requiert pas la connaissance formelle des mécanismes de dégradation.

La capacité à transformer des données bruitées en information pertinentes pour des décisions de diagnostic/pronostic est un autre avantage pouvant être souligné.

L'inconvénient principal de ces approches est que leur efficacité est grandement dépendante de la quantité et de la qualité des données opérationnelles de système.

1.7.2.3 Pronostic basé sur l'expérience

Le pronostic basé sur l'expérience est basé sur la formalisation des mécanismes physiques de détérioration des composants par modèles stochastiques (loi de fiabilité, processus markoviens ou non-markoviens) initiés par connaissances a priori et jugement d'expert.

Principe de fonctionnement, avantages et inconvénients

Le principal avantage de cette méthode est qu'elle ne nécessite pas la connaissance pointue du (des) mécanisme(s) physique(s) de dégradation. Aussi, elle est relativement simple à mettre en œuvre et peu coûteuse.

Les principaux inconvénients sont les suivants :

- le manque de réactivité face au changement de comportement d'un système ou de l'environnement,
- les applications se focalisent sur des composants critiques, traités individuellement et donc le développement d'approches orientées "système" est rare,
- il existe fréquemment un décalage entre les modèles (système mono composant à 2 états) mis au point et la réalité industrielle (système multi composants multi états). L'origine de ce décalage provient souvent de l'incapacité des méthodes à effectuer les calculs engendrés par un système complexe.

1.8 Conclusion

Nous avons présenté dans ce chapitre un aperçu général sur le système de production et les différents modes de défaillances. Ensuite, un lexique de la terminologie utilisée dans le domaine de la surveillance industrielle ainsi que le diagnostic industriel, suivi d'une méthode de diagnostic en spécifiant la méthode de reconnaissance des formes. Bien que la littérature sur le pronostic soit beaucoup plus restreinte que celle sur le diagnostic, divers travaux proposent des catégories de pronostic.

CHAPITRE 02

OUTILS DE LA RECONNAISSANCE DES FORMES POUR LE DIAGNOSTIC INDUSTRIEL

Le développement des différentes méthodes du diagnostic est indispensable, ceci est dû à la complexité accrue des systèmes et des besoins des clients. Ce développement apparaît dans la combinaison des différentes techniques de l'intelligence artificielle et les méthodes du diagnostic. Cette combinaison a montré son efficacité durant son application et ces résultats engendrés dans tous les domaines industriels. Une présentation dans ce chapitre des deux techniques de la reconnaissance de formes : la logique floue et les réseaux de neurones, ainsi leur combinaison pour obtenir le système neuro-flou.

2.1 Introduction

Les méthodes de RdF supposent qu'aucun modèle n'est disponible pour décrire les relations de cause à effet. La seule connaissance repose sur l'expertise humaine confortée par un solide retour d'expérience. La plupart de ces méthodes sont basées sur l'IA : la logique floue, les Réseaux de Neurones Artificiels (RNAs), les réseaux de neurones temporels, les Réseaux Neuro-Flous.

2.2 La logique floue

2.2.1 Introduction

Dans cette partie, nous allons présenter brièvement les concepts fondamentaux de la logique floue dans une perspective industrielle, définir et expliquer la terminologie utilisée en logique floue, la théorie des ensembles flous ainsi que le mode de raisonnement propre aux variables floues.

2.2.2 Définitions

*La **logique floue** permet de faire le lien entre modélisation numérique et modélisation symbolique, ce qui permet des développements industriels spectaculaires à partir d'algorithmes très simples de traduction de connaissances symboliques en entité numérique et inversement.*

La logique floue a été formulée par Lotfi A. Zadeh dans le milieu des années soixante. Elle constitue une généralisation de la logique booléenne classique, et ajoute cependant une fonctionnalité déterminante : la possibilité de calculer un paramètre en disant simplement dans quelle mesure il doit se trouver dans un intervalle de valeurs.

*Un **sous-ensemble flou** A de X est définie par une fonction d'appartenance qui associe à chaque élément x de X , le degré $\mu_A(x)$, compris entre 0 et 1, pour lequel x appartient à A : $\mu_A : X \rightarrow [0, 1]$.*

2.2.3 Structure d'une commande Floue

La Commande floue est l'application la plus utilisée de la logique floue. La structure conventionnelle d'une commande floue est présentée par la *figure 2.1*. Elle est composée de quatre blocs distincts dont les définitions sont données ci-dessous.

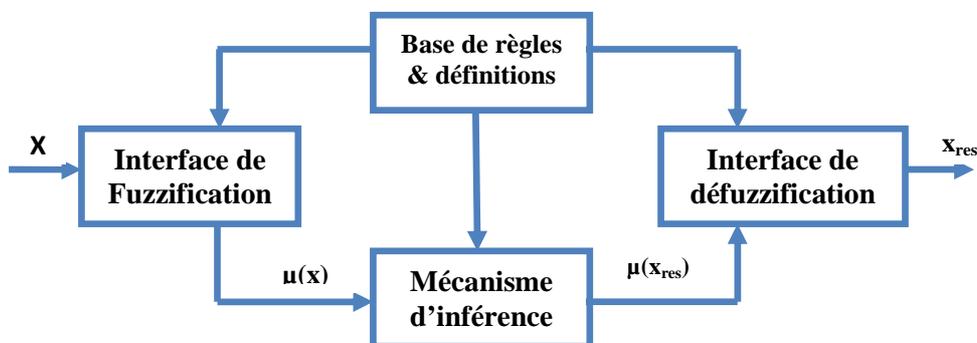


Figure 2.1: Structure d'une commande floue

Où : x représente le vecteur des entrées, x_{RES} celui des commandes (résultats), $\mu(x)$ et $\mu(x_{RES})$ les fonctions d'appartenance correspondantes.

2.2.3.1 Bases de règles et définitions

On regroupe dans ce bloc, d'existence virtuelle, l'ensemble des définitions utilisées dans la commande floue (univers de discours, partitions floue, choix des opérateurs...), ainsi que la base de règles «Si...alors...» de la stratégie de commande de l'expert.

a- Partition floue

La création et l'utilisation d'une base de règles nécessitent l'existence, pour chaque univers de discours considéré, de sous-ensembles flous particuliers. La définition de ces sous-ensembles flous fait l'objet de la partition floue.

La partition floue d'un univers de discours U consiste à définir n sous-ensembles flous F_i de façon à recouvrir U . C'est à dire que pour tout élément x de U , il faut assurer une appartenance minimale à l'union de F_i .

$$\cup F_i \supseteq U_s = \{x \in U; \mu_{u_s}(x) = \varepsilon\} \quad (2.1)$$

La condition (2.1) se traduit au niveau des fonctions d'appartenance par la condition ;

$$\forall x \in U; \mu_{F_1}(x) \vee \dots \vee \mu_{F_n}(x) \geq \varepsilon \quad (2.2)$$

Où \vee est un opérateur d'union (appelé également "ou"). Comme la fonction max minore toutes les fonctions utilisées comme opérateurs d'union, pour assurer une partition floue de niveau, il faut et il suffit que tout élément x de U possède un degré d'appartenance à l'union des F_i avec pour opérateur d'union la fonction max, supérieure ou égal à ε .

L'ensemble de sous-ensembles flous à définir dans une partition d'un univers de discours est fixé par l'expert. Plus ce nombre est important et plus on définit de classes sur cet univers, ce qui permet d'augmenter la sensibilité de la commande floue.

b- Bases de règles

Une fois la partition des univers de discours réalisée, il est possible de définir la base de règles. Celle-ci caractérise les relations entre les classes d'évènements possibles en entrée et les commandes correspondantes. Par conséquent, si l'on considère n univers de discours U_i pour les prémisses des règles floues et si pour chaque univers U_i on définit une partition en m_i sous ensembles flous, le nombre maximum de règles r_{max} est de:

$$r_{max} = \prod_{i=1}^n m_i \quad (2.3)$$

Le nombre de règles définis par l'expert peut-être inférieur à r_{max} .

2.2.3.2 Interface de fuzzification

Les opérateurs utilisés dans la commande floue agissent sur des sous-ensembles flous. Par conséquent, il est nécessaire de transformer les variables non floues provenant du mode extérieur en des sous-ensembles flous.

Pour ce faire, on utilise un opérateur dit de fuzzification qui associe à une mesure de la variable x_0 une fonction d'appartenance particulière $\mu_{x_0}(x)$.

Le choix de l'opérateur de fuzzification dépend de la confiance que l'on accorde aux mesures effectuées. Ainsi si la mesure x_0 est exacte, les sous ensemble flou x_0 doit être représentée par un fait précis. Par conséquent, on utilise comme opérateur de fuzzification : la transformation dite de singleton (*figure 2.8*). La fonction d'appartenance du sous-ensemble flou x_0 est alors définie par:

$$\mu_{x_0} : U \rightarrow U, \mu_{x_0}(x) = 1 \text{ si } x = x_0; \mu_{x_0}(x) = 0 \text{ si } x \neq x_0 \quad (2.4)$$

La *figure 2.2* montre l'aspect de cette fonction d'appartenance.

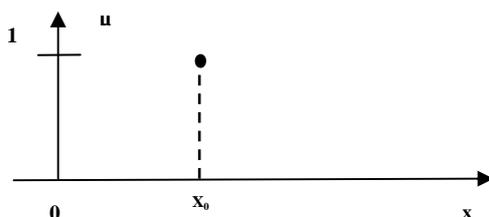


Figure 2.2: Méthode de fuzzification pour une mesure exacte

Ainsi, le sous ensemble flou X_0 réalisé par cette méthode de fuzzification ne comprend que l'élément x_0 . Par contre, si la mesure de la variable est incertaine, par exemple à cause de bruit, le sous ensemble flou X_0 doit être représentée par un fait imprécis. On utilise alors une méthode de fuzzification qui associe à la variable mesurée x_0 une fonction d'appartenance telle que, par exemple:

$$\mu_{x_0}(x) = \max \left\{ 0; 1 - \frac{|x - x_0|}{\varepsilon} \right\} \quad (2.5)$$

La représentation graphique de cette fonction est représentée par la *figure 2.3*. Ce sous-ensemble flou comprend donc la mesure x_0 avec une appartenance unité et les valeurs voisines de x_0 avec une appartenance inversement proportionnelle à l'écart avec x_0 .

La base du triangle est fonction de l'importance relative des erreurs de mesures. En effet, plus elles sont importantes, plus la mesure de la variable x_0 devient imprécise, et donc, plus le triangle doit s'élargir.

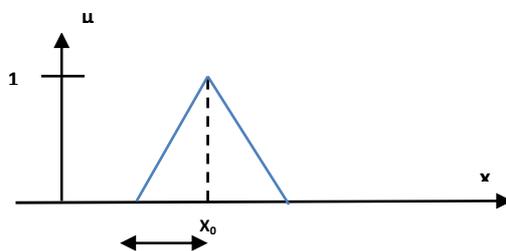


Figure 2.3: Méthode de fuzzification pour une mesure incertaine

2.2.3.3 Mécanismes d'inférence

A partir de la base de règles fournie par l'expert, et du sous ensemble flou X_0 correspondant à la fuzzification du vecteur de mesure $x_0 = [x_{0,1}, \dots, x_{0,n}]^T U$, le mécanisme d'inférence calcule le sous-ensemble flou $\mu(x_0)$ relatif à la commande du système.

En général, plusieurs valeurs de variables floues, convenablement définies par des fonctions d'appartenance, sont liées entre elles par des règles, afin de tirer des conclusions. On parle alors de déductions floues. Dans ce contexte, on peut distinguer deux types de règles d'inférences; celle avec une seule règle et celle avec plusieurs règles.

Dans les inférences de régulateur par la logique floue interviennent les opérateurs *ET* et *OU*. L'opérateur *ET* s'applique aux variables à l'intérieur d'une règle, tandis que l'opérateur *OU* lie les différentes règles.

Pour le réglage par la logique floue, on utilise en général, une des méthodes suivantes:

- Méthode d'inférence max-min,
- Méthode d'inférence max-prod,
- Méthode d'inférence som-prod.

2.2.3.4 Interface de défuzzification

Les méthodes d'inférences fournissent une fonction d'appartenance résultante $\mu_{RES}(x)$ pour la variable de sortie x , il s'agit donc d'une information floue. Les méthodes les plus couramment utilisées sont:

- La méthode de centre de gravité,
- La méthode de maximum,
- La méthode des surfaces,
- La méthode des hauteurs.

2.3 Les réseaux de neurones

Les RNAs sont un outil bien adapté aux problèmes de perception, de classification et de prédiction. Leur utilisation s'inscrit particulièrement dans les méthodologies de surveillance en l'absence de modèle du procédé.

L'utilisation des RNAs est principalement guidée par leurs propriétés comme la capacité d'apprentissage, le parallélisme dans le traitement, l'adaptation aux non-linéarités des systèmes et la rapidité de traitement.

Nous verrons ainsi les architectures les plus couramment utilisées pour ces problèmes de surveillance des systèmes industriels.

2.3.1 Généralités

Deux éléments principaux constituent un RNA, tout d'abord le(s) modèle(s) de neurones utilisés pour constituer le réseau et ensuite l'architecture du réseau. Chaque neurone artificiel est un processus élémentaire qui reçoit un certain nombre d'entrées de neurones amont. A chacune de ces entrées est associé un poids w représentatif de la force de connexion entre les neurones correspondants. (figure 2.4)

Ceci met donc en avant deux caractéristiques propres à chaque neurone:

- un « potentiel » ou « activation »,
- une fonction de transfert qui donne la sortie du neurone en fonction de son activation.

Les calculs du potentiel P_k et de la sortie S_k s'expriment par les relations suivantes :

$$P_k = \sum_i W_{ki} * X_i \quad (2.6)$$

$$S_k = N(P_k) \quad (2.7)$$

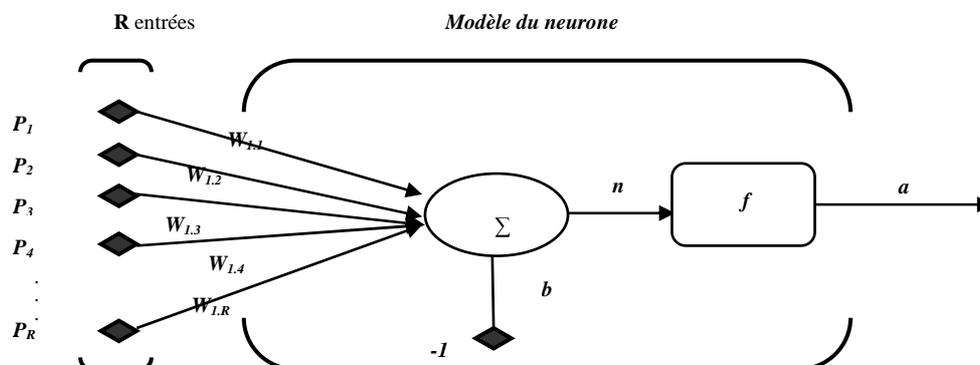


Figure 2.4: Modèle de neurone

Où: k: numéro d'ordre du neurone cible. i : numéro d'ordre du neurone émetteur.

N : fonction d'activation du neurone, ou fonction neurone linéaire ou non.

2.3.2 Les architectures neuronales

2.3.2.1 Les réseaux de neurones non bouclés « feedforward »

Un réseau de neurone non bouclé est présenté par un ensemble de neurones connectés entre eux telle que l'information circulant des entrées vers les sorties sans retour en arrière [16].

Le calcul de Y (sortie) se fait en propageant les calculs de la gauche vers la droite, avec éventuellement des connexions directes linéaires : $y = a*x + fw(x)$ (2.8)

Ce type de réseau comprend deux groupes d'architectures: les réseaux Monocouche et les réseaux Multicouches. La figure 2.5 illustre un RNA Multicouches non bouclé.

Ils diffèrent par l'existence ou non de neurones intermédiaires appelés neurones cachés entre les unités d'entrées et les unités de sorties appelées nœuds sources ou nœuds d'entrée et nœuds de sortie respectivement.

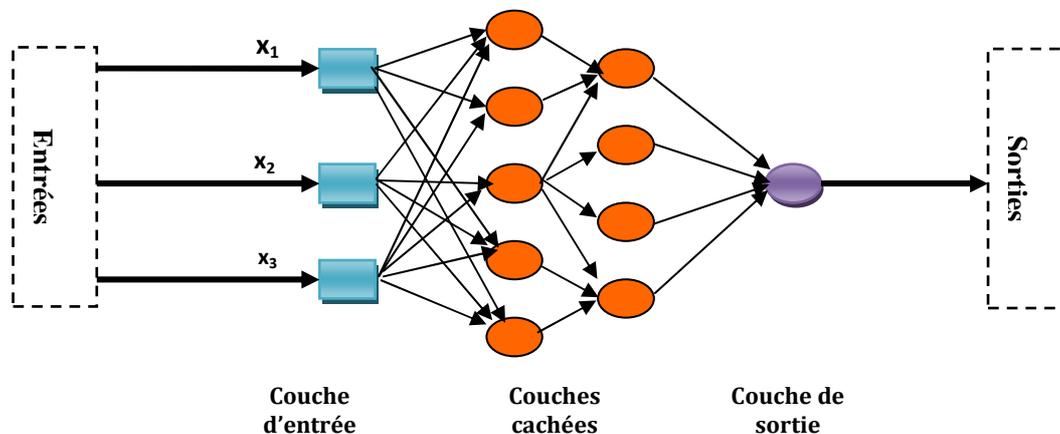


Figure 2.5: Réseau de neurone non bouclé

2.3.2.2 Les réseaux de neurones bouclés (récurrents)

Un réseau de neurone bouclé à temps discret réalise une ou plusieurs équations aux différences non linéaires, par composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions [6].

Ces réseaux caractérisent par la présence d'au moins une boucle de rétroaction au niveau des neurones ou entre les couches, et la prise en compte de l'aspect temporel du phénomène. Mais ce sont des modèles plus durs à mettre en œuvre.

2.3.3 Les types d'apprentissage des réseaux de neurones

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux. Elle ne concerne cependant pas tous les modèles, mais les plus utilisés. L'apprentissage est une phase du développement du réseau de neurones durant laquelle on calcule les poids des neurones de telle manière que les sorties du réseau soient aussi proche que possible des sorties désirées.

L'apprentissage RNA est une phase qui permet de déterminer ou de modifier les paramètres du réseau, afin d'adopter un comportement désiré. Les procédures d'apprentissage peuvent se subdiviser, en trois grandes catégories : apprentissage supervisé et non supervisé.

2.3.3.1 Apprentissage non supervisé

Les données fournies en entrée ne contiennent pas d'information sur la sortie désirée. L'apprentissage est réalisé à l'aide de règles qui modifient les paramètres du réseau en fonction des exemples fournis en entrée.

La figure 2.6 illustre, un exemple d'apprentissage non supervisé.

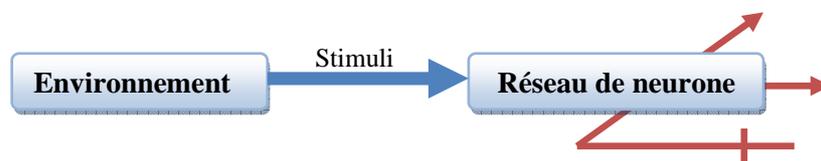


Figure 2.6: Illustration d'apprentissage non supervisé

Les RNAs qui utilisent ce type d'apprentissage sont appelés « auto organisatrice » où ce type d'apprentissage possède souvent une moindre complexité dans le calcul par rapport à l'apprentissage supervisé. En résumé, on peut dire :

- On fournit seulement des exemples X à l'algorithme.
- Il doit trouver w « les poids » tel que les X soient correctement groupés selon F_w (avec une bonne généralisation).

2.3.3.2 Apprentissage supervisé

Permet de déterminer les poids synaptiques à partir d'exemples étiquetés auxquels un expert a associé des réponses du réseau. Les paramètres du réseau sont donc modifiés de manière à minimiser l'erreur entre la sortie cible (fournie par l'expert) et la sortie réelle du réseau.

La figure 2.7 illustre, un exemple d'apprentissage supervisé.

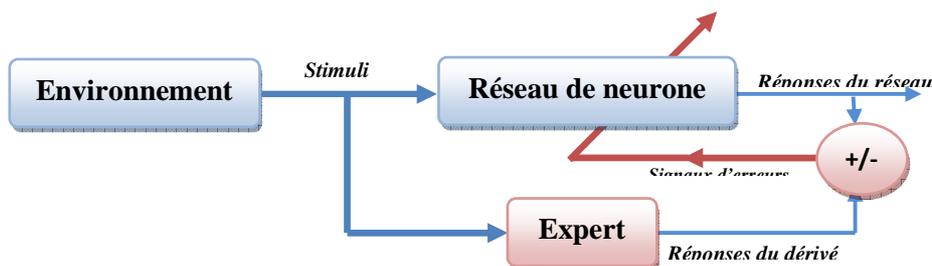


Figure 2.7: Illustration d'apprentissage supervisé

2.3.4 Les réseaux de neurones les plus utilisés

Aujourd'hui, le nombre de types de réseaux neuronaux possibles est assez élevé. On doit présenter les types les plus utilisés.

2.3.4.1 Perceptron Multicouches (PMC)

Le perceptron multicouches est un réseau orienté de neurones artificiels organisé en couches et où l'information voyage dans un seul sens, de la couche d'entrée vers la couche de sortie [6].

La figure 2.8 donne l'exemple d'un réseau contenant une couche d'entrée, deux couches cachées et une couche de sortie.

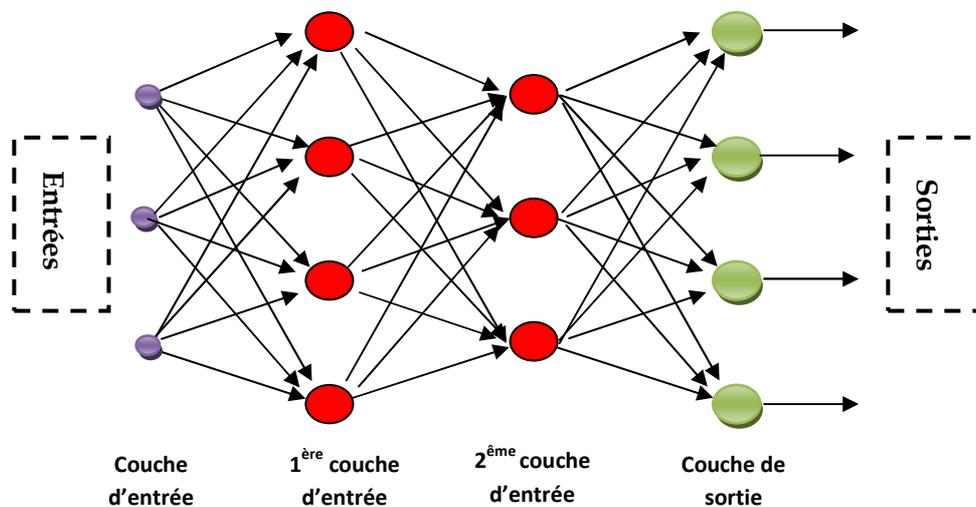


Figure 2.8: Exemple de réseau de type perceptron Multicouches

La création d'un perceptron multicouche pour résoudre un problème donné passe par l'inférence de la meilleure application possible telle que définie par un ensemble de données d'apprentissage constituées de paires de vecteurs d'entrées et de sorties désirées. Cette inférence peut se faire, entre autre, par l'algorithme dit de rétropropagation.

L'algorithme de rétropropagation, va donc réaliser l'apprentissage du réseau en modifiant les poids des connexions neurone par neurone en commençant par la couche de sortie. Le critère d'apprentissage étant la minimisation de la racine de l'erreur quadratique moyenne [14].

2.3.4.2 Les Réseau de neurones à Fonctions de Base Radiales

Les réseaux à fonctions de base radiales (RBF) ont été proposés par J. Moody et C. Darken. On retrouve une organisation comportant une couche d'entrée, une couche cachée et une couche de sortie. Chaque neurone caché ne réagit ici qu'à une petite partie de l'espace d'entrée [16].

Pour un réseau comportant n entrées et m unités cachées, l'activation des neurones cachés est donnée par une fonction de type gaussienne (formule 2.9) (les fonctions d'entrée et d'activation sont confondues):

$$a_i = \exp \left(- \frac{1}{2} \sum_{k=1}^n \frac{(e_k - c_{k,i})^2}{\sigma_{k,i}^2} \right) = \prod_{k=1}^n \exp \left(- \frac{1}{2} \frac{(e_k - c_{k,i})^2}{\sigma_{k,i}^2} \right) \quad (2.9)$$

Où i désigne l'indice du neurone, k parcourt l'ensemble des entrées notées e_k , et c_{ki} et $\sigma_{k,i}^2$ sont des paramètres appelés respectivement centres et variances des gaussiennes. La figure 2.9 présente la forme de cette fonction d'activation pour un neurone possède une seule entrée.

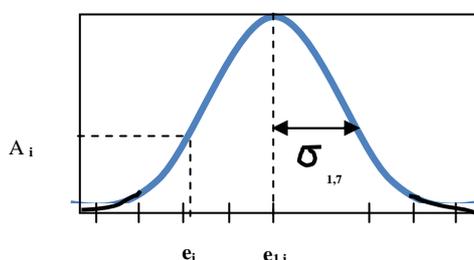


Figure 2.9: Fonction d'activation d'un neurone caché possédant une seule entrée

Chacun de ces neurones ne s'active donc de manière significative que pour des valeurs d'entrée relativement proches des centres des gaussiennes. Les connexions provenant des neurones d'entrée ne sont pas pondérées. L'activation d'un neurone de sortie d'indice i est donnée par :

$$\sigma_i = \frac{\sum_{j=1}^m w_{ij} a_j}{\sum_{j=1}^m a_j} \quad (2.10)$$

Où j parcourt l'ensemble des indices des neurones cachés. Les neurones de ce type réalisent donc une somme pondérée des valeurs d'activation des neurones cachés. Le terme $\sum_{i=1}^m a_i$ appelé facteur de normalisation n'est pas obligatoire. On parle de réseau normalisé lorsqu'il est employé.

L'apprentissage se fait dans ces réseaux par modification des poids des connexions entre les neurones cachés, les neurones de sortie, les centres et les variances des gaussiennes. On réalise comme précédemment une descente de gradient ayant pour but de minimiser l'erreur quadratique, dont l'expression est donnée par l'équation (2.11).

$$q = \frac{1}{2} \sum_i [a_i - s_i]^2 \quad (2.11)$$

Ce modèle souffre cependant d'un inconvénient par rapport aux réseaux Multicouches puisque contrairement à ceux-ci, son domaine d'approximation (i.e. domaine dans lequel il réalise une approximation satisfaisante) est strictement borné.

2.4 Réseaux Temporels

Peut-on distinguer un modèle qui utilise le temps et un modèle qui le traite ?

- Un modèle qui utilise le temps si son calcul se déroule dans le temps alors que la notion de temporalité est totalement absente du problème traité par l'architecture.
- Un modèle qui traite le temps s'il se préoccupe de problèmes d'ordre temporel en gardant d'une façon ou d'une autre la trace du temps.

2.4.1 Représentation du temps

Plusieurs travaux de recherche [6], [7], [10] et [14] sont consacrés aux réseaux de neurones temporels. Une Classification des réseaux de neurones temporels en fonction du degré d'intégration du temps est illustrée à la *figure 2.10* en dessous.

On trouve principalement deux familles de réseaux de neurones temporels:

- avec une représentation externe du temps,
- avec une représentation interne du temps.

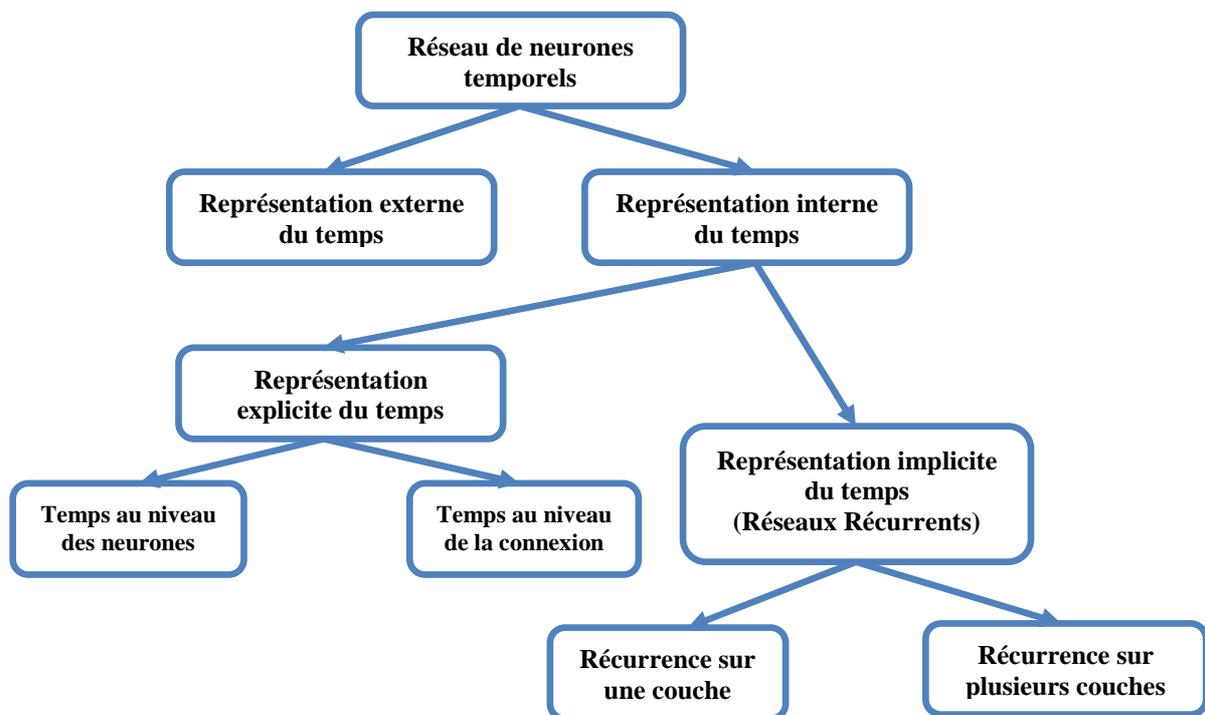


Figure 2.10: Classification des réseaux de neurones temporels

2.4.1.1 Représentation externe (spatiale) du temps

L'information temporelle contenue dans les données est alors transformée en une information spatiale, c'est à dire une forme qu'il s'agit de reconnaître. Dès lors, les techniques de classification par réseaux de neurones habituellement employées deviennent applicables.

Cette transformation du temporel en spatial s'obtient par l'utilisation classique de ligne à retard. Au lieu de présenter au réseau chaque événement, dès son apparition, il convient d'attendre un certain temps avant de procéder à la classification de la forme obtenue. Chaque retard temporel représente une dimension de la représentation spatiale.

Ce type de représentation du temps fait donc appel à un mécanisme externe qui est chargé de retarder ou de retenir un certain temps les données, ce qui conduit à l'appeler également *représentation externe* du temps.

La représentation spatiale du temps présente plusieurs désavantages. Tout d'abord elle suppose l'existence d'une interface avec le monde extérieur dont le rôle est de retarder ou de retenir les données jusqu'au moment de leur utilisation dans le réseau. Se pose ainsi le problème de la connaissance de l'instant où les données doivent être traitées c'est-à-dire de la définition de la taille de la fenêtre.

2.4.1.2 Représentation dynamique du temps

Les réseaux de neurones dits dynamiques traitent le temps d'une façon totalement interne au réseau. On peut distinguer deux manières de gérer le temps en interne : le temps est pris en compte implicitement ou explicitement. Dans les deux cas, le réseau possède la capacité de mémoriser des informations soit implicitement par la récurrence des connexions, soit explicitement par des retards au niveau des connexions.

a- Représentation explicite du temps dans les réseaux de neurones

- Représentation *explicite du temps au niveau des connexions*:

Les connexions à délais, une des représentations explicites du temps dans les réseaux de neurones est l'utilisation des retards aux niveaux des connexions. La différence entre ce type de réseau de neurones et les *réseaux récurrents* se situe au niveau du sens de la propagation du signal. Les réseaux à délais au niveau des connexions sont des *réseaux feedforward*.

- Représentation *explicite du temps au niveau des neurones*:

Le temps peut également être traité localement au niveau du neurone, ce qui permet d'avoir une certaine robustesse temporelle : deux entrées d'un neurone donné ne doivent plus être nécessairement synchrones pour avoir un certain effet.

b- Représentation implicite du temps : Réseaux de neurones récurrents

La connectivité des unités dans les réseaux de neurones récurrents ne se limite pas, comme dans le cas des réseaux à propagation avant, à des architectures dans lesquelles l'information se propage de l'entrée vers la sortie, couche après couche.

La prise en compte implicite du temps est réalisée en réintroduisant en entrée du réseau l'état précédent du réseau. C'est le cas des réseaux récurrents d'*Elman* et de *Jordan*.

2.4.2 Modèle de Jordan

Consiste à renvoyer la couche de sortie vers la couche d'entrée. Il s'agit d'un modèle avec une mémoire à un pas de temps puisque l'information de contexte concerne les classes reconnues : on ne conserve pas de trace des activités internes du réseau.

Dans ce cas, il est possible d'interpréter les valeurs de sorties comme des probabilités a posteriori, ce qui présente un intérêt pour les modèles hybrides.

2.4.2.1 Architecture

L'architecture de ce modèle est multicouche (*figure 2.11*), la dernière couche est rebouclée sur la première.

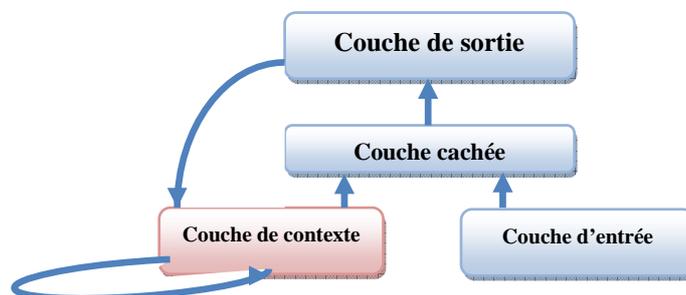


Figure 2.11: Réseau récurrent de Jordan

Les cellules d'entrée se répartissent en deux groupes : les cellules de plan et les cellules d'état. Les cellules de sortie sont rebouclées sur les cellules d'état par des connexions de poids fixes, de même pour les cellules d'état qui rebouclent sur elle-même par des connexions de poids fixes. L'état interne est défini par l'ensemble des cellules de sortie et des cellules de plan.

2.4.2.2 Comportement

Une forme d'entrée i est appliquée sur les cellules de plan. Ce vecteur, appelé vecteur de plan, ne sera plus modifié pour la suite de la production de la séquence. Le vecteur d'état des cellules de sortie varie dans le temps du fait des connexions récurrentes sur les cellules d'état. Ces connexions modifient les valeurs d'activation des cellules d'état et imposent une entrée variable dans le temps au réseau multicouche. En utilisant des vecteurs de plan différents, le même réseau apprend plusieurs séquences différentes.

2.4.2.3 Algorithme d'apprentissage

L'algorithme d'apprentissage est une généralisation de la *rétropropagation de gradient*. Plutôt que d'utiliser une valeur fixe pour les sorties désirées, des contraintes sont imposées sur ces valeurs. Une contrainte spécifique, par exemple, un intervalle pour la valeur de la sortie. Il y a un apprentissage après chaque forme produite.

2.4.3 Modèle d'Elman

C'est la couche cachée qui est renvoyée vers l'entrée. Il y a alors conservation de la trace des activités internes du réseau sur un pas de temps.

Une solution proposée consiste à définir une architecture hiérarchique dans laquelle des sous-réseaux récurrents sont utilisés pour identifier les différentes syllabes constituant un mot.

2.4.3.1 Architecture

Il s'agit d'une structure multicouche où les boucles de rétroaction relient la couche cachée avec les cellules d'entrée (*figure 2.12*).

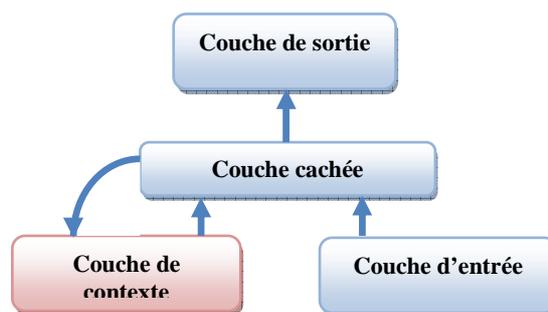


Figure 2.12: Réseau récurrent d'Elman

La couche d'entrée consiste en deux ensembles de cellules, les cellules de contexte et les cellules d'entrée. Il y a autant de cellules cachées que de cellules de contexte. Chaque cellule cachée est reliée à une seule cellule de contexte par une connexion fixe, de poids égal à 1. L'état interne est défini par l'ensemble des cellules de la couche cachée.

2.4.3.2 Comportement

La couche cachée a la tâche d'établir, à la fois, les associations de la fonction de sortie et les associations de la fonction de transition. A chaque instant, une sortie est calculée sur la base de l'état interne (contexte) et de l'entrée présente. Il y a une copie du vecteur d'activation de la couche cachée sur la couche de contexte. De cette façon, le résultat intermédiaire du traitement à la date précédente ($t-1$) peut influencer le traitement à la date actuelle (t).

2.4.3.3 Algorithme d'apprentissage

C'est la rétropropagation de gradient, les vecteurs d'activation de la couche cachée développés par le réseau sont significatifs du problème soumis. Ces vecteurs d'activation sont les représentations internes construites par le réseau.

2.5 Réseaux Neuro-Flous

2.5.1 Introduction

Pendant que les réseaux neuronaux sont intéressants pour reconnaître des modèles, ils ne peuvent pas expliquer comment ils atteignent leurs décisions. De même pour les systèmes de la logique floue qui peuvent raisonner avec l'information imprécise sont intéressants pour expliquer leurs décisions mais ne peuvent pas automatiquement acquérir les règles qu'ils utilisent pour prendre ces décisions. Pour cela, une approche a été développée qui est principalement connu comme approche neuro-floue.

2.5.2 Objectif

- Les réseaux neuro-flous sont nés de l'association des réseaux de neurones avec la logique floue, de manière à tirer profit des avantages de chacune de ces deux techniques.
- La principale propriété des réseaux neuro-flous est leur capacité à traiter dans un même outil des connaissances numériques et symboliques d'un système.
- Ils permettent donc d'exploiter les capacités d'apprentissage des réseaux de neurones d'une part et les capacités de raisonnement de la logique floue d'autre part.

2.5.3 Définitions

Le système neuro-flou est un système flou formé par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones. La technique d'apprentissage opère en fonction de l'information locale et produit uniquement des changements locaux dans le système flou d'origine.

L'intégration des réseaux de neurones et les systèmes d'inférence flous peut être exprimée à travers la figure 2.13.

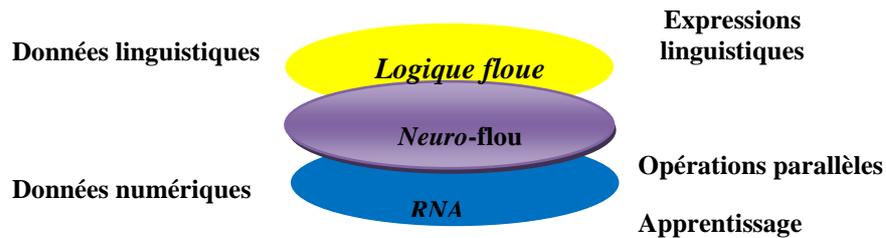


Figure 2.13: le système Neuro-Flou [6]

On peut résumer les avantages et les inconvénients des réseaux de neurones et de la logique floue dans le tableau 2.1:

Réseaux de neurones	Logique floue
Avantages	
<ul style="list-style-type: none"> ▪ Le modèle mathématique non requis ▪ Aucune connaissance basée sur les règles ▪ Plusieurs algorithmes d'apprentissage sont disponibles 	<ul style="list-style-type: none"> ▪ Le modèle mathématique non requis ▪ La connaissance antérieure sur les règles peut être utilisée ▪ Une interprétation et implémentation simple
Inconvénients	
<ul style="list-style-type: none"> ▪ Boite noire (manque de traçabilité) ▪ L'adaptation aux environnements différents est difficile et le réapprentissage est souvent obligatoire (sauf pour RBF) ▪ La connaissance antérieure ne peut pas être employée (apprentissage à partir de zéro) ▪ Aucune garantie sur la convergence de l'apprentissage. 	<ul style="list-style-type: none"> ▪ Les règles doivent être disponibles ▪ Ne peut pas apprendre Adaptation difficile au changement de l'environnement ▪ Aucune méthode formelle pour l'ajustement

Tableau 2.1: Comparaison entre la logique floue et les réseaux de neurones [6], [16]

2.5.4 Le système neuro-flou hybride

Les approches neuro-floues modernes sont de cette forme : Un réseau neuronal et un système flou combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous.

Dans ce modèle, les algorithmes d'apprentissage des systèmes de neurones sont utilisés pour déterminer les paramètres du système d'inférence floue (SIF).

Le système neuro-flou partage la structure des données et la représentation de connaissance. Le SIF peut utiliser l'expertise humaine pour implémenter les composantes essentielles dans la base de connaissance et la base de données et exécute le raisonnement flou pour inférer la valeur de sortie totale.

2.5.5 Architectures neuro-floues

Diverses associations des méthodes et architectures neuro-floues ont été développées depuis 1988 [6], [14], [16], [22]. Pour cela, nous présentons brièvement les différents modèles neuro-flous qui font usage des complémentarités de réseaux neuronaux et systèmes d'inférence flou dans la section suivante. Nous citons quelques travaux dans ce domaine comme NEFCLASS, ANFIS, NEFCON, NEFPROX et NEFDIAG.

➤ **NEFCLASS (Neuro-Fuzzy CLASSification)**

Modèle utilisé généralement en classification. Il est constitué de 3 couches : une couche d'entrée avec les fonctions d'appartenance, une couche cachée représentée par des règles et une couche de sortie définissant les classes *Figure 2.14*.

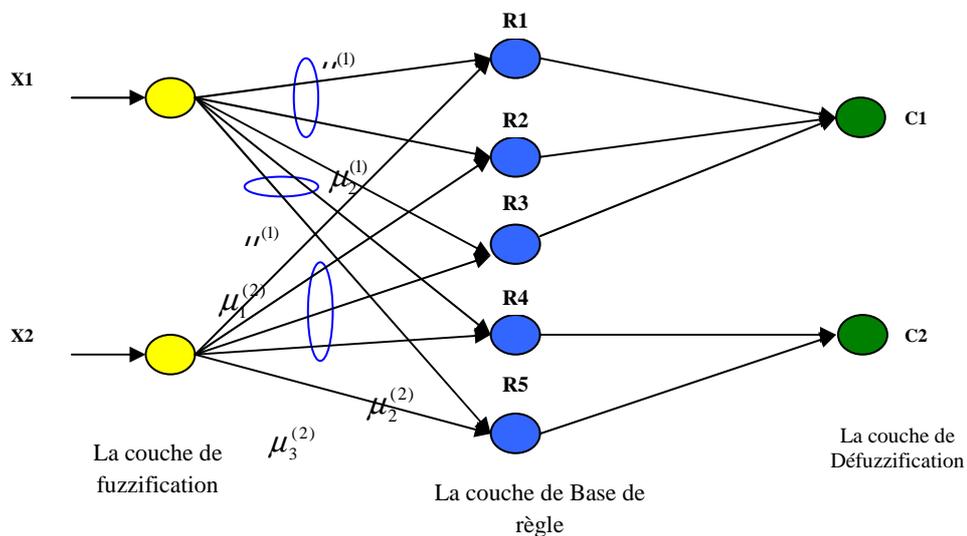


Figure 2.14: Architecture NEFCLASS

Ce modèle est facile à mettre en application, il évite l'étape de défuzzification, tout en étant précis dans le résultat final, avec une rapidité bien supérieure aux autres architectures.

➤ **ANFIS (Adaptive-Network-based Fuzzy Inference System)**

ANFIS représente un système à inférence flou mis en application dans le cadre des réseaux adaptatifs. Il utilise la procédure d'apprentissage Hybride (*Figure 2.15*).

Cette architecture affine les règles floues obtenues par des experts humains pour décrire le comportement d'entrée-sortie d'un système complexe. Il est implémenté dans la boîte à outils « Neuro-Fuzzy » de MATLAB. Ce modèle donne de très bons résultats en poursuite de trajectoire, approximation non linéaire, commande dynamique et traitement du signal.

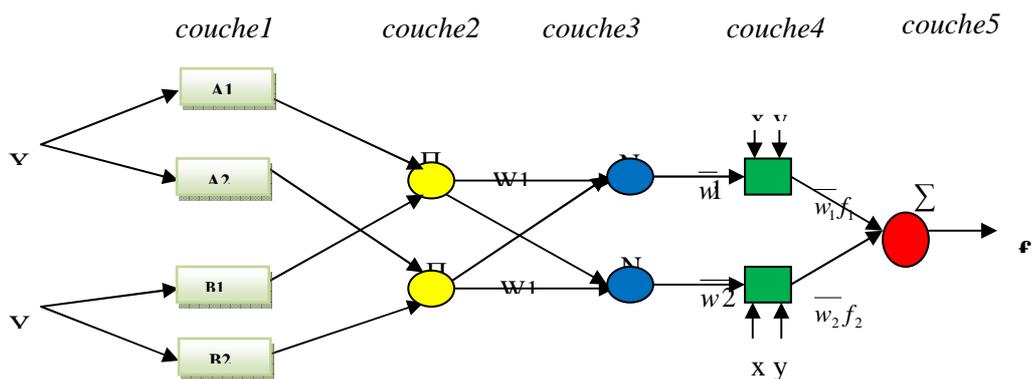


Figure 2.15: Architecture ANFIS

➤ **NEFCON (Neuro-Fuzzy Controller)**

Modèle formé de 3 couches. Une couche cachée formée par des règles, une couche d'entrée incluant les nœuds d'entrée avec les sous-ensembles flous d'antécédences et une couche de sortie avec un nœud de sortie et les sous-ensembles des conséquences. L'élaboration des règles est similaire à l'architecture NEFCLASS, avec une différence en sortie. Cette architecture est généralement utilisée en approximation de fonctions et en contrôle flou (Figure 2.16).

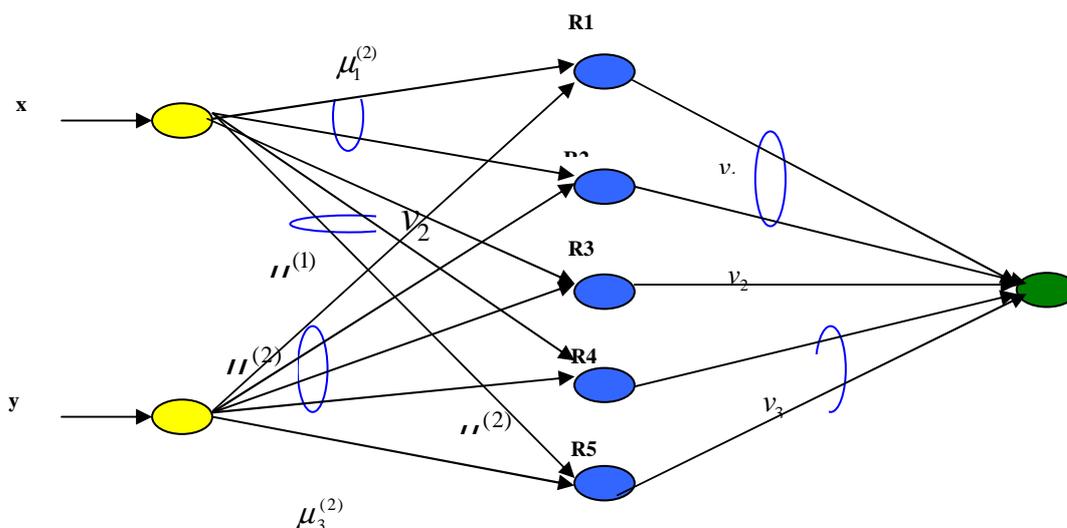


Figure 2.16: Architecture NEFCON

➤ **NEFPROX (Neuro Fuzzy function apPROXimator)**

Modèle obtenu par l'association des deux architectures: NEFCLASS et NEFCON, il est utilisé dans différentes applications comme la classification et l'approximation de fonctions. Le NEFCLASS utilise un algorithme supervisé pour définir les règles floues, le NEFCON utilise un algorithme d'apprentissage non supervisé avec le calcul de l'erreur de sortie. Les deux modèles emploient la rétro propagation afin de définir les sous-ensembles flous.

Comparé au modèle ANFIS, NEFPROX (figure 2.17) est beaucoup plus rapide, mais ANFIS donne de meilleurs résultats en approximation.

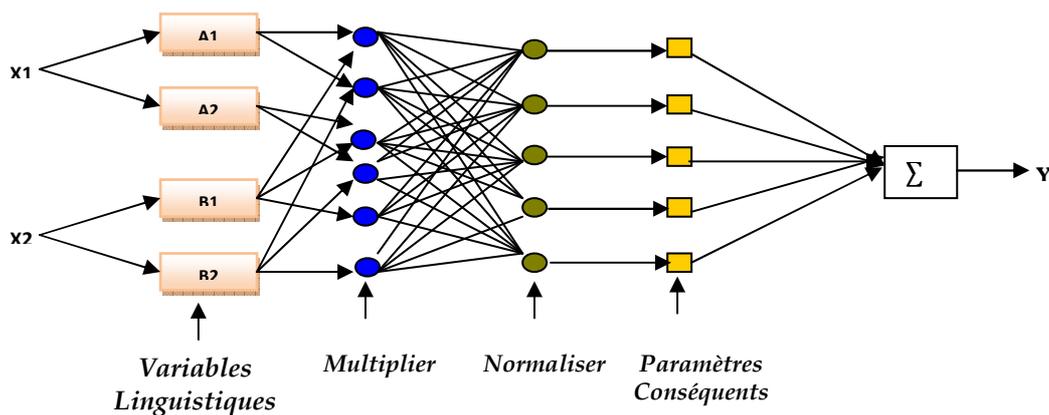


Figure 2.17: Architecture NEFPROX

Le NEFPROX est le premier système interprétable et lisible, dédié à l'approximation de fonction. Néanmoins, ses résultats en classification restent moins bons que ceux donnés par le NEFCLASS.

➤ NEFDIAG (*Neuro-Fuzzy DIAGnosis*)

NEFDIAG est modèle Neuro-Flou de trois couches, fait son apprentissage par un ensemble de formes, telle que chaque forme soit affectée (classée) vers une des classes prédéfinies. La figure 2.18 illustre un système neuro-flou représenté par NEFDIAG avec N entrées, M règles floues et P sorties tel que les entrées représentent les observations de chaque composant du système à surveiller, donc pour chaque composant, il y a une ou plusieurs observations.

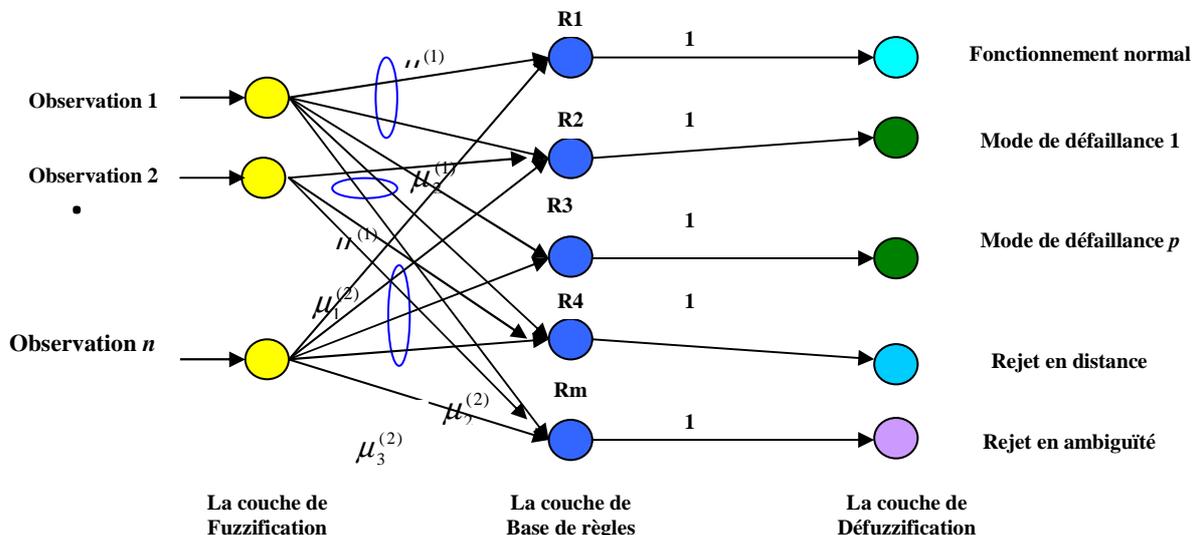


Figure 2.18: Architecture NEFDIAG

2.5 Conclusion

Il existe différents critères de classer les types de réseaux de neurones. Mais dans cette étude, nous nous sommes limités aux réseaux de neurones bouclés, car ils s'y prêtent le mieux à notre application. Le Réseau Neuro-flou d'Elman est l'architecture créée pour le pronostic industriel.

ANALYSE ET SPECIFICATION DU NOUVEAU SYSTEME

La phase d'analyse est une étude du problème et de ce qui existe. Pour cela, nous présentons brièvement dans cette partie la Société des CIMents d'Ain-Touta-BATNA, le processus de fabrication des ciments et plus précisément la « zone de clinkérisation ». Ensuite, nous exposons le positionnement du projet et les exigences fonctionnelles, on les regroupées dans un cahier des charges fonctionnel. Ce dernier est un document par lequel le demandeur exprime fonctionnellement son besoin.

3.1 Introduction

Toute étude doit commencer par ce qui existe, seul l'examen du problème à résoudre permet de définir de manière réaliste les objectifs à atteindre. Pour cela, nous commençons ce chapitre par une présentation générale du champ d'étude « SCIMAT-Batna ».

Ensuite et d'après la communication et l'interview orales auprès le chef de département de production, les ingénieurs de production et les opérateurs de la zone de cuisson, et l'étude du système actuel ont mené à déterminer les besoins essentiels attendus, nous proposons un cahier des charges fonctionnel pour le nouveau système.

3.2 Etude de l'existant

L'existant est le point de passage qui matérialise le premier contact des concepteurs avec un domaine ignoré. De plus, il est essentiel que les concepteurs parviennent à une vue claire des besoins qui figurent dans le double but visé par cette étape à savoir :

- Prendre connaissance avec le domaine dont l'entreprise souhaite améliorer le fonctionnement.
- Recenser l'ensemble des objectifs que poursuit l'entreprise concerne ce domaine.

3.2.1 Présentation de l'entreprise

L'industrie du ciment occupe une place prépondérante dans les économies de toutes les nations puisqu'elle est à la base du développement de secteurs vitaux dans l'économie des pays, par la production des matériaux les plus usités.

3.2.1.1 Identité

- *Raison Sociale* : SCIMAT (Société des Ciments d'Ain-Touta), Filiale du Groupe ERCE.
- *Forme Juridique* : Société par Action (SPA).
- *Siège sociale* : Rue Benflis –BP : 67-05000 –El –Boustène, Batna -ALGERIE

3.2.1.2 Situation Géographique

La Société des Ciments de Ain Touta (SCIMAT) se compose d' :

- Une Direction Générale : située à Batna.
- Une Unité Cimenterie: située dans la commune de Tilatou, Daïra de Seggana, à 50 Km à l'ouest de Batna elle couvre une superficie de 20 hectares et implantée sur l'axe routier Batna-Bari ka (Route Nationale N° 5) et sur la rocade ferroviaire Ain-Touta-M'sila.

3.2.2 Etapes de Fabrication du ciment

Le Ciment produit par la cimenterie Ain-Touta est fabriqué à partir d'un processus à voie sèche (*figure 3.1*).

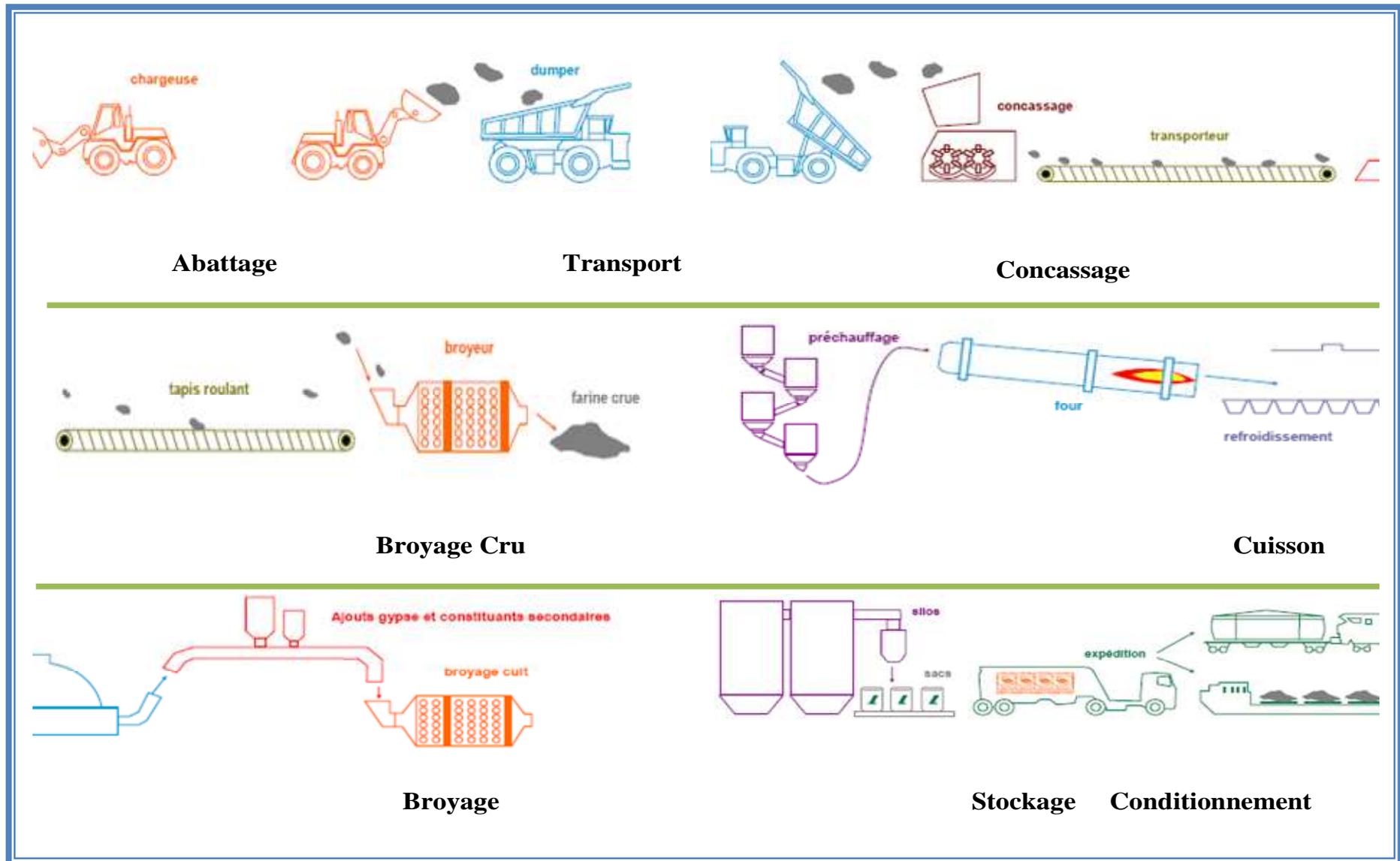


Figure 3.1 : Schéma synoptique du process de fabrication du ciment

3.2.2.1 Matières premières

- *Calcaire*: Roche constituée principalement de carbonates de calcium (75 à 92% de CaCO_3) extraite de la carrière par abattage à l'explosif.
- *Argile* : Roche constituée principalement de silice, d'alumine et de fer extraite du Gisement par ripage.
- *Minerai de fer* : Roche qui se rencontre le plus souvent sous forme d'oxyde de fer (Fe_2O_3) et qui joue un rôle utile de fondant.
- *Gypse*: Roche constituée principalement des sulfates de calcium ($\text{CaSO}_4 \cdot 2\text{H}_2\text{O}$) et qui joue un rôle utile de retardateur de prise.
- *Pouzzolane*: Roche d'origine volcanique constituée principalement de silice, utilisée comme ajout actif.

3.2.2.2 Exploitation des carrières

- L'abattage du calcaire par l'utilisation des explosifs.
- L'extraction des argiles par ripage au bulldozer.
- Le chargement par chargeuses pneumatique.
- Le transport du calcaire depuis le gisement jusqu'à l'installation de concassage est assuré par dumper et alors que le transport de l'argile est assuré par camion à benne.
- Le dosage des matières premières s'effectue selon les recommandations du Service contrôle qualité (laboratoire).

3.2.2.3 Stockage et reprise des matières premières

- Mise en stock des matières : Les transporteurs déversent les matières par les deux stackers (équipement de stockage) pour la réalisation de la mise en tas des matières dans les halls.
- Le mélange du calcaire et d'argile, qui est extrait du stock par pont-gratteur est amené par transporteur à bande à la trémie de l'atelier de broyage du cru.
- Le calcaire pur et le minerai de fer pour le broyage cru sont repris du hall de stockage par gratteur portique, et sont amenés aux trémies par transporteurs à bandes.
- Le calcaire pur et le gypse sont repris par gratteur portique, et sont amenés à l'atelier de broyage de clinker.

3.2.2.4 Broyage cru

- Les broyeurs crus sont alimentés par trois trémies de 250M^3 , chacune destinée respectivement pour : mélange, calcaire et minerai de fer.
- L'extraction des matières se réalise au moyen des extracteurs à tablier métalliques. Le dosage des matières qui alimentent les broyeurs s'effectue par transporteurs à bandes peseuses. Le broyage des matières dosées constitue une farine crue. Cette dernière est amenée par un élévateur à un séparateur statique. La matière grossière est recyclée vers le broyeur.
- La matière fine est récupérée dans deux cyclones puis transportée par aéroglisseur et un système élévateur de farine (Air lift) jusqu'aux silos d'homogénéisation pour stockage.

3.2.2.5 Cuisson

- L'extraction de la farine homogénéisée dans les silos suivant un système CLAUDUS PETERS s'effectue par des extracteurs alimentant deux élévateurs, deux aéroglisteurs.
- La farine crue est amenée à deux trémies tampons de 65 M³ chacune.
- Le système d'alimentation des fours placé sous les trémies est de fabrication SCHENK, comportant pour chaque four deux sorties avec débitmètres.
- La farine crue dosée passe par deux air-lifts qui la transportent en haut des préchauffeurs à cyclones.
- Chaque tour de préchauffage est constituée de quatre étages de cyclones :
 - L'étage supérieur comporte un cyclone double de 4,35 m de diamètre.
 - Les trois étages au dessous comportant chacun un cyclone simple de 6,6m de diamètre.
- La farine crue alimentée aux préchauffeurs subit une décarbonatation partielle suite échange thermique à contre courant à des températures variant de 320°C - 850°C selon étages cyclones.
- Le long du four, la farine passe par les transformations suivantes par l'utilisateur du gaz naturel comme combustible : calcination, transition, cuisson et refroidissement.
- Le clinker produit à une température jusqu'à 1400°C est refroidit dans des refroidisseurs à ballonnets.
- Le clinker est transporté depuis les refroidisseurs par les transporteurs à augets jusqu'aux silos de stockage de Clinker (03 silos de 15000 Tonnes chacun).
- Le système de transport clinker présente la possibilité d'évacuer le clinker de mauvaise qualité dans une trémie de 90 M³.
- Le clinker fabriqué et répondant aux exigences de la qualité produit est stocké dans les silos.

3.2.2.6 Broyage Clinker

- L'extraction du clinker des silos de stockage se réalise par des alimentateurs vibrants qui alimentent des transporteurs à bandes pour remplissage des trémies clinker.
- Les broyeurs sont alimentés par 04 trémies de 280m³ chacune avec deux sorties destinée respectivement pour : clinker, gypse, ajouts. Les deux dernières trémies sont alimentées à partir du hall de stockage des ajouts calcaire et gypse.
- L'extraction des matières des trémies s'effectue aux moyens des bandes doseuses. Le dosage des matières qui alimentent les broyeurs à ciment à circuit fermé se réalise par les dominats.
- Le broyeur est équipé d'un système d'injection d'eau au niveau des deux compartiments ce qui permet de maintenir la température de ciment à l'intérieur du broyeur au dessous de 115°C et cela pour éviter la transformation du gypse en plâtre.
- Le ciment à la sortie du broyeur est amené par un élévateur et une aéroglistière à un séparateur dynamique.

3.2.2.7 Expéditions

Le ciment est extrait des silos à l'aide d'un système de fluidisation par air, ces silos sont répartis comme suit : 03 Silos de 8000 tonnes chacun au niveau de la voie route et 02 Silos de 8000T chacun au niveau de la voie ferrée.

3.2.3 Système de contrôle, contexte existant

3.2.3.1 Système FLS-ACE-625

Ce système est constitué de 04 fonctions principales à savoir :

- Communication de l'opérateur.
- Système de contrôle moteur FLS -625.
- Système traitement de mesure 410.
- Système de traitement d'alarmes FLS-575.

3.2.3.2 Système de régulation FLS 470

- C'est un système PID pour la régulation de processus par voie manuelle, automatique ou par ordinateur.
- Le régulateur PID type FLS -470 est un régulateur électronique d'application générale destiné aux boucles de régulation analogiques.

3.2.3.3 Système de contrôle FLS-325

C'est un système de contrôle moteur FLS-325 logique à relais. Il est basé sur 03 fonctions principales à savoir :

- Communication de l'opérateur : En exploitant le diagramme synoptique et les enregistreurs des paramètres de marche.
- Contrôle de séquences de démarrage, d'arrêt et des verrouillages de marche.
- Traitement des alarmes.

3.2.3.4 Système de conduite S.D.R

L'exploitation de la cimenterie se fait par un système SDR (Supervision -Dialogue Reports)

- Surveillance (supervision) : Porte sur le traitement des valeurs des types analogiques et numériques. Les variables du processus qui sont introduites par l'opérateur sont traitées sous formes de point. Chaque point est identifié par le système et représente à tout moment l'information valable sur l'état de la chaîne de fabrication à savoir.
- Dialogue : La communication entre l'opérateur et le système se fait via un écran d'affichage est comporte sur :
 - Ecriture des rapports d'usine sur demande.
 - Introduction des valeurs du processus spécifiées par l'opérateur.
 - Affichage et sélection de certains nombre de points montrant code, texte descriptifs valeurs.
 - Modification du seuil d'alarme, des points de consignes et des paramètres de marches.
 - Rapports : rapport d'usine, rapport d'alarme, rapport de communication de l'opérateur, aperçu d'alarmes, journaux de déroulements, rapport de production.

3.2.3.5 CEM-SCANNER (Système de Contrôle des Réfractaires)

C'est un système basé sur un détecteur de rayons Infra Rouge à balayage ultrarapide inclus dans le logiciel FLS/SDR.

Il comporte le contrôle de deux éléments de base :

- Détermination de l'état du revêtement réfractaire et la mesure.
- Instantanée et continue de niveau de température de l'ensemble du four. Il fournit à l'opérateur les informations ci-après :
 - Détection immédiate de points chauds et alarmes.
 - Enregistrement de courbes de températures de toute la surface spécifiée du four.
 - Profils des températures MAX-MIN et Moyenne du four.
 - Présentation tridimensionnelle du revêtement du réfractaire et du croûtage.
 - Détection des formations d'anneaux.

3.2.3.6 Système de Contrôle et de Conduite (QCX)

- Le système dosage et de contrôle de qualité *QCX* comprend : un ordinateur et un générateur.
- Un spectromètre à rayon X à commande automatique pour analyse des routines suivantes : Mélange calcaire/argile, Farine crue, Minerai de fer, Gypse, SO₃ dans le ciment.
- Les résultats de ces analyses sont convertis en pourcentage d'oxydes permettant de calculer les modules de farine : MA, MS et LSF (*voir annexe B*).
- Le système QCX contrôle automatiquement la composition chimique de la farine crue basée sur les analyses périodiques d'échantillonnage soit automatique ou manuelle.
- Le réglage des points de consignes des alimentateurs cru se fait automatiquement ou manuellement pour réglage des proportions de matières (mélange, minerai de fer & calcaire) afin d'obtenir la qualité exigée de farine crue.
- Rapports : le système donne des rapports journaliers des différents types d'analyses, des alarmes du système QCX sur la manipulation de l'opérateur.

3.2.3.7 SDR/FUZZY LOGIC

F.L. SMIDTH a mis au point un système de contrôle de four. Le contrôle d'un four présente certains problèmes en raison de la complexité du processus, des temps de réponse ainsi que des possibilités de mesure limitées. Le degré d'automatisation du contrôle d'un four a été relativement faible, et c'est pourquoi la marche satisfaisante du four dépendait en grande partie de l'expérience et de la compétence des opérateurs.

Le système est basé à son contrôle aux quelques paramètres essentielles (*voir Annexe D*)

Le contrôle par *Fuzzy Logic* est utilisé au cours de la période de mise en marche du four durant laquelle l'opérateur suit attentivement le déroulement de la conduite automatique, et en ajoutant, retirant ou modifiant les règles de conduite au moyen du pupitre de commande, il « apprend » à l'ordinateur la conduite correcte du four. Lorsque l'opérateur est convaincu que la conduite du four se fait de manière adéquate, il peut le laisser fonctionner en automatique.

3.3 Cahier des Charges Fonctionnel

3.3.1 Définition

Le Cahier des Charges Fonctionnel (CdCF) est le document par lequel le demandeur exprime Fonctionnellement son besoin ou celui qu'il est chargé de traduire. Donc, il est un préalable à tout projet informatique.

Les objectifs d'un *CdCF* sont les suivants :

- Définir les objectifs que doit atteindre la solution.
- Indiquer les contraintes à respecter impérativement.
- Etre un outil de dialogue entre les différents acteurs.
- Diminuer les risques d'erreur lors de la réalisation ou l'installation.

3.3.2 Description du projet

C'est un projet du LAP-BATNA, consacré à résoudre le problème de classification des modes de défaillances par un système Neuro-Flou dans le cadre du pronostic industriel. Donc, nous disposons d'un nombre des défaillances appartenant à différentes catégories et nous souhaitons créer un système en temps réel permettant de trouver automatiquement les dégradations et les classer d'une manière claire, précise, efficace et partageable par tous les acteurs de l'équipe de production.

3.3.3 Positionnement

Le futur système sera exploité au sein de l'atelier de clinkérisation de la SCIMAT- Batna.

La question qui se pose : Pourquoi le four ?

- ✚ Chacune des opérations impliquées dans le processus de fabrication de ciment est importante et doit être correcte, faute de quoi le ciment risque de ne pas avoir la qualité exigée pour son emploi. Cependant la cuisson est sans doute l'opération la plus sensible et la plus importante. En effet, la cuisson ou la clinkérisation est l'opération au cours de laquelle s'effectuent les réactions chimiques essentielles. Elle détermine en grande partie la qualité du ciment. Les autres opérations y participent également, mais il n'est pas toujours possible de remédier aux défauts de la cuisson.
- ✚ D'après les travaux de collègues [5] et [27], on peut déduire que l'installation de cuisson est le système le plus important et le plus sensible aux défaillances d'où il est déterminant de la qualité du ciment à produire.
- ✚ Ainsi que la complexité des processus physiques et réactions chimiques qui se produisent à l'intérieur du four de cimenterie rend difficile l'automatisation de sa conduite La plupart des échanges qui se produisent à l'intérieur du four ne sont pas directement mesurables. Cependant, il est indispensable pour le contrôle commande de trouver un paramètre qui soit le reflet de ces échanges.
- ✚ Aussi, d'après l'étude statistique [33] faite pour déterminer l'équipement le plus défaillant, le four enregistre le plus fort taux de pannes annuel de 29.20%. Ces données sont illustrées sur le *tableau 3.1* des taux de pannes des équipements de la chaîne technologique suivant:

N°	Désignation d'équipement	Le taux de pannes	Nombre de pannes/an
01	Concasseur 02	6.19%	7 pannes
02	Concasseur 03	0.87%	1 panne
03	Concasseur 06	2.65%	3 pannes
04	Aéroglossière	11.5%	13 pannes
05	Bouche Vrac	13.27%	15 pannes
06	Armoire Bruleur à gaz	6.19%	7 pannes
07	Armoire générateur chaleur	5.31%	6 pannes
08	Elévateur	5.31%	6 pannes
09	<i>Four</i>	<i>29.2%</i>	<i>33 pannes</i>
10	Moteur électrique	2.65%	3 pannes
11	Broyeur Ciments	16.81%	19 pannes

Tableau 3.1: Taux de pannes des équipements de la chaîne technologique

Alors, le taux de pannes de four est toujours le plus élevé. Ce qui implique le sens et l'intérêt que nous accordons à cela. Il est important de rappeler que le four est considéré comme étant le poumon de toute installation.

3.3.4 Caractéristiques de l'installation

✚ Caractéristiques du Four rotatif

- Type du four : Four rotatif UNAX à 04 étages de préchauffeurs
- Capacité journalière : 2X1500 t/j de clinker
- Diamètre : 4,55 mètres
- Longueur : 68 m
- Vitesse de rotation : 2,0 tr/mn
- Revêtement : Réfractaire à base d'alumine et de magnésie chrome.
- Température de cuisson : Jusqu'à 1400°C
- Température Clinker sortie refroidisseurs : de 150 °C à 200 °C

✚ Système d'alimentation

Type : Système SCHENK ; Capacité d'alimentation : 120 t/h

✚ Refroidisseurs à ballonnets

- Types de refroidisseurs : Refroidisseurs UNAX
- Nombre de ballonnets : 09
- Longueur ballonnet : 19,8 m
- Diamètre ballonnet : 2,10 m
- Revêtement : Réfractaire à base d'alumine + Acier réfractaire + des releveurs.

✚ Caractéristiques de farine alimentée aux fours

- Chimique : facteur de saturation en chaux LSF (voir annexe B).
- Physique : finesse farine (14% sur tamis 90/μ), taux d'humidité (Max.0, 5% H₂O), densité (0,8).

Caractéristiques du clinker

- Granulométrie : 0-25 mm ;
- Température clinker 150 °C - 200 °C sortie refroidisseurs à ballonnets.

3.3.5 Composition chimique des produits

La *Farine* est constituée d'un mélange finement broyé en proportions correctes de calcaire, argile et minerai de fer apportant les quatre oxydes essentiels, à savoir : Chaux CaO (notée C) sous forme essentiellement de carbonate de calcium CaCO₃ ; Silice SiO₂ (S), Alumine Al₂O₃ (A); Oxyde de fer Fe₂O₃ (F).

Le *clinker* est un constituant du ciment, qui résulte de la cuisson d'un mélange composé d'environ 80 % de calcaire et de 20 % d'argile : la "farine" ou le "cru". Cette cuisson, la clinkérisation, se fait à une température d'environ 1450°C.

Le clinker se présente sous la forme de nodules durs et cristallisés, de teinte gris foncé pour les ciments habituels et verte pour le clinker de ciment blanc. La composition des clinkers gris est en moyenne :

- Silicate tricalcique (SiO₂ – 3 CaO): 50 à 65%
- Silicate bicalcique (SiO₂ – 2 CaO) : 15 à 20%
- Aluminate tricalcique (Al₂O₃ – 3CaO) : 5 à 15%
- Ferro-aluminate tétracalcique (Al₂O₃Fe₂O₃ – 4 CaO) : 5 à 10%

La composition minéralogique du clinker obtenu après cuisson des matières crues est fonction de la composition du mélange, de la température et du temps de cuisson et des conditions de refroidissement.

3.3.6 Expression des besoins

Le nouveau système devra regrouper toutes les fonctionnalités nécessaires de collection, de gestion, de traitement et de publication. L'exploitation du nouveau système représentera un outil de surveillance et de diagnostic prédictif des dégradations du système de production. Donc, l'objectif principal est d'assurer la gestion efficace du four. Aussi, d'éliminer les variations de forme de conduite par différents opérateurs, assurant ainsi une conduite stable et uniforme du four.

3.3.7 Exigences fonctionnelles

Le nouveau système doit répondre aux besoins cités précédemment. D'après la communication et l'interview orales auprès le chef de département de production, les ingénieurs de production et les opérateurs de la zone de cuisson, et l'étude du système actuel ont mené à déterminer les besoins essentiels attendus :

- Conduite et gestion du four.
- Contrôle des pré-alarmes et des alarmes.
- Dialogue : la communication entre l'opérateur et le système se fait via un écran d'affichage est comporte sur :
 - Alimentation du four : homogénéisation et alimentation de la trémie du Schenck.

- Préchauffement et cuisson : précalcination, tirage de gaz et cuisson.
- Refroidissement et transport clinker.
- Introduction des valeurs du processus spécifiées par l'opérateur.
- Modification du seuil d'alarme, des points de consignes et des paramètres de marches (température, pression, débit,...etc.).
- Rapports : les rapports dans le système comprennent : rapport d'usine, rapport d'alarme, rapport de communication de l'opérateur, aperçu d'alarmes, journaux de déroulements, rapport de production.

3.3.8 Exigences non fonctionnelles

Pour répondre aux besoins du nouveau système et aux demandes du chef de projet et de l'équipe de production, il est important de répondre aux exigences de qualité et de performance suivantes :

- ✚ L'ergonomie du nouveau système s'intègre et obéit aux règles classiques et connues de Windows. Autrement, l'utilisateur découvrira la convivialité du logiciel et l'accès à ces différentes fonctionnalités par l'intermédiaire des menus et des boîtes de dialogues et des images parlantes.
- ✚ La mise en forme doit faciliter au maximum la navigation à l'aide d'une présentation claire et intuitive, et doit éviter toute complexité.
- ✚ Formulaire de commande simple : la conception et la présentation de celui-ci doit donc particulièrement être soignés pour ne pas rebuter l'utilisateur.
- ✚ Le temps de réponse du système doit être le plus court possible.

3.3.9 Présentation de la Conduite Future du Four

La conduite automatique de l'atelier de cuisson sera faite à base du système neuro-flou. Ce dernier comprend en outre les modules fonctionnels suivants : supervision, diagnostic prédictif, traitements des pré-alarmes et alarmes, dialogue et rapports. Pour assurer un fonctionnement correct et une protection adéquate des équipements électriques et mécaniques, la commande comporte des boucles de régulation automatique et des verrouillages.

Les paramètres essentiels dont il faut tenir compte sont les suivants :

- Composition chimique des matières amenées au four et débit d'alimentation,
- Débit du combustible et combustion,
- Qualité du clinker,
- Rendement du préchauffeur à cyclones,
- Rendement du refroidisseur.

La zone de cuisson contient des analyseurs qui suivent la farine crue durant son transformation en clinker. Des échantillons sont extraits d'une manière manuelle ou automatique selon le programme illustré dans le *tableau 3.2* suivant:

N°	Nature échantillon	point de prélèvement	Traitement en labo	Type d'analyse	contrôle	fréquence
1	farine crue	en aval de broyeurs avant les silos	séchage	%H2O tamisage rayons X	fonctionnement broyeurs alimentation LSF/MS/MA	1 h
2	farine crue	après la trémie tampon du four	séchage	rayons X	alimentation LSF/MS/MA	1/4 h
3	clinker	sortie refroidisseur	broyage séparation	poids au litre CaO libre rayons X	degré de cuisson contrôle qualité	1h 1/2h 1/24h
4	gaz de fumée	four entrée	analyse sur place	Analyse O2,CO,CH4 Analyse Orsat O2, CO,CH4, Nox	contrôle combustion surveillance analyse automatique	continu selon besoin
5	gaz de fumée	après cyclone		Analyse O2,CO,CH4	contrôle combustion	continu
				Analyse Orsation O2, CO,CH4, Nox	surveillance analyse automatique	continu
6	poussière électrofiltre	sortie électrofiltre		rayons X	Contrôle accumulation K, Na, S, Cl	occasionnel

Tableau 3.2: Programme d'échantillonnage

Plusieurs paramètres sont utilisés dans la surveillance du procédé de clinkérisation, parmi eux ceux qui sont des paramètres essentiels :

- La concentration NO_x dans le gaz de sortie,
- Le débit de combustible,
- La forme de la flamme,
- La vitesse de rotation du four.

Les conditions de la chauffe dans le précalcinateur et le fonctionnement du préchauffeur sont surveillées par la mesure des températures et des pressions. L'alimentation du four est régulée par des alimentateurs-presseurs qui sont généralement synchronisés avec la vitesse de rotation du four comme l'indique le *tableau 3.3* suivant :

vitesse rotation (t/mn)	Alimentation farine (tonnes)	GAZ Nm3			Clinker (tonnes)	CCS Kcal/Kg K	J1/2J01S1 (%)
		V01	V31	Gaz T			
1	58	3280	578	3858	788	1051	50
1,1	63	3432	606	4038	866	1000	54
1,2	69	3632	642	4272	945	970	56
1,3	75	3734	659	4393	1024	720	58
1,4	81	3890	687	4577	1103	890	60
1,5	86	4072	718	4790	1181	870	62
1,6	92	4245	749	4994	1260	850	64
1,7	98	4458	787	5245	1339	840	68
1,8	104	4664	824	5488	1418	830	70
1,9	109	4862	858	5720	1496	820	72
2	115	5056	893	5949	1575	810	75

Tableau 3.3: Maitrise de la proportionnalité entre Vitesse de rotation, Alimentation Four et Consommation Gaz & Tirage

La détermination de l'état du revêtement réfractaire et la mesure de la température de l'ensemble du four sont réalisés par un détecteur de rayons Infra Rouge à balayage ultrarapide. Ce dernier détecte d'une façon immédiate de points chauds et enregistre les courbes de températures de toute la surface spécifiée du four (*Annexe B*).

Développer un programme spécifique en fonction du nombre et des caractéristiques des composants du mélange (aussi du type de clinker à obtenir). En utilisant les résultats obtenus sur les paramètres *LSF*, *MS* et *MA* par FLS Automation (*Annexe B*).

Aussi, les différents paramètres essentiels pour la surveillance du procédé de production sont résumés dans le *tableau 3.4* ci-dessous :

Paramètres W1 (four)	Unité	Intervalle			Déviations
		MAX	PC	MIN	
A54T1	C°	860	84 0	820	>860
					<820
A50A1	%	0,25	0,1	0	>0,25%
A50A2	%	0,1	0	0	>0,1%
A50A3	%	5	4	2	>5%
					<2%
K01P1	mbar	PC<1			>1
W01T4	C°	1400	13 00	1200	>1400C°
					<1200C°
U01T1	C°	150	13 0	130	>150C°
					<130C°

Tableau 3.4: Paramètres de contrôle du four

Le système devra permettre à l'utilisateur de clair et référencé. L'application comprendra principalement plusieurs modules accessibles à l'aide de menus :

- Un module de supervision du process,
- Un module de pronostic des dégradations,
- Un module de visualisation des pré-alarmes et des alarmes,
- Un module de sélection des équipements,
- Un module de la logique floue (FUZZY CONTROL).
- Un module de neuro-flou (NEURO-FUZZY SYSTEMS).

3.4 Conclusion

L'étude et le suivi d'un comportement d'un système, soit pour l'automatiser ou soit pour l'améliorer, ont besoin de connaître tous ce qui existe. Le fruit de cette étude que nous avons fait au sein de la cimenterie donne lieu un cahier des charges fonctionnel. C'est un résumé (extrait) du cahier des charges technique proposé par le concepteur, contenant seulement les points nécessaires pour la zone de cuisson du Four.

ENVIRONNEMENT DE TRAVAIL

Dans ce chapitre, une présentation détaillée des différents outils utilisés pour l'implémentation d'un nouveau système neuro-flou. Puisque le processus de fabrication du ciment d'Ain-Touta-Batna dans son implantation utilise les automates programmables, on a choisi le langage STEP7. Ce dernier a l'avantage d'exploiter d'autres logiciels et outils comme le logiciel de la supervision WINCC. Aussi, la solution neuro-floue nécessite les outils FuzzyControl et NeuroSystems. Le nouveau système neuro-flou permet de classifier les défaillances analysées et spécifiées par la méthode AMDEC.

4.1 Introduction

Une installation et configuration matérielle est nécessaire pour réaliser le système étudié. Ainsi que plusieurs outils sont élaborés pour cette démarche : l'outil Matlab puisque contenant un module de neuro-flou est utilisé. Différents langages de programmation et outils sont regroupés dans le progiciel PCS7. Puisque le but est le diagnostic industriel, nous avons choisis l'*AMDEC* (Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité) pour extraire les modes de défaillances les plus critiques.

4.2 Automate Programmable Industriel (API)

L'automate programmable ou système d'automatisation est un appareil qui commande un processus. Par exemple une machine à imprimer pour l'impression de journaux, une installation de remplissage de ciment, une presse pour le moulage de formes plastiques sous pression, etc. Ceci est possible grâce aux instructions d'un programme stocké dans la mémoire de l'appareil.

4.2.1 Définition

L'automate programmable est la pièce maîtresse du processus d'automatisation. Il permet de résoudre à moindre coût les tâches d'automatisation les plus diverses. L'automate a pour fonction de commander les opérations d'une machine ou d'une installation en fonction d'une séquence fonctionnelle prédéfinie dépendant des signaux de capteurs [32].

4.2.2 Constitution de l'automate

Les automates programmables sont des appareils fabriqués en série conçus indépendamment d'une tâche précise. Tous les éléments logiques, fonctions de mémoire, temporisations, compteurs etc. nécessaires à l'automatisation sont prévus par le fabricant et sont intégrés à l'automate durant la marche par programmation.

Ils se distinguent principalement par le nombre :

- *entrées et sorties,*
- *emplacements de mémoire,*
- *compteurs,*
- *temporisations,*
- *fonctions spéciales,*
- *la vitesse de travail,*
- *et le type du traitement du programme.*

4.2.3 Automate Programmable Industriel S7-400

L'automate S7 est constitué d'une alimentation, d'une CPU et d'un module d'entrées ou de sorties. A ceux-ci peuvent s'ajouter des processeurs de communication et des modules de fonction qui se chargeront de fonctions spéciales, telles que la commande d'un moteur.

L'automate programmable contrôle et commande une machine ou un processus à l'aide du programme S7. Les modules d'entrées/sorties sont adressés dans le programme S7 via les adresses d'entrée et adresses de sortie.

Dans notre travail, on a utilisé l'automate *Siemens S7-400* (figure 4.1). Le S7-400 est une plateforme d'automatisation conçue à la fois pour l'industrie manufacturière et le génie des procédés. Il se distingue grâce à ses hautes performances, sa puissance de communication et ses grandes capacités de mémoire. Il peut s'adapter à toutes les applications spécifiques telles que la disponibilité élevée, et ou la sécurité [31].



Figure 4.1: Automate S7-400

4.2.4 Caractéristiques

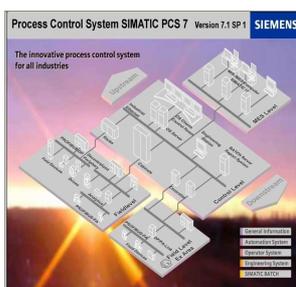
- Le montage de jusqu'à 4 CPU de 144Ko à 20Mo dans un seul châssis de base pour augmenter les puissances de commande, de calcul ou de communication.
- Un concept innovant de mémoire avec accès simultané aux données et au programme qui génère une hausse très notable des performances de traitement.
- La modification de la configuration matérielle en RUN permettant de procéder à des modifications de l'installation sans arrêt de l'exploitation.
- Des variantes à haute disponibilité et de sécurité.
- De puissantes fonctions intégrées de diagnostic du système ainsi que des fonctions configurables de diagnostic de processus.

Le CPU choisie pour notre automate, est le CPU 416-3 DP. Ce type de CPU a plusieurs avantages par rapport aux autres CPU.

4.3 Progiciel PCS7

4.3.1 Description du projet PCS7

4.3.1.1 Qu'est ce qu'un projet PCS 7 ?



PCS 7 est un système de conduite de process de SIEMENS qui, grâce à de nombreuses fonctions automatiques.

Un projet PCS 7 est constitué des objets suivants :

- *Configuration matérielle*
- *Blocs*
- *Diagrammes CFC et SFC*

Il s'agit des objets de base présents quel que soit le nombre de stations opérateur, de modules ou de leur mise en réseau.

4.3.1.2 Applications PCS 7

Le projet sera créé sur une station d'ingénierie (Engineering Station). L'ES possède différentes applications. Toutes ces applications disposent d'une interface graphique pour une utilisation facile et une représentation claire des données de configuration.

Plusieurs applications sont regroupées dans PCS7:

- *SIMATIC Manager*, l'application centrale et le portail d'accès à toutes les autres applications utilisées pour créer un projet PCS 7.
- *HW Config*, Configuration de l'ensemble du matériel d'une installation.
- *Editeur CFC et SFC*, Création de diagrammes CFC et de commandes séquentielles.
- *PCS 7 OS* avec divers éditeurs, Exécution de la configuration OS.

4.3.2 SIMATIC Manager

4.3.2.1 Qu'est-ce que SIMATIC Manager ?

- SIMATIC Manager est l'application centrale, pour ainsi dire le "centre" de PCS 7. A partir de là, vous allez ouvrir toutes les autres applications dans lesquelles vous devez effectuer des paramétrages pour le projet PCS 7.
- SIMATIC Manager et toutes les autres applications sont "interconnectées".
- SIMATIC Manager permet l'accès facile à toutes les données créées dans le SIMATIC Manager et les applications associées.
- En raison de la fonction centrale représentée par SIMATIC Manager dans PCS 7, il est recommandé de familiariser dans tous les cas avec son organisation et sa structure.

4.3.2.2 Systèmes d'automatisation

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

- Le SIMATIC S7-200 est un micro-automate compact d'entrée de gamme. Il possède un progiciel qui lui est propre et n'a pu être pris en compte dans la suite de ce manuel, les propriétés du système S7-200 ne supportant pas une conversion logicielle des programmes S5 en programmes S7.
- Le mini-automate modulaire SIMATIC S7-300 fait encore partie de l'entrée de gamme.
- Le SIMATIC S7-400 couvre le milieu, le haut et le très haut de gamme.

4.3.3 Programmer avec STEP 7

4.3.3.1 Qu'est-ce que STEP 7 ?

Ce logiciel est destiné à la programmation des automates Siemens. Le logiciel STEP 7 permet de concevoir, configurer, programmer, tester, mettre en service et maintenir les systèmes d'automatisation SIMATIC.

STEP 7 intègre en particulier les outils suivants :

- Tous les langages de programmation pour automates programmables définis dans le standard CEI 61131-3: schémas contact, logigrammes, listes d'instructions, graphes séquentiels (S7-GRAPH) et langages structurés (S7-SCL).

- Le logiciel de simulation automate S7-PLCSIM pour la mise au point de programmes sans disposer des automates cible
- Outil de configuration graphique des composants matériels et des réseaux de Communication.

4.3.3.2 Applications disponibles

Le logiciel de base STEP7 met à la disposition des utilisateurs différentes applications (voir figure 4.2) :

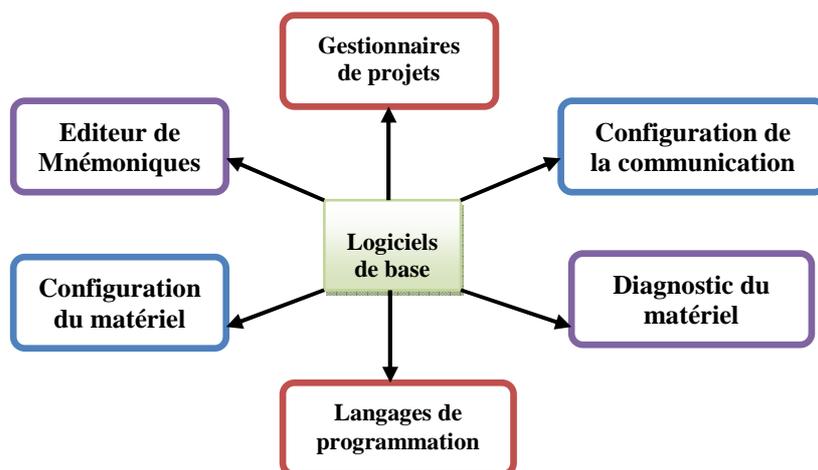


Figure 4.2: Applications disponibles du STEP7

4.3.3.3 Types de blocs

Pour créer des programmes S7, trois langages de programmation CONT, LIST ou LOG sont utilisés [31]. Ces langages disposent d'un éditeur de programme et utilisent plusieurs types de bloc comme l'indique le tableau 4.1 suivant:

Bloc	Désignation	Rôle
OB	Bloc d'Organisation	constitue l'interface entre le système d'exploitation de la CPU S7 et le programme utilisateur. C'est ici qu'est défini l'ordre d'exécution des blocs du programme utilisateur
FB	Bloc Fonctionnel	Est un bloc de code à données statiques. il dispose d'une mémoire, il est possible d'accéder à ses paramètres depuis n'importe quelle position du programme utilisateur
FC	Fonction	Les sont des blocs de code sans mémoire. De ce fait, il faut que les valeurs calculées soient traitées aussitôt après l'appel de la fonction
DB	Bloc de Données	Est une zone servant à mémoriser les données utilisateur. On distingue les DB globaux auxquels tous les blocs de code ont accès et les DB d'instance qui sont associés à un appel de FB déterminé
UDT	Type de Données Utilisateur	Est un type complexe défini par l'utilisateur au besoin et qui est réutilisable. Un type de données utilisateur peut servir à générer plusieurs blocs de données de même structure. Les UDT s'emploient comme des blocs.

Tableau 4.1: Types de blocs utilisés en STEP7

4.3.3.4 Mnémoniques STEP7

Les mnémoniques sont toutes les entrées, sorties et tous les marqueurs pris dans la liste des mnémoniques, ainsi que tous les blocs de données globaux des programmes S7 associés.

L'accès direct aux mnémoniques *STEP7* se fait :

- par le dialogue de sélection des variables,
- par la barre des variables du Graphics Designer.

4.3.4 S7-PLCSIM

- Dans *S7-PLCSIM*, le programme utilisateur *STEP 7* est exécuté et sera essayé dans un automate programmable simulé. La simulation étant réalisée entièrement dans le logiciel *STEP7*, l'utilisateur n'a pas besoin de matériel *S7*.
- Avec *S7-PLCSIM*, la simulation des programmes utilisateur *STEP7* qui ont été développés pour les automates *S7-300* et *S7-400*.
- *S7-PLCSIM* offre une interface simple au programme utilisateur *STEP7* servant à visualiser et à modifier différents objets tels que les variables d'entrée et de sortie.
- *S7-PLCSIM* offre une interface utilisateur graphique permettant de visualiser et de modifier les variables de ces programmes, d'exécuter en mode Cycle unique ou Cycle continu le programme du système cible simulé ou de modifier l'état de fonctionnement de l'automate simulé.
- *S7-PLCSIM* comprend également un objet COM appelé *S7ProSim* pour accéder par programme à un système cible simulé [30].

4.3.5 WinCC (Windows Control Center)



C'est un logiciel de supervision et de configuration en fonctionnement monoposte ou multiposte. Les fonctions du système de contrôle-commande sont réparties selon leurs tâches [32].

WinCC est un système graphique pour afficher et commander les vues de processus. Il représente les informations de processus sous forme de vues, de courbes ou de vues d'alarmes :

- *Système de courbes* pour analyser le processus : les valeurs de processus acquises et mémorisées durant la conduite du processus sont représentées sous forme de courbes et servent à analyser le processus. Ces courbes s'affichent dans les vues de processus, les fenêtres de processus ou les blocs d'affichage.
- *Système d'alarmes* pour diagnostiquer le processus. Les alarmes provenant du système d'automatisation s'affichent par ordre chronologique dans plusieurs listes d'alarmes. Les alarmes s'affichent dans des listes d'alarmes ou sont acquittées par l'opérateur quand il possède l'autorisation appropriée.
- *Système de journaux* pour documenter le processus. Dans la conduite du processus, le système de journaux sert à consigner des commandes et des alarmes ainsi qu'à acquérir les journaux configurés, les éditer et les imprimer.
- *Système d'archives* pour enregistrer et afficher les valeurs de processus, les alarmes et les journaux. Dans la conduite du processus, le système d'archives sert à enregistrer et à afficher des valeurs de processus, des alarmes et des journaux.

Dans le cas réel, où le programme est chargé dans le CPU de l'automate, les valeurs affichées sont des valeurs réelles, soit des valeurs de capteurs, de régulateurs, ou des calculs différents. Dans le cas inverse, elle sera une simulation du processus, soit pour les tests, soit pour les exposés.

4.4 Add-Ons PCS7

4.4.1 FuzzyControl++

4.4.1.1 Qu'est ce que Fuzzy Control++?



FuzzyControl++ est l'outil Siemens dédié à la configuration de tâches telles que la régulation non linéaire, des fonctions de prédiction de processus complexes et plus généralement les tâches qui ne peuvent être résolues avec des outils de programmation standard.

4.4.1.2 Caractéristiques Fuzzy Control++

- La configuration des systèmes de la logique floue sur PC sous Windows 2000 ou Windows XP est assortie d'une aide en ligne très complète.
- Les fonctions d'affectation sont définies très simplement à l'aide de l'outil FuzzyControl++.
- Les règles peuvent être saisies avec un éditeur tabulaire ou matriciel.
- Les modifications de règles sont reconnues instantanément, et dans l'hypothèse où aucune règle ne convient, il est possible d'affecter une valeur par défaut qui sera prise en compte dans ce cas.
- FuzzyControl++ utilise, comme méthode d'inférence, la méthode éprouvée Takagi-Sugeno.
- L'analyse en ligne des systèmes de logique floue est réalisée à l'aide d'outils de courbes et des graphiques à 2, 3 ou 4 dimensions.
- Un générateur de courbes est également disponible pour la simulation de process en mode hors ligne.

La *figure 4.3* montre l'interface de création d'un système d'inférence flou.

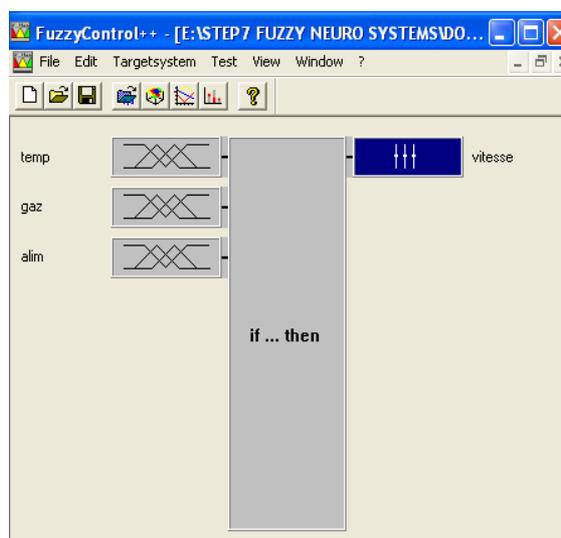


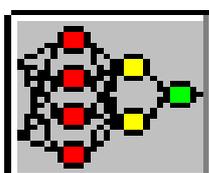
Figure 4.3 : Fuzzy Control++

4.4.1.3 Utilisation Fuzzy Control++

- L'outil de configuration FuzzyControl++, pour automatisation de procédés technologiques, permet de développer d'une manière efficace les systèmes de la logique floue.
- Il est ainsi possible de transformer l'expérience formulée verbalement en modèles ou description logique dans des contrôleurs et des régulateurs.
- FuzzyControl++ est exécutable sur tout système SIMATIC S7, système de contrôle de procédés SIMATIC PCS7 et les systèmes SCADA SIMATIC WinCC.
- Fuzzy control++ met à disposition une bibliothèque de blocs fonctionnels et de vues graphiques.

4.4.2 NeuroSystems

4.4.2.1 Qu'est ce que NeuroSystems ?



NeuroSystems soutient non seulement l'apprentissage, mais également la configuration et l'analyse des réseaux neuronaux. Avec *NeuroSystems*, on peut créer les réseaux neuronaux rapidement et efficacement.

4.4.2.2 Caractéristiques NeuroSystems

- La configuration des systèmes de *NeuroSystems* sur PC sous Windows 2000 ou Windows XP est assortie d'une aide en ligne très complète.
- L'analyse en ligne des systèmes de réseaux de neurones est réalisée à l'aide d'outils de courbes et des graphiques à 2, 3 ou 4 dimensions.
- Un générateur de courbes est également disponible pour la simulation de process en mode hors ligne.
- Le procédé d'apprentissage éprouvé avec des fonctions de paramétrage automatiques.
- Optimisation facile des perceptrons à plusieurs niveaux, des réseaux de base radiaux et des réseaux mixtes neuro-Fuzzy.

La *figure 4.4* montre l'interface de création d'un système neuronal.

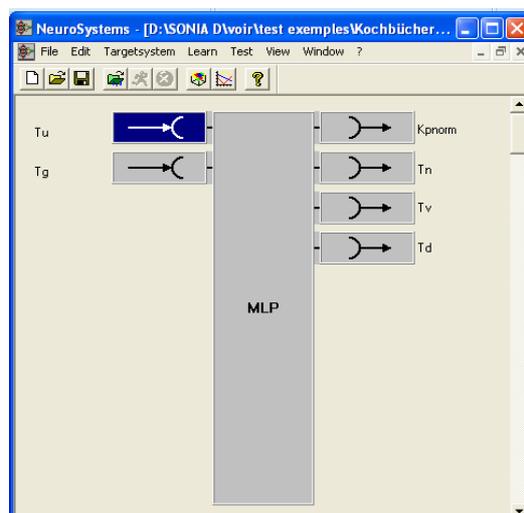


Figure 4.4: Neuro Systems

4.4.2.3 Utilisation NeuroSystems

- L'outil de configuration NeuroSystems a été conçu pour la conception de réseaux neuronaux sur les systèmes d'automatisation SIMATIC S7 et le système de supervision SIMATIC WinCC.
- NeuroSystems sont employés dans les domaines des processus de technologie de commande, d'identification de modèle, de prévision, de classification et d'optimisation.
- NeuroSystems donne ainsi de bons résultats pour des applications de reconnaissance de formes, de modélisation, d'optimisation ou de régulation.
- NeuroSystems dispose d'une bibliothèque composante pour SIMATIC S7.

4.4.3 Fuzzy-NeuroSystems

Lors de la mise en œuvre de réseaux mixtes Neuro-Fuzzy, il est possible d'optimiser les systèmes à base de logique floue à l'aide de l'outil NeuroSystems pour ce qui concerne le rattachement et la forme des fonctions d'appartenance.

NeuroSystems vous permet un échange de données de système-croisement par l'intermédiaire des dossiers de *FPL* (Fuzzy Programmation Language) et de *SNL* (Siemens Neuro Language). Cette fonction permet une mise au point aisée de systèmes Fuzzy et Neuro-Fuzzy (voir *figure 4.3*).

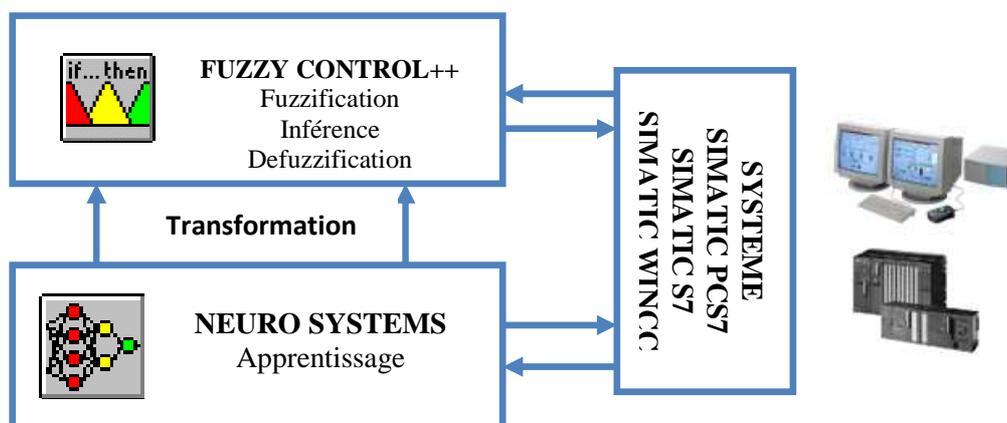


Figure 4.5: Neuro-Fuzzy Systems

4.4.4 GRAFCET

La complexité croissante des systèmes industriels automatisés impose une définition précise des caractéristiques fonctionnelles qui en régissent les cycles d'exploitation et ce, indépendamment, de leur matérialisation technologique : électrique, mécanique, pneumatique et électronique.

Et de la solution retenue pour la partie commande, l'*AF CET* (Association Française pour la Cybernétique Economique et Technique) et l'*ANPA* (Agence Nationale pour la Production Automatisée) ont mis au point et développé une représentation graphique qui traduit, sans ambiguïté, l'évolution des cycles d'automatisme séquentiel.

Ce diagramme fonctionnel : le **GRAF CET** (**GRA**phe **F**onctionnel de **C**ommande **E**tape **T**ransition) permet de décrire les comportements attendus de l'automatisme de commande au niveau :

- ✚ du traitement des informations délivrées par la partie opérative,
- ✚ et des ordres transmis à cette même partie opérative.

Le GRAFCET est un outil méthode, descriptif du cahier des charges de tout système séquentiel.

4.5 AMDEC

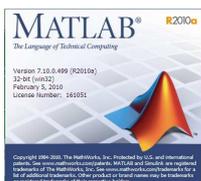
Nous avons choisi la méthode AMDEC pour représenter toutes les informations sur le système à étudier. AMDEC est une technique déductive et qualitative avec laquelle les effets (conséquences) des défaillances des composants élémentaire sont systématiquement identifiés.

L'utilisation des tableaux d'AMDEC à des fins de diagnostic et pronostic industriel conduit à utiliser une procédure déductive, c'est-à-dire à utiliser ces tableaux comme outil d'identification des causes de défaillances à partir des effets observés.

La démarche consiste d'abord à définir le système, ses fonctions et ses composants. Ensuite, l'ensemble des modes à défaillances des composants doit être établi. Pour chaque mode de défaillance, sont recherchées ensuite les causes possibles de son apparition. Finalement, une étude des effets sur le système est faite pour chaque combinaison (cause, mode de défaillance).

4.6 MATLAB

4.5.1 Qu'est ce que NeuroSystems ?



MATLAB est une abréviation de *MAT*rix *LAB*oratory, était destiné à faciliter l'accès au logiciel matriciel. *MATLAB* est un langage de calcul scientifique de haut niveau et un environnement interactif pour le développement d'algorithmes, la visualisation et l'analyse de données, ou encore le calcul numérique.

4.6.2 Caractéristiques Matlab

- Résoudre des problèmes de calcul scientifique plus rapidement.
- *MATLAB* offre un certain nombre de fonctionnalités pour la documentation et le partage de votre travail.
- Des boîtes à outils supplémentaires (collections de fonctions *MATLAB* à vocation spécifique, disponibles séparément) élargissent l'environnement *MATLAB* pour résoudre des catégories particulières de problèmes dans ces domaines d'applications : *SIMULINK*, réseaux de neurones et la logique floue.

4.6.3 Utilisation Matlab

MATLAB utilisé dans une grande variété d'applications :

- Le traitement du signal et d'images.
- Les communications.

- La conception de systèmes de contrôle.
- Les tests et les mesures.
- La modélisation et l'analyse financière.

4.6.4 Principales fonctionnalités

- Langage de haut niveau pour le calcul scientifique.
- Environnement de développement pour la gestion du code, des fichiers et des données.
- Outils interactifs pour l'exploration itérative, la conception et la résolution de problèmes.
- Fonctions mathématiques pour l'algèbre linéaire, les statistiques, l'analyse de Fourier, le filtrage, l'optimisation et l'intégration numérique.
- Fonctions graphiques 2-D et 3-D pour la visualisation des données.
- Outils pour la construction d'interfaces graphiques personnalisées.
- Fonctions pour l'intégration d'algorithmes développés en langage MATLAB, dans des applications et langages externes (C/C++, Fortran, Java, COM et Microsoft Excel).

4.5.5 Modules complémentaires à Matlab

- ✚ **Neural Network Toolbox** : Réseaux associatifs, à rétropropagation, de Hopfield, de Kohonen, à auto-organisation, à base radiale et autres réseaux - Fonctions compétitives, limites, linéaires, de transfert sigmoïde - Architectures récurrentes, rétroactives - Fonctions d'analyse des performances et graphes - Taille de réseaux illimitée - Blocs 'réseaux de neurones' sous SIMUUNK.
- ✚ La **Fuzzy Logic Toolbox** de Matlab permet de présenter des informations de haut niveau concernant le système d'inférence flou (*Fuzzy Inference System*, ou *FIS* en abrégé). À partir de l'éditeur correspondant, il est possible de choisir le type de régulateur (Mamdani, Sugeno), de définir, visualiser et éditer les différentes fonctions d'appartenance, construire la base de règles, choisir les méthodes d'implication.
- ✚ **ANFIS** : Adaptive Neural Fuzzy Inference Systems (Système d'inférence floue paramétré par apprentissage neuronal).

4.7 Conclusion

Après la *phase d'analyse* qui consiste à étudier l'existant en recensant toutes les informations nécessaires du champ d'étude, notamment en termes d'expression des besoins de l'utilisateur par rapport au système projeté, et la définition de l'environnement de travail que ce soit hardware et software, maintenant une *phase de conception* nécessaire pour l'élaboration du nouveau système.

CHAPITRE 05

VERS UNE APPROCHE NEURO-FLOUE POUR LE PRONOSTIC INDUSTRIEL

La *Phase de conception* à commencer par la modélisation du système selon les données du cahier de charge, il en a découlé la réalisation de la base de données, et le choix de l'architecture à utiliser. L'application de la méthode d'analyse des dysfonctionnements et la classification des différents paramètres à étudier est le contexte de ce chapitre. Ainsi que les différents algorithmes nécessaires pour réaliser le nouveau système.

5.1 Introduction

L'un des problèmes d'une grande importance dans la surveillance industrielle est constitué par la possibilité d'effectuer une détection des alarmes et éventuellement de prédire une défaillance. Pour cette raison, notre étude s'est focalisée au pronostic industriel.

Notre étude se propose de développer un système d'aide au pronostic basé sur une technique située à l'intersection d'une technique neuronale et de la logique floue, qui est appliqué au procédé de clinkérisation de la SCIMAT- Batna.

5.2 Atelier de Clinkérisation

L'atelier de clinkérisation regroupe deux lignes de fours rotatifs dont le débit de clinker est 1560 t/h. Dans cette partie, on va présenter une conception brève du four, ensuite un aperçu sur les différentes zones de four, et enfin décrire la problématique de la conduite du four.

5.2.1 Conception du four rotatif

Le four, de la SCIMAT (*figure 5.1*), est un cylindre d'acier rotatif posé avec une inclinaison de 3 à 5% vers la sortie. Il est calé à intervalles par des trains porteurs à galets. La rotation lui est transmise à travers une couronne dentée, un pignon et un réducteur de vitesse. Le four aura tendance à se déplacer vers l'aval au cours de sa rotation. Il est protégé contre les très hautes températures de service par revêtement de briques réfractaires maçonnées à l'intérieur sur toute longueur. Le revêtement est sensible à la déformation élastique du four et les variations de température.



Figure 5.1: Four Rotatif de la SCIMAT

L'extrémité la plus en aval du four, la sortie, est fermée par un fond traversé par une tuyère. La température de cuisson est aux environs de 1450°C. La chauffe est faite soit au fuel, soit au gaz, soit au charbon pulvérisé ou une combinaison de ces trois. Les matières premières sont enfournées à l'entrée du four et se déplacent à contre-courant des gaz chauds, à travers le four jusqu'à ce qu'elles en ressortent par le refroidisseur sous forme de clinker cuit.

5.2.2 Zones de réaction du Four Rotatif

Dans tous les systèmes de fours, les matières premières sont soumises au même traitement (figure 5.2) :

- préchauffement,
- calcination,
- cuisson,
- refroidissement.

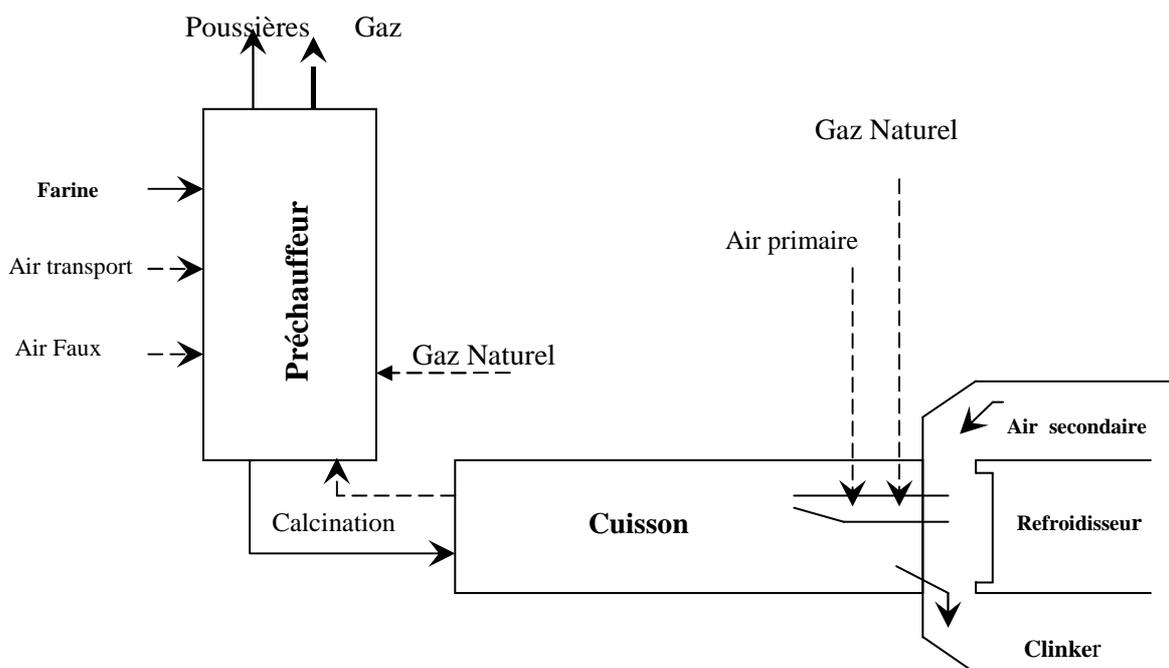


Figure 5.2: Zones de réaction du Four Rotatif

5.2.3 Problématique dans la conduite du four

Le four est la zone la plus sensible aux défaillances à condition de ces différentes fonctions à remplir. Les interactions entre ces diverses fonctions sont extrêmement complexes et difficiles à identifier. Les connexions déterminantes sur le résultat du processus de cuisson sont celles reliant la production de la chaleur et les transformations chimiques de la matière.

Le fonctionnement du four influence plusieurs paramètres comme la les paramètres de la combustion, la qualité du clinker et la consommation de l'énergie. L'environnement du four pose d'énormes difficultés pour disposer de mesures fiables caractérisant son fonctionnement. Le four en tant que système à commander est caractérisé par l'aspect multi-variable.

5.3 Pronostic de l'atelier

Le pronostic est réalisé par une technique neuro-floue temporelle de classification qui utilise les paramètres d'identification dont les variations caractérisent les évolutions du système en temps réel. Les données utilisées pour le réseau sont considérées comme des séries temporelles à surveiller.

5.3.1 Démarche proposée

Afin d'implémenter ce type de perceptron flou temporel et l'exploiter pour diagnostiquer d'une façon dynamique un système de production dédié, nous avons proposé un logiciel informatique *NEFPROG V 1.0* « *NEuro FUZZY PROGNosis* ». Ce dernier est un logiciel informatique de simulation interactive, développé au sein de LAP-Université de Batna. Ce logiciel est consacré essentiellement au développement, à l'apprentissage et au test d'un système neuro-flou de classification des défaillances d'un procédé industriel.

5.3.2 Pronostic industriel basé sur la *RdF Neuro-floue*

La méthode de pronostic industriel est une meilleure exploitation des caractéristiques des systèmes neuro-flous, qui s'appuie sur une représentation numérique et en même temps symbolique des formes.

Le fonctionnement d'un système de diagnostic par *RdF* se déroule en trois phases :

- une **phase d'analyse**
- une **phase de choix**
- une **phase d'exploitation**

Le diagnostic de défaillances dans ce cadre est essentiellement vu comme un problème de *classification*. Le but principal est de construire un bloc de correspondance tel qu'à partir d'un ensemble d'informations décrivant la situation courante de processus, il est possible de prédire d'une défaillance et d'obtenir les causes probables de situations anormales. Quand le diagnostic est basé sur des observations multiples, elles sont regroupées pour former des classes qui définissent un mode de fonctionnement du processus, auquel une nouvelle observation sera comparée pour être identifiée.

La *figure 5.3* représente le pronostic industriel par base de la *RdF Neuro-floue*.

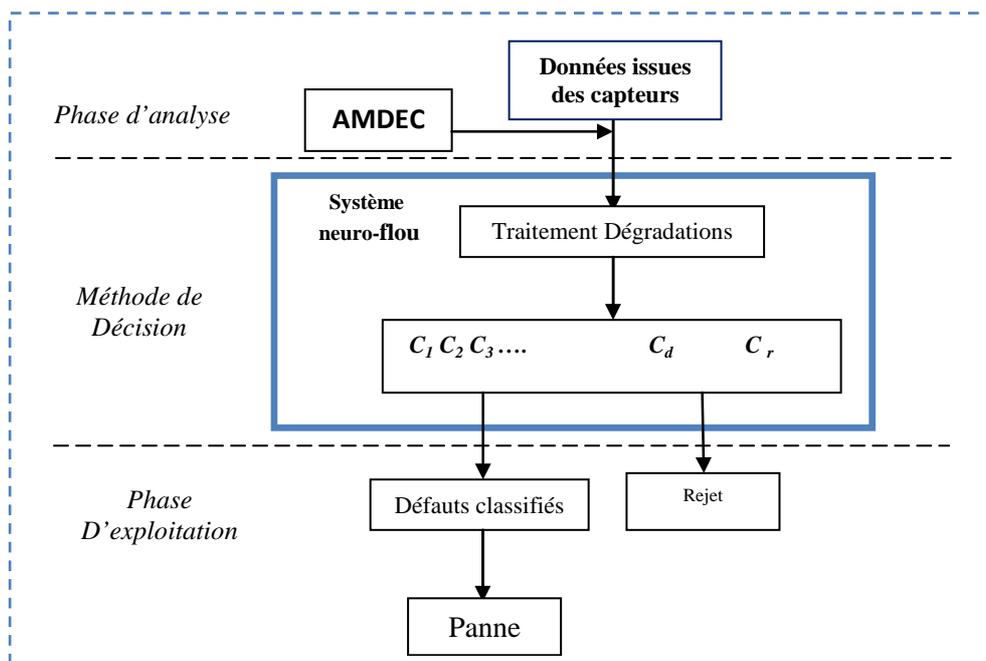


Figure 5.3: Pronostic industriel par *RdF Neuro-Floue*

5.3.3 Phase d'analyse

Une **phase d'analyse** où il s'agit d'étudier les informations fournies par les différents capteurs, les historiques ainsi que l'expérience des opérateurs, afin de définir l'espace de représentation du système ainsi que les classes qui représenteront les différents états de fonctionnement.

5.3.3.1 AMDEC

L'étape d'analyse a pour objectif l'identification des dysfonctionnements pouvant affecter la mission du système. Cette analyse est grandement facilitée par la connaissance des modèles structurels et fonctionnels de l'installation.

L'AMDEC permet une analyse systématique et très complète, composant par composant, de tous les modes de défaillance possibles et précise leurs effets sur le système global. Cette étape a pour objectif l'identification des dysfonctionnements pouvant affecter la mission du système. Cette analyse est grandement facilitée par la reconnaissance des modèles structurels et fonctionnels de l'installation.

L'AMDEC de l'atelier de clinkérisation est illustrée au *tableau 5.1* suivant :

Identifiant composant	Fonction états	Mode Défaillance	Causes possibles	Effets	Moyen de décision	Parades de l'opérateur humain	Criticité
Trémie d'alimentation four (Schenck)	Alimentation Four (farine crue)	Variation débit	Mauvais pesage	Bourrage cyclone	CCS perturbée	Contrôle Pesage	2
			Débit mètre erroné	Bourrage cyclone	Température cyclone augmentée	correction d'arrivants dans la trémie	
			Forme de croustage (croustage épais ou décroustage)	Qualité Clinker	Cem-Scanner, Etranglement	Vérification locale (nettoyage, diminution MP)	
		Mauvaise qualité Farine crue	Tas non homogène	Qualité Clinker	Variation paramètres (LSF, MS, MA),	Correction Labo	2
			Mal hétérogénéisation farine		Disponibilité air actif et inactif	Maintenance (suppresseur, compresseur)	
Analyseur	Analyse MP	Décarbonatation perturbée	Etranglement (blocage)	Bourrage	Pression, température	Correction chimique	1
			Manque jupe		Augmentation CO2, diminution O2	Mettre en équilibre (Gaz, Oxygène)	
		Choke thermique (Changement chimique brusque de la MP)	Décroustage du four répétitif	Taches rouges	Moment augmente	Diminution d'alimentation	2
			usure brique prématurée		Mauvaise qualité clinker	Stabilité de croustage	
Cyclones	Préchauffement	bourrage	blocage	Qualité Clinker	Diminution pression	Débouillage (arrêt du four)	3
		Mauvais changement chaleur	Air faux	Consommation gaz	Mauvais qualité clinker	Elimination fuite	2
			Etranglement		Mauvais qualité clinker	nettoyage cyclone	

Identifica tion composant	Fonctio n états	Mode Défaillance	Causes possibles	Effets	Moyen de décision	Parades de l'opérateur humain	criti cité
FOUR (calcinati on, cuisson, refroidiss ement)	Clinkéri sation (Air, Gaz, MP)	Variation débit gaz (insuffisance ou absence)	Basse pression gaz	Blocage	Perturbation des températures	Arrêt du four	3
			Membrane vanne de détention défailante	Qualité Clinker	Débit de gaz insuffisant		
		Instabilité combustion	Ventilateur Final	Qualité KK	Arrêt gaz		3
			Ventilateur primaire		Arrêt bruleur		
	Tirage	Augmentation CH4 et diminution O2					
	Mauvaise qualité Clinker	Perturbation chimique farine	Consommatio n gaz	Perturbation : poids litre diminue, chaux libre augmente et Consommation calorifique spécifique augmente	Correction Gaz/alimentation	3	
	Refroidi ssement rapide du clinker	Mauvais refroidissement	Mauvais tirage	Qualité KK	Bas lourd, accouplement défectueux	Diminution vitesse four (élimination vibration), augmenter tirage	1
Cisaillement des releveurs			Air secondaire diminue Température clinker augmente		arrêt four (changer équipement)		
Silos stockage	Transpo rt du Clinker	Mauvais transport	Transporteur à auget défaillant	Rejet Clinker	blocage	Arrêt du four, maintenance équipement	1

Tableau 5.1: L'AMDEC du procédé du clinkérisation

5.3.3.2 Contexte de l'étude

La stabilité dans le fonctionnement du four est l'objectif principal qui guide le travail de l'opérateur. La stabilité recherchée consiste à maintenir les variables critiques du processus dans certaines bandes qui assurent la continuité de son fonctionnement, et en conséquence, la continuité de la production.

L'idéal est de conduire le four à un niveau thermique inférieur et maintenir le reste de ses variables toujours autour de leurs consignes respectives (stabilité totale du reste de ses variables) à condition, bien sûr que le niveau thermique minimum soit compatible avec une production de clinker de qualité. Le système de conduite peut recevoir des signaux mesurés des variables du processus.

Les variables de notre système utilisées sont représentées sur le *Tableau 5.2* suivant :

<i>Les entrées de RNF (observations)</i>		<i>La couche des règles</i>	<i>Les sorties de RNF (modes de défaillance)</i>
<i>N°</i>	<i>Désignation</i>		
1	Vitesse de rotation four	Vide ou Règles (connaissance A priori)	<ol style="list-style-type: none"> 1. <i>Variation Débit Gaz</i> 2. <i>Instabilité Combustion</i> 3. <i>Mauvaise Qualité Clinker</i> 4. <i>Mauvais refroidissement</i>
2	Puissance moteur principal		
3	Température cuisson		
4	Moment four		
5	Température air secondaire		
6	TIRAGE		
7	Température alimentation		
8	four		
9	Débit alimentation four		
10	Température Clinker		
11	% CO		
12	% O2		
13	% CH4		
14	% NOX		
15	Poids au litre		
16	Température virole		
17	Température après cyclone		
18	Débit combustible		
	Vitesse ventilateur tirage		

Tableau 5.2: Les entrées et les sorties du système

5.3.3.3 Architecture du système Neuro-Flou

L'architecture proposée du système neuro-flou temporel est un perceptron multicouches avec une duplication de la couche cachée sur une couche appelée couche de contexte, identique à l'architecture proposée par Elman.

Mais, les poids sont modélisés par des ensembles flous. Les activations, les sorties et les fonctions de propagation seront changées.

L'Architecture du système neuro-floue temporel est illustrée sur la figure 5.4. Les règles créées avec le système sont des connaissances à priori. Chaque variable à une partition initiale est modifiée au long de la phase d'apprentissage (nombre de ensembles flous pour chaque variable).

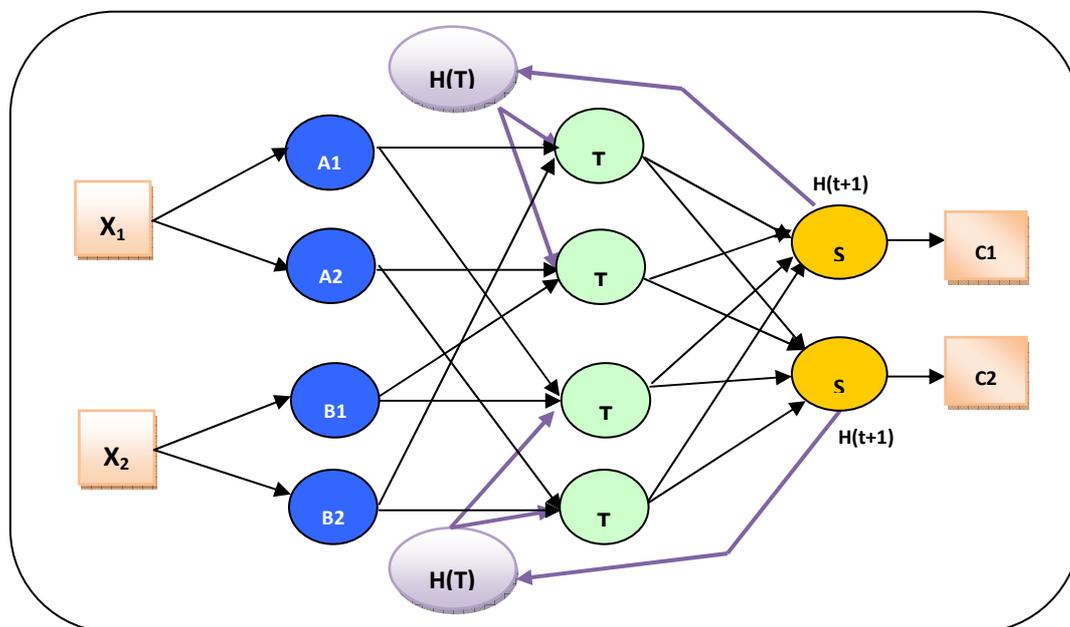


Figure 5.4: Architecture Neuro-Floue d'Elman

Dans ce réseau, les nœuds qui sont en entrée et les nœuds de la couche contexte qui copie la valeur des nœuds de la quatrième couche sont tous connectés à la troisième couche.

Ils représentent les variables linguistiques dans les règles floues de la troisième couche.

- Le système neuro-floue a été choisi comme une méthode de décision.
- L'apprentissage des systèmes de neuro-flous artificiels est une phase qui permet de déterminer ou de modifier les paramètres du système, afin d'adopter un comportement désiré.

5.3.4 Phase de choix

Une *phase de choix* du système de défaillance qui consiste à construire une règle de décision qui établira des frontières entre les différentes classes. La règle de décision permettra d'affecter ou non une nouvelle observation à l'une des classes connues.

Dans cette partie, on utilise FuzzyControl++ pour illustrer les différentes classes de défaillances ainsi le vecteur d'entrée de notre système (figure 5.5).

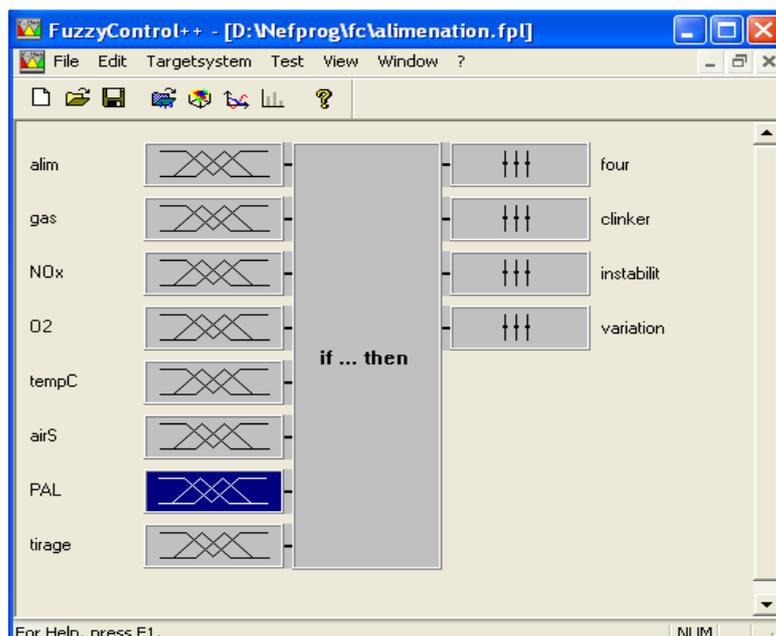


Figure 5.5: Système Flou

Puisque FuzzyControl++ limite le nombre d'entrée en huit et le nombre de sorties en quatre, voici les exemples (figure 5.6 et 5.7) extraites de notre système. Les fonctions d'appartenance de chaque paramètre sont données ci-dessous :

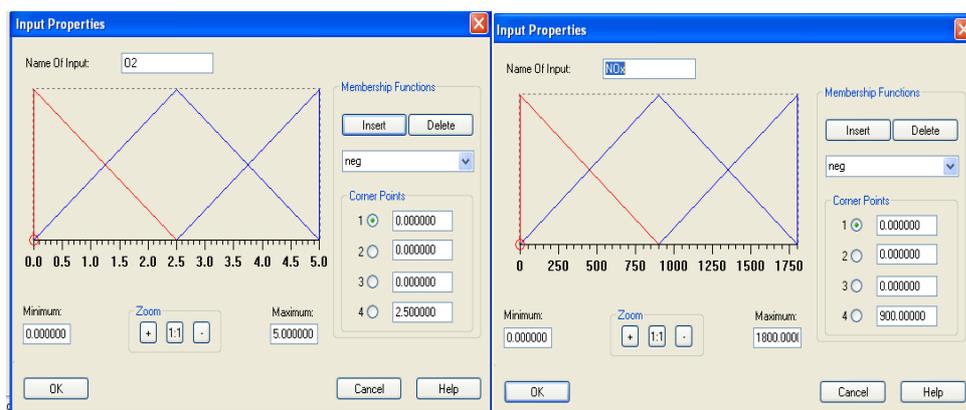


Figure 5.6: Fonctions d'appartenance de %O₂ et NO_x

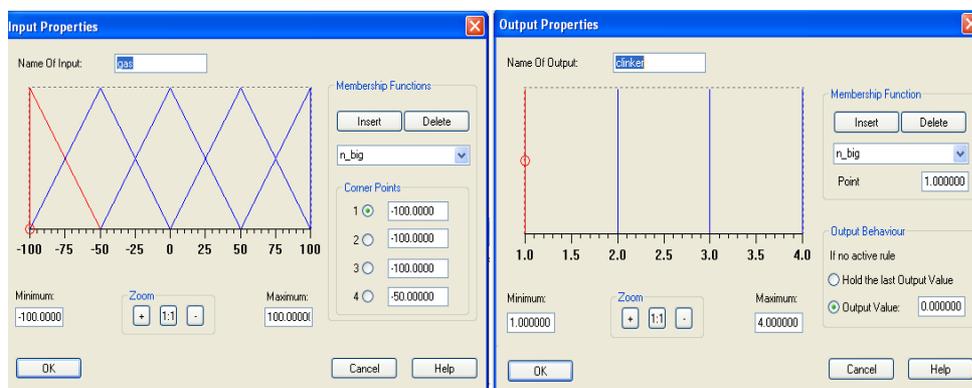


Figure 5.7: Fonctions d'appartenance de clinker et gaz

5.3.5 Phase d'exploitation

Une *phase d'exploitation* qui consiste à implémenter le système afin de proposer une décision pour toute nouvelle observation effectuée. La bonne exploitation du système dépend de la pertinence de l'espace de représentation et des performances de la règle de décision.

L'étape d'apprentissage est basée sur la descente de gradient d'erreur quadratique moyenne commise par le RNF. Après avoir calculé l'activation en avant des neurones des différentes couches, l'erreur est ensuite rétro propagée dans le sens inverse de l'activation pour pouvoir calculer, pour chaque neurone, sa contribution (5.1).

$$J_m(U, V) = \sum_{k=1}^N \sum_{i=1}^m (u_{ik})^p (d_{ik})^2 \quad (5.1)$$

Des corrections sont ensuite apportées aux différents paramètres du système (les poids, les seuils et les paramètres des fonctions d'appartenance).

L'utilisateur définit le nombre initial des fonctions d'appartenance pour partitionner les domaines des données d'entrées et spécifie le nombre k , nombre maximum de neurones des règles qui seront créés dans la couche cachée.

Les *principales étapes d'apprentissage* sont ainsi présentées:

- *Initialisation* : pour chaque donnée issue des capteurs, il y a une unité d'entrée, et pour chaque mode de défaillance il y a une unité de sortie. Pour chaque unité d'entrée une partition floue initiale est spécifiée.
- *Apprentissage des règles* : Le système peut démarrer avec une base de connaissance partielle des formes, et raffiner durant l'apprentissage. La règle sera créée par la recherche (pour une forme donnée f) la combinaison des fonctions d'appartenance telle que chaque entrée produit la plus grande fonction d'appartenance. Si cette combinaison n'est pas identique pour les règles existantes dans la base des règles et le nombre de règles n'est pas maximum, alors une règle sera créée et ajoutée à la base des règles.
- *Apprentissage des FA* : pour l'apprentissage des fonctions d'appartenance, une simple rétro propagation sera utilisée. Une décision sera créée qui dépend de l'erreur de sortie pour chaque unité de règles. Chaque règle change ses fonctions d'appartenance par le changement de leurs supports.

a. Apprentissage des règles

D'abord toutes les variables du système à diagnostiquer sont partitionnées par des fonctions d'appartenance avant que l'algorithme d'apprentissage ne démarre.

Si l'utilisateur ou l'expert du système détermine ces fonctions d'appartenance et les nomment lui-même, alors ces noms sont utilisés en tant que « vocabulaire » pour la description du problème qui est représenté par la base de règles.

Les règles sont créées par le déroulement des données d'apprentissage. Au début, la base de règles est vide, ou contient quelques règles « définie par l'utilisateur ou l'expert de système » comme une connaissance à priori.

Le nombre maximum des règles sera défini par l'utilisateur, si notre système a trouvé des règles > k_{max} règles, alors il raffine la base des règles (candidats).

Pour le premier cycle, tous les antécédents sont créés (*Annexe D*), pour chaque point des ensembles de données utilisées pour la création de règles, une combinaison des ensembles flous est sélectionnée.

Pour chaque variable, sa fonction d'appartenance, qui a le maximum degré de ces FA pour la valeur d'entrée courante. Si l'antécédent combiné de ces ensembles flous n'existe pas dans la liste antécédents, additionner-le dans la liste des antécédents. Durant le deuxième cycle, le meilleur conséquent pour chaque antécédent est sélectionné et les règles sont complétées.

L'algorithme d'apprentissage suivant [16] est utilisé pour créer k unités de règles de NEFPROG.

Algorithme01: Apprentissage des règles floues

Début

- (1) sélectionner la forme suivante (f,t) de Γ .
- (2) pour chaque unité d'entrée $x_i \in U_1$ chercher la fonction d'appartenance

$$\mu_{ii}^{(i)}$$

$$\mu_{ji}^{(i)}(p_i) = \max_{j \in \{1, \dots, q_i\}} \{\mu_j^{(i)}(x_i)\}$$

- (3) si le nombre des règles $\leq k_{max}$, et il n'existe pas de règle R avec :

$$W(x_1, r) = \mu_{i1}^{(1)}, \dots, W(x_n, R) = \mu_{in}^{(n)}$$

Alors créer un nœud et connecter la au nœud de sortie CI si $t_i=1$.

- (4) s'il y a des formes non traitées dans Γ et $k \leq k_{max}$ alors aller a (1). Sinon

Fin.

Tableau 5.3: Apprentissage des règles floues

b. apprentissage des fonctions d'appartenance

Pour utiliser la technique de descente de gradient; il faut que la fonction d'appartenance et toutes les fonctions qui seront utilisées pour l'évaluation des règles floues seront différentiables. Aussi nous pouvons utiliser les fonctions d'appartenance triangulaire ou trapézoïdale.

Algorithme02 : Apprentissage des ensembles flous

Début

1- sélectionner la forme suivante (p,t) de Γ , propager la, par NEFDIAG et déterminer le vecteur de sortie C .

2- pour chaque unité de sortie C_i ; déterminer la valeur de delta

$$\delta C_i = t_i - O_{C_i}$$

3- pour chaque unité de règle R avec $O_R > 0$

a- déterminer la valeur de δ

$$\delta_R = O_R(1 - O_R) \sum_{C \in U_3} W(R, C) \delta_C .$$

b- trouver x' tel que :

$$w(x', R)(O_{x'}) = \min_{x \in U_1} \{w(x, R)(\delta_C)\}$$

c -pour l'ensemble flou $w(x', R)$ déterminer les valeurs de δ pour les paramètres a, b, c utilisant le pas d'apprentissage $\sigma > 0$.

$$\delta_b = \sigma \cdot \delta_R \cdot (c-a) \operatorname{sgn}(O_{x'} - b),$$

$$\delta a = -\sigma \delta_R \cdot (c-a) + \delta_b,$$

$$\delta c = \sigma \delta_R \cdot (c-a) + \delta_b.$$

Appliquer le changement aux $w(x', R)$.

4 -si l'itération est terminée, ou les critères de fin sont validé ; alors stop

Sinon aller a (1).

Fin

Tableau 5.4: Apprentissage de fonctions d'appartenance

5.4 Conclusion

Au sein de l'entreprise de fabrication des ciments « SCIMAT », la technique de la logique floue était déjà appliquée sur de cas réel à l'année 1997. Mais malheureusement, le contrôle d'un four présente certains problèmes en raison de la complexité du processus, des temps de réponse ainsi que des possibilistes de mesure limitées (voir annexe D).

Dans les années récentes, un développement des deux techniques : les réseaux de neurones artificiels et la logique floue, qui sont appliqués actuellement séparément ou en combinaison dans le domaine de diagnostic des pannes.

Dans ce travail, en combinant les deux techniques précédentes en utilisant une base de données d'apprentissage régénérées afin d'étudier les modes de défaillances dans l'atelier de clinkérisation, en ajoutant l'aspect temporel. C'est une approche destinée au pronostic industriel, c'est-à-dire prédire des défaillances futures du système.

Donc, une nouvelle approche Neuro-Floue Temporelle est présentée. Afin d'implémenter le progiciel obtenu au chapitre suivant.

CHAPITRE 06

IMPLEMENTATION NEFPROG

Les couts élevés da maintenance des équipements complexes rendent nécessaires d'augmenter l'effort de développement d'outils support au « processus de pronostic ». Dans ce cadre, le travail rapporté dans cette thèse vise la spécification d'un cadre applicatif pour l'implémentation d'un système de pronostic. A ce jour, aucun outil de reconnaissance de forme n'est appliqué à la cimenterie d'Ain-Touta. Mais, à l'année (1987-1997), la logique Floue est intervienne à la conduite automatique de la zone four. Pour cela, un nouveau système est proposé dans le domaine de la RdF, utilisant la logique floue et les RNAs pour le diagnostic prédictif de la zone de cuisson.

6.1 Introduction

L'atelier choisi pour notre étude est la clinkérisation, c'est la partie sensible de défaillances. Après l'alimentation du Four, à partir du Schenck, la farine crue sera préchauffée dans les cyclones dont le but de l'homogénéisation. Ensuite, une étape de cuisson sera produite pour l'obtention du Clinker, afin de refroidir ce dernier dans un refroidisseur à ballonnets.

Dans cette partie, on essaye de présenter l'outil « NEFPROG », ces différents composants de programme de l'automate, les variables et les entrées/sorties du système dans une table mnémorique. Ainsi qu'un tableau qui résume les différentes pré-alarmes et alarmes utilisées. Nous avons, ensuite, présenter l'interface graphique du système et les résultats obtenus par le système neuro flou.

6.2 Composants du programme

Tous les composants du programme sont affichés y compris la station SIMATIC 400 au démarrage du SIMATIC Manager du programme. Une vue globale du système est présentée sur la *figure 6.1*.

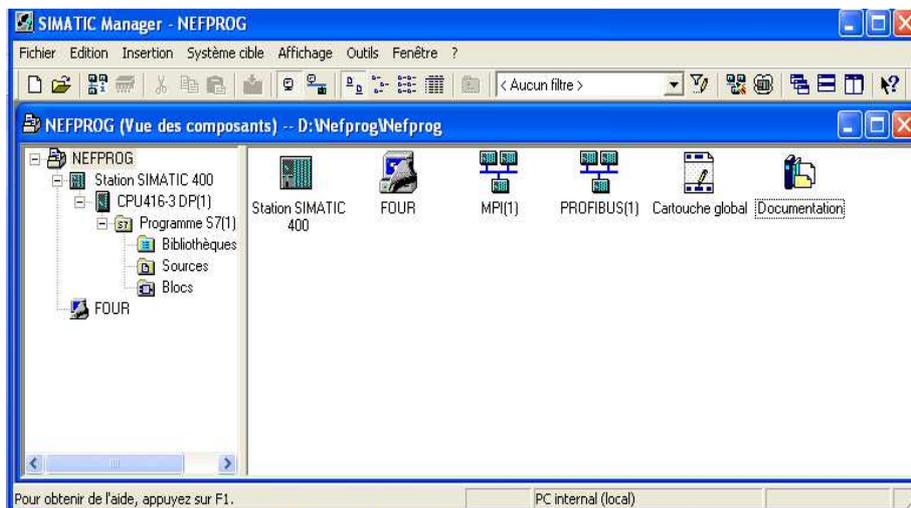


Figure 6.1: Composants du programme

Le programme S7 constitue de :

- Station SIMATIC 400
 - Programme S7 de CPU 416-3 DP, c'est le type de CPU installé aux automates de la Cimenterie de Ain-Touta.
 - Blocs : l'ensemble des blocs programmés pour chaque équipement.
- Table des mnémoniques :
- Station Opérateur (OS) : c'est la partie supervision du process.

6.2.1 Station Opérateur

L'interface homme/machine est possible en créant des vues dans la station opérateur (OS). Ces vues présentent le processus du système, les équipements de l'atelier, les alarmes peuvent avoir lieu, et d'autres informations.

L'interface est exécutée dans WINCC Runtime, où l'opérateur peut entrer des données, voir des indications, démarrer ou arrêter la marche du système.

Les vues utilisées sont illustrées sur la figure 6.2 ci-dessous :

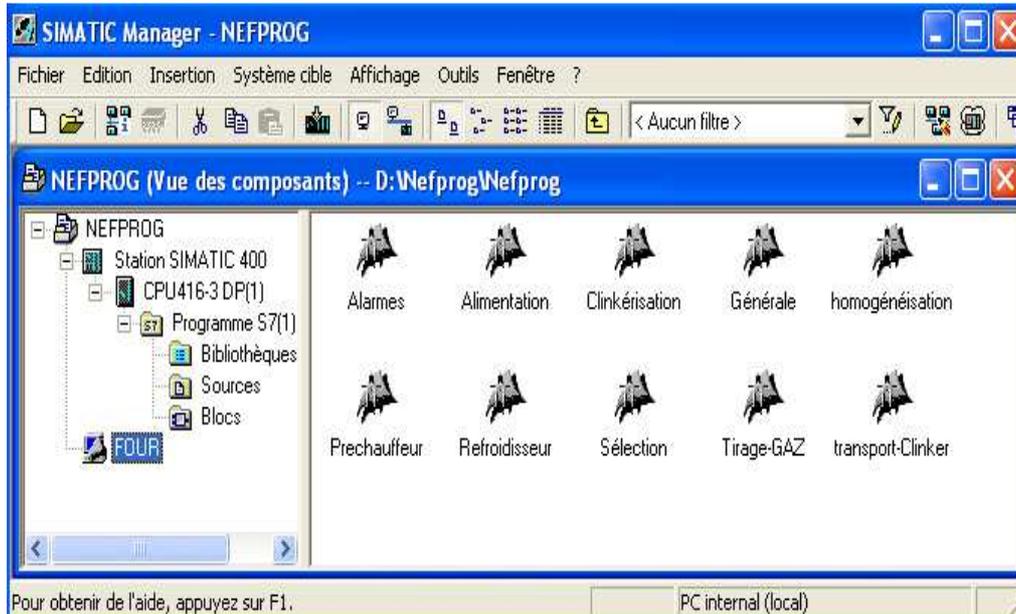


Figure 6.2: Interface OS du système

6.2.2 Table Mnémotique

Les mnémoniques du programme sont l'ensemble de toutes les entrées et les sorties du système. Elles sont affectées aux blocs système, blocs utilisateur, et aussi aux composants, équipements et alarmes. Les mnémoniques de notre système pour les entrées, les sorties, les temporisations, les compteurs, les mémentos et les blocs sont listés dans la table des mnémoniques (tableau 6.1) suivante :

Et	Mnémonique	Opérande	Type de do	Commentaire
208	W1A07U41	M 412.6	BOOL	ALARME.SURPRESSEUR
209	W1A08L9	M 411.1	BOOL	NIVEAU MAX.AEROGLISSIERE
210	W1A13C31	M 413.3	BOOL	COMMANDE OUV.REGISTRE PAPILLON
211	W1A13P1	MD 344	REAL	AIR.COMP.MIN
212	W1A50A01	MD 102	REAL	AI.CO%.APRES CYCLONES
213	W1A50A02	MD 106	REAL	AI.O2%.APRES CYCLONES
214	W1A50A03	MD 110	REAL	AI.CH4%.APRES CYCLONES
215	W1A50A1	M 402.5	BOOL	DI.TEST GAZ ANALYSEUR.APRES CYCLONES
216	W1A50A2	M 402.6	BOOL	DI.DEFAUT HUMIDITE.APRES CYCLONES
217	W1A50P1	MD 98	REAL	PRESSION APRES CYCLONE
218	W1A50T1	MD 114	REAL	AI.TEMPERATURE.APRES CYCLONES
219	W1A54L1	M 402.7	BOOL	DI.NIVEAU MAX.APRES CYCLONES
220	W1A54P2	MD 122	REAL	AI.PRESSION.APRES CYCLONES
221	W1A54T1	MD 118	REAL	AI.TEMP.APRES CYCLONES PREALARME
222	W1A54T1-BOOL	M 300.7	BOOL	AI.TEMP.APRES CYCLONES
223	W1A54T2	MD 126	REAL	AI.TEMP.APRES CYCLONES ALARME
224	W1A54T2-BOOL	M 308.7	BOOL	AI.TEMP.APRES CYCLONES
225	W1K01P1	MD 206	REAL	AI.PRESSION.AIR SECONDAIRE
226	W1K01T1	MD 210	REAL	AI.TEMPERATURE.AIR SECONDAIRE
227	W1P11C31	M 412.1	BOOL	PERM.LOCALE.CHAUFFAGE FILTRE
228	W1P11C32	M 412.4	BOOL	COMMANDE .CHAUFFAGE FILTRE
229	W1P11T41	MD 320	REAL	TEMP.MIN.CHALEUR DE FOND
230	W1P11T42	MD 324	REAL	TEMP.MAX.CHALEUR DE FOND
231	W1P21T41	MD 328	REAL	TEMP.MIN.CHALEUR DE FOND

Tableau 6.1: Les Mnémoniques du Système

6.2.3 Les blocs

Chaque bloc exécute une fonction (FC : fonction) ou plusieurs fonctions (FB : bloc fonctionnel) d'un équipement ou d'une alarme. La *figure 6.3* présente un ensemble de blocs système (OB : bloc d'organisation) et de blocs utilisateur. La séquence d'exécution des blocs est programmée dans le bloc système OB1, par des appels aux autres blocs.

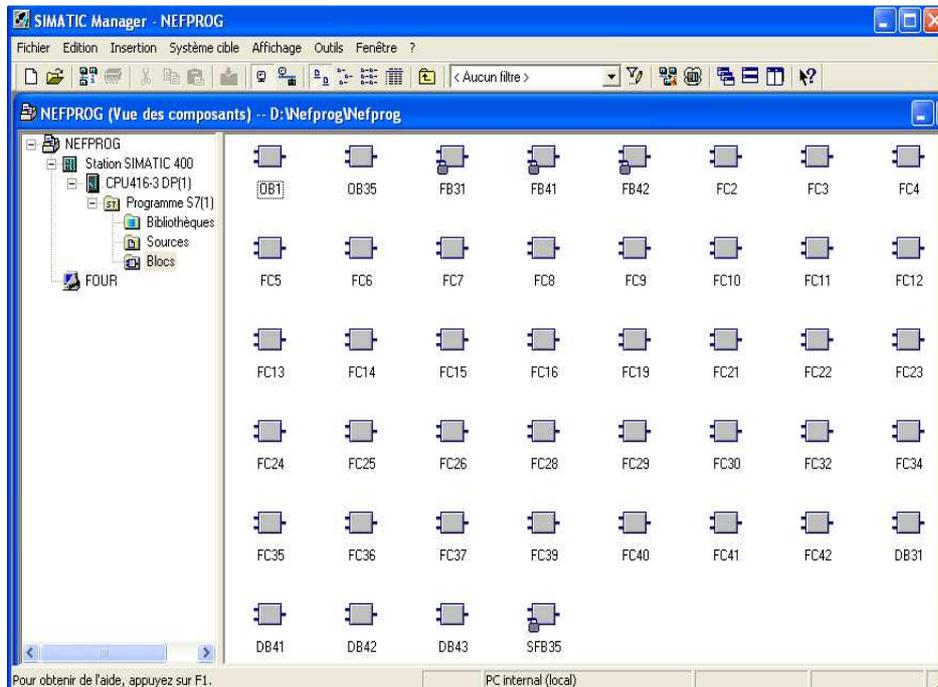


Figure 6.3: Blocs du programme S7

L'application du **GRAFCET** à l'étude du système automatisé séquentiel: clinkérisation. Toute solution logicielle, c'est-à-dire relevant de la logique programmée, doit prendre en compte le langage de programmation du système d'informatique industrielle dans lequel le GRAFCET doit être implanté.

Dans le programme STEP7, plusieurs fonctions d'automatisation sont utilisées sous forme par des blocs. Un automate programmable comporte une mémoire programmable pour le stockage des ces instructions comportant les fonctions suivantes (*figures 6.4, 6.5, 6.6*):

- ▣ logique séquentielle et combinatoire,

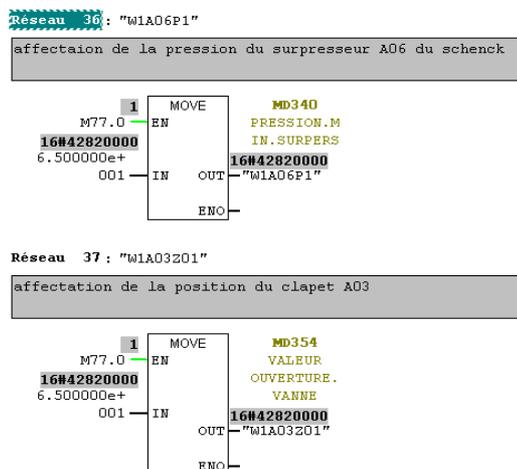


Figure 6.4: Forçage

✚ temporisation,

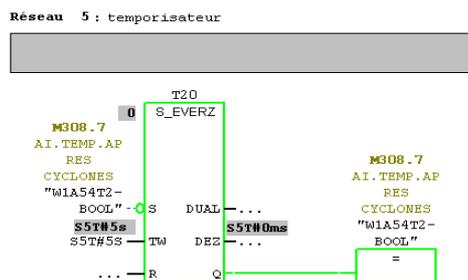


Figure 6.5: Temporisateur

✚ comptage, décomptage, comparaison,

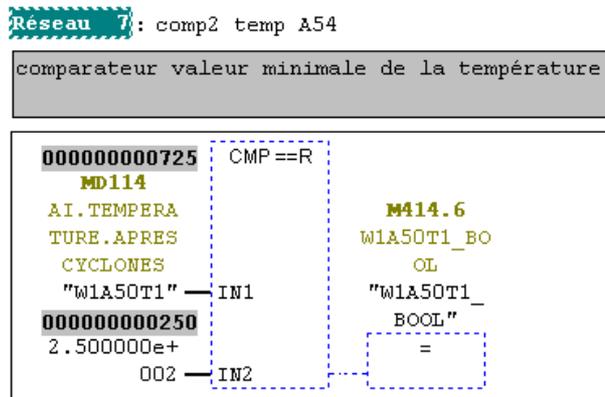


Figure 6.6: Comparaison

✚ calcul arithmétique

✚ réglage, asservissement ; régulation

6.2.4 Besoin de dialogue entre les opérateurs et la partie commande

La complexité des systèmes automatisés rend nécessaire le dialogue entre les opérateurs et la partie commande. Ce dialogue, encore désigné dialogue homme machine satisfait les besoins suivants :

- ✚ visualisation des valeurs numériques : visualisation en temps réel d'états, de valeurs numériques (nombre de pièces fabriquées, durée d'une opération, état de on fonctionnement d'un ventilateur,..)
- ✚ affichage des messages : affichage en temps réel des messages (textes en clair tels que : défaut pompe1, température +450°C, arrêt ventilation, fin de remplissage,..)

Avec une solution technologique évoluée cet affichage peut utiliser des images avec incrustation de messages et valeurs numériques.

- ✚ saisie : saisie d'ordres ou de consignes (types de pièces, temps de cuisson,...)
- ✚ réglage : accès en lecture ou écriture aux paramètres autorisés du programme utilisateur (réglage, d'un temporisateur, initialisation d'un compteur, ...)

Les informations échangées correspondent à des données :

- ✚ de type objet automate, valeur d'une temporisation, d'un compteur, défaut d'entrée ou de sortie,...
- ✚ de type application, valeurs ou messages élaborés par le programme.

6.3 Processus de clinkérisation

6.3.1 Alimentation Four

L'alimentation du four est la première étape du process. L'entrée ou l'alimentation du four par la farine crue, c'est le produit résultant de broyage cru et stocké dans des silos farine crue. Dans cette partie, un bloc pour la programmation du processus d'alimentation est réalisé et une vue sera affichée à l'opérateur pour surveiller la partie alimentation.

Pour cet équipement, un programme de la commande de marche de l'alimentateur est réalisé. La figure 6.7 suivante montre l'exécution du bloc de l'alimentation du four « SCHENCK ».

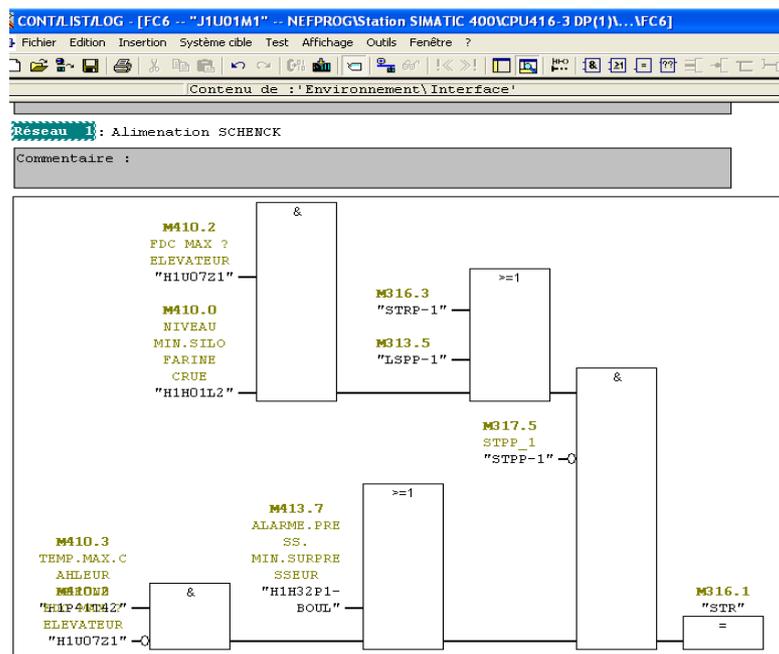


Figure 6.7: Commande de marche alimentateur Schenck

✚ Conditions de mise en marche

1. Schenck plein.
2. Four et Air lifts en marche.
3. Elévateur en marche.
4. Aérogliissière non plein

✚ Mise en marche

1. Vérification de la position correcte des registres
2. Démarrage Alimentation four
 - Démarrer Air lift ensuite le filtre.
 - Mettre alimentation synchronisée
 - Presser bouton démarrage.
3. Démarrage contrôle pesage.

6.3.2 Four

La première étape de démarrage du four est le préchauffement. Mais, avant le démarrage de la chauffe il faut s'assurer de la disponibilité des équipements nécessaires pour le démarrage, ensuite le four doit être chauffé d'une durée normalement de 18 à 24 heures.

La durée de chauffe est déterminée par le chef de service de process ou de fabrication. Pendant la durée de chauffage, on augmente l'ouverture de la buse centrale (Cône bruleur) progressivement.

6.3.2.1 Conditions de mise en marche

✚ Conditions pratiques

- toutes les portes sont fermées.
- contrôler que les analyses de GAZ fonctionnent.
- contrôler l'arrivée d'eau.
- toutes les opérations d'entretien mécanique et électrique sont achevées (tous les ateliers ont signalé que le four est prêt).
- contrôler que l'équipement du bruleur à GAZ est connecté.
- assurer que les broyeurs à cru seront prêts à fonctionner et qu'il y a assez de farine crue.
- démarrer la station de pompage hydraulique W08.

✚ Conditions Electriques

- il n'y a pas non plus verrouillage du transport clinker.
- le four n'est pas verrouillé à partir d'autres ateliers.
- émet un signal à l'atelier d'homogène signifiant que l'alimentation peut commencer.

✚ Autres conditions

- Transport poussière : Air lift H1U07, W1A13, suppresseur W1A21 en marche et silo d'homogénéisation ouvert.
- Moteurs du four doivent être démarrés (moteurs vireurs arrêter) et que le four ne se trouve pas position critique max 2.
- Le laboratoire doit être prévenu du démarrage, un échantillon doit être prélevé de l'alimentation du four immédiatement, pour connaître la faculté de cuisson de la farine crue.

✚ Mise en marche

- airs lifts en marche.
- démarrer transport poussière.
- démarrer les pompes à eau de la tour de conditionnement.
- électro-filtre mis sous haute tension.

✚ Marche normale

- laisser marcher le four.
- essayer de maintenir une production construite régler sur d'autres valeurs pour maintenir la marche du four aussi stable que possible.
- à partir de ces valeurs et la composition chimiques, régler le détroit de gaz, ensuite le tirage.

Dans ce qui suit, des exemples (*figures 6.8, 6.9 et 6.10*) de commandes de marche des équipements du process sous STEP7.

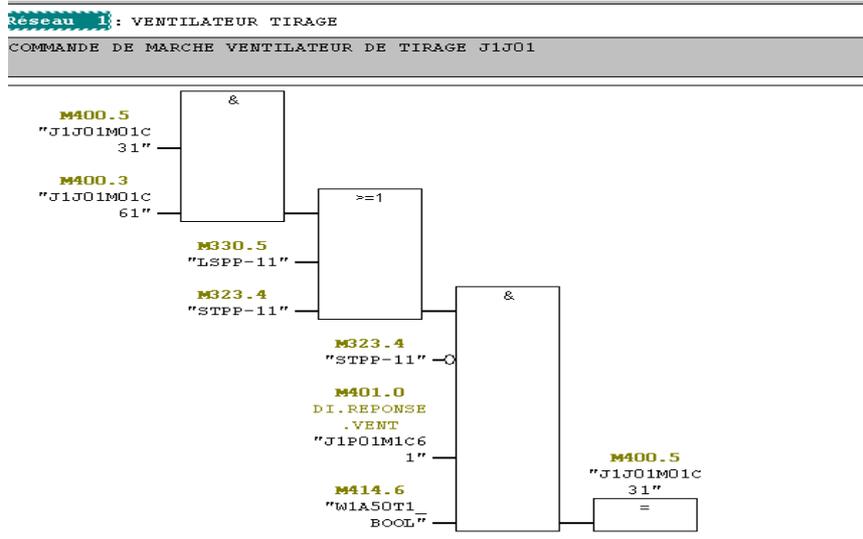


Figure 6.8: Commande de marche de ventilateur de tirage

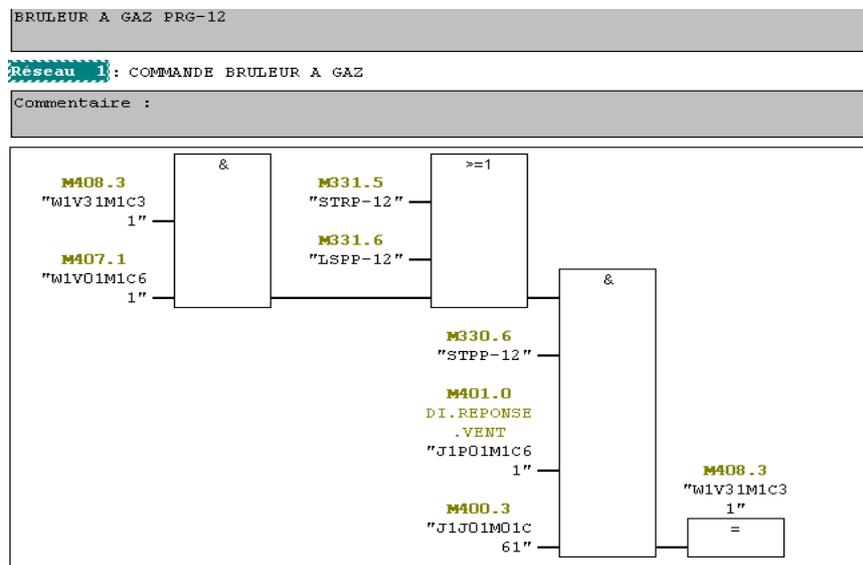


Figure 6.9: Commande de marche de bruleur à gaz

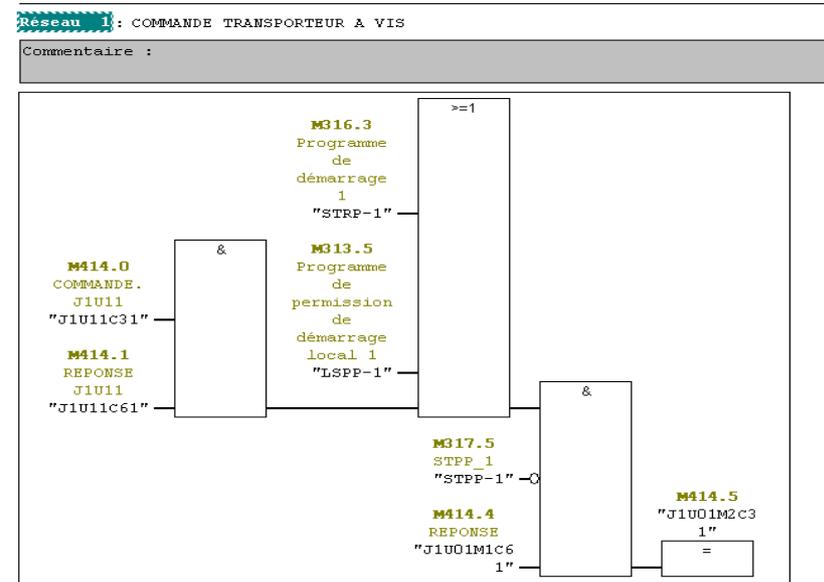


Figure 6.10: Commande de marche de transporteur à vis

6.3.2.2 Séquence de démarrage

Dans le cas où toutes les conditions de démarrage du four sont disponibles, le four peut être démarré selon une séquence de démarrage (figure 6.11).

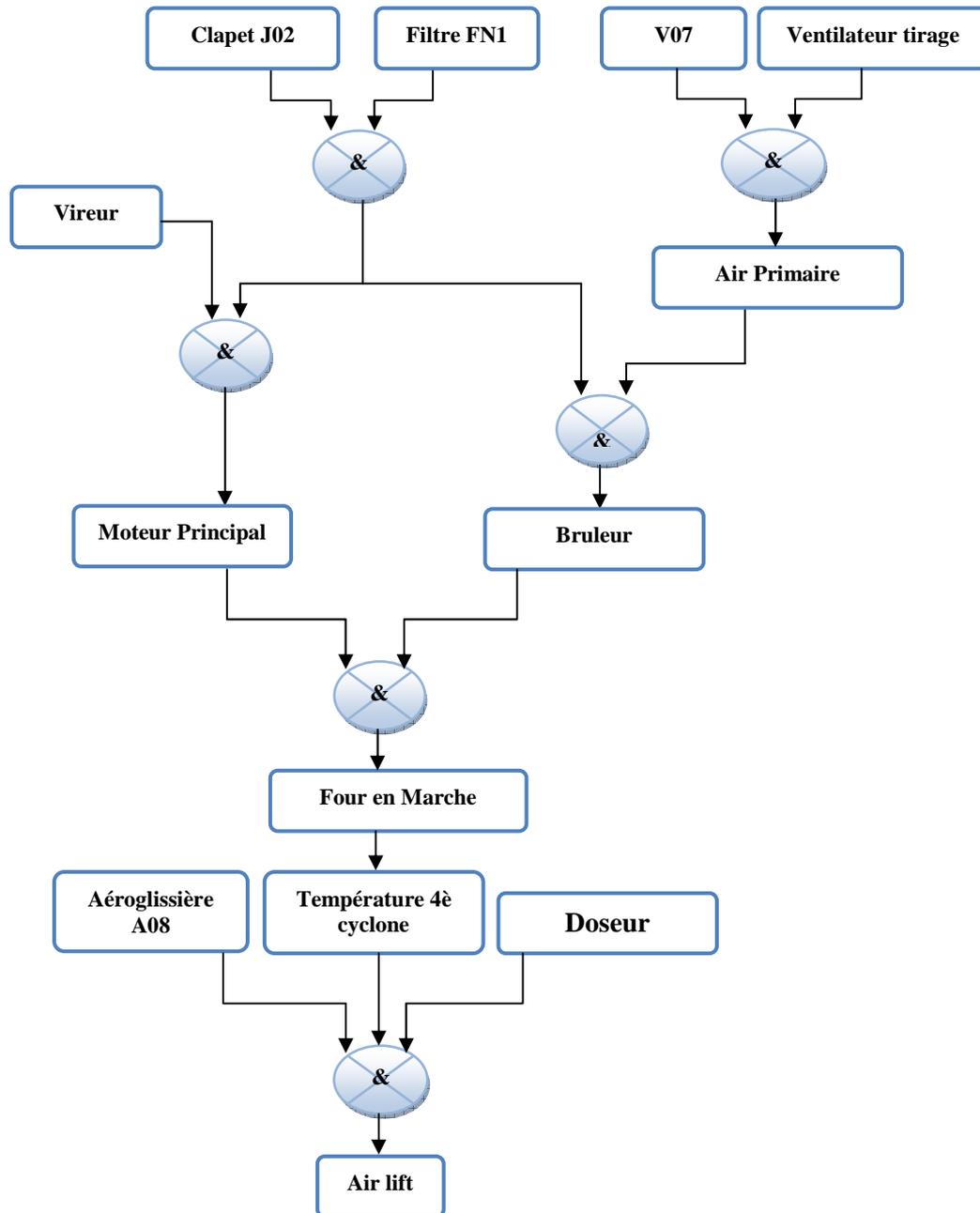


Figure 6.11: Séquence de démarrage de l'atelier de clinkérisation

Chaque équipement doit être soumis à trois types de verrouillage : verrouillage de marche, verrouillage de sécurité et verrouillage de protection.

6.3.2.3 Mise en Arrêt (Refroidissement du four)

A l'arrêt du four, on procède au refroidissement comme suit :

1. Fermer le registre J1/2J02 complètement,
2. Virage du four.

Durant de fortes pluies le four doit être viré continuellement. Le refroidissement du four doit se faire aussi lentement que possible, risque de chute de briques et déplacement en spirale de la brique.

La *figure 6.12* suivante représente l'arrêt du programme PRG13, déclenche un arrêt du four.

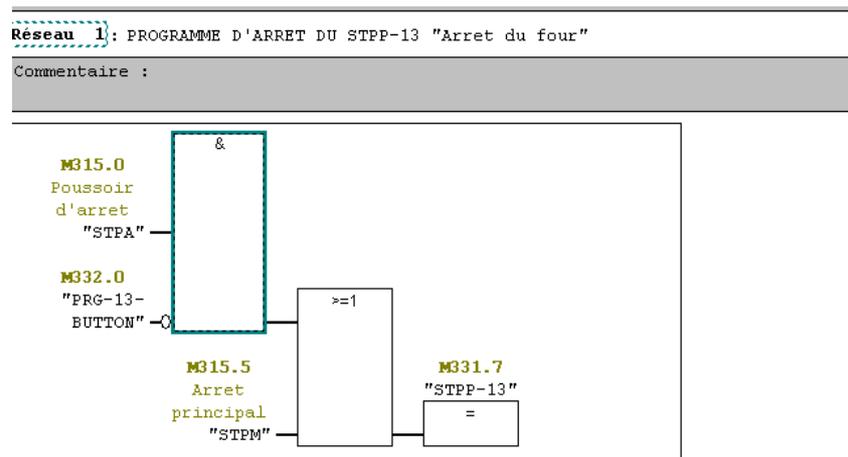


Figure 6.12: Arrêt du programme PRG13

6.3.2.4 Les alarmes

La notion alarme est introduite dans un système de diagnostic, par contre un système de pronostic la connaît comme une pré-alarme. Les deux notions permettent l'avertissement de l'opérateur qu'il y a des défaillances. La *figure 6.13* ci-dessous illustre une différence de deux notions précédentes.

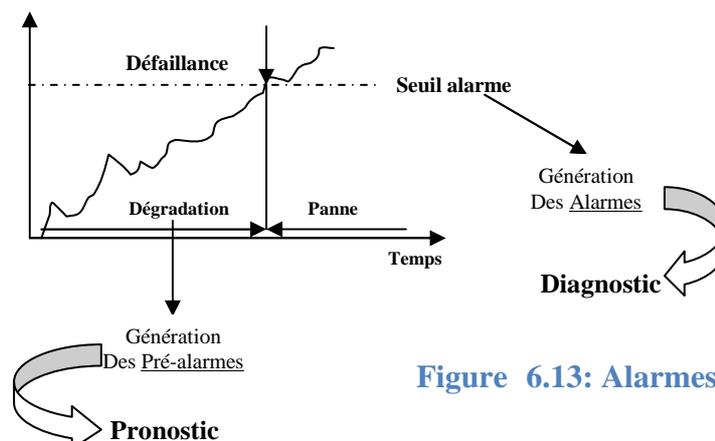


Figure 6.13: Alarmes et Pré-alarms

Les pré-alarms permettent le contrôle du système et processus à travers des éléments clignotants. Ces derniers déclenchent des alarmes qui sont journalisées dans la liste des alarmes, mais les informations relatives à ces alarmes sont stockées dans la base de données des événements.

La figure 6.14 montre un ensemble des pré-alarmes qui peut déclencher une ou plusieurs alarmes.

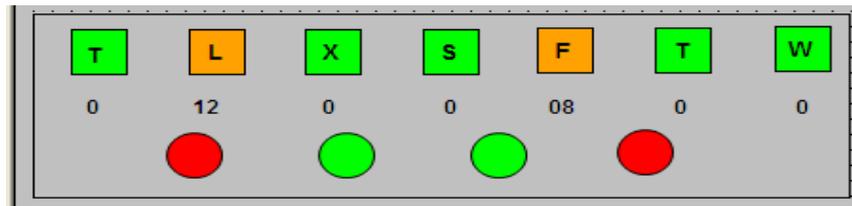


Figure 6.14: Traitement des pré-alarmes

6.3.3 Boucles de régulation

La régulation, une des tâches de l'automate, est une méthode de résolution de tâches de régulation complexes. Les processus dont la description mathématique est complexe et pour lesquels il n'est pas possible de calculer un modèle théorique simple ne peuvent pas être traités avec des régulateurs classiques.

Pour cela, le process de la cuisson plusieurs régulations sont réalisées. Les régulateurs utilisés dans notre système sont de trois type : un régulateur classique (Dédit gaz), un régulateur flou (alimentation four) et un régulateur neuro-flou (température four).

6.3.3.1 Régulation classique

Le WINCC nous a permis de créer l'interface d'un régulateur et afficher ces commandes à partir d'une *Template* illustrée à la figure 6.15 Suivante :



Figure 6.15: Template régulateur classique

Ensuite, créer toutes les variables nécessaires de cette régulation. La figure 6.16 suivante donne l'outil 'Stock de variables' pour créer les variables du régulateur.

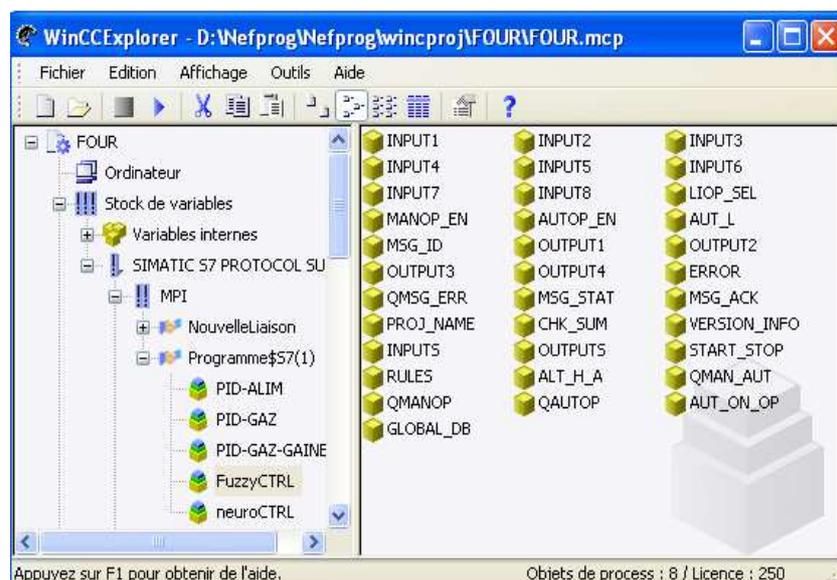


Figure 6.16: Stock de variables de régulateur

Le bloc de régulateur classique est représenté à la figure 6.17 Ci-dessous :

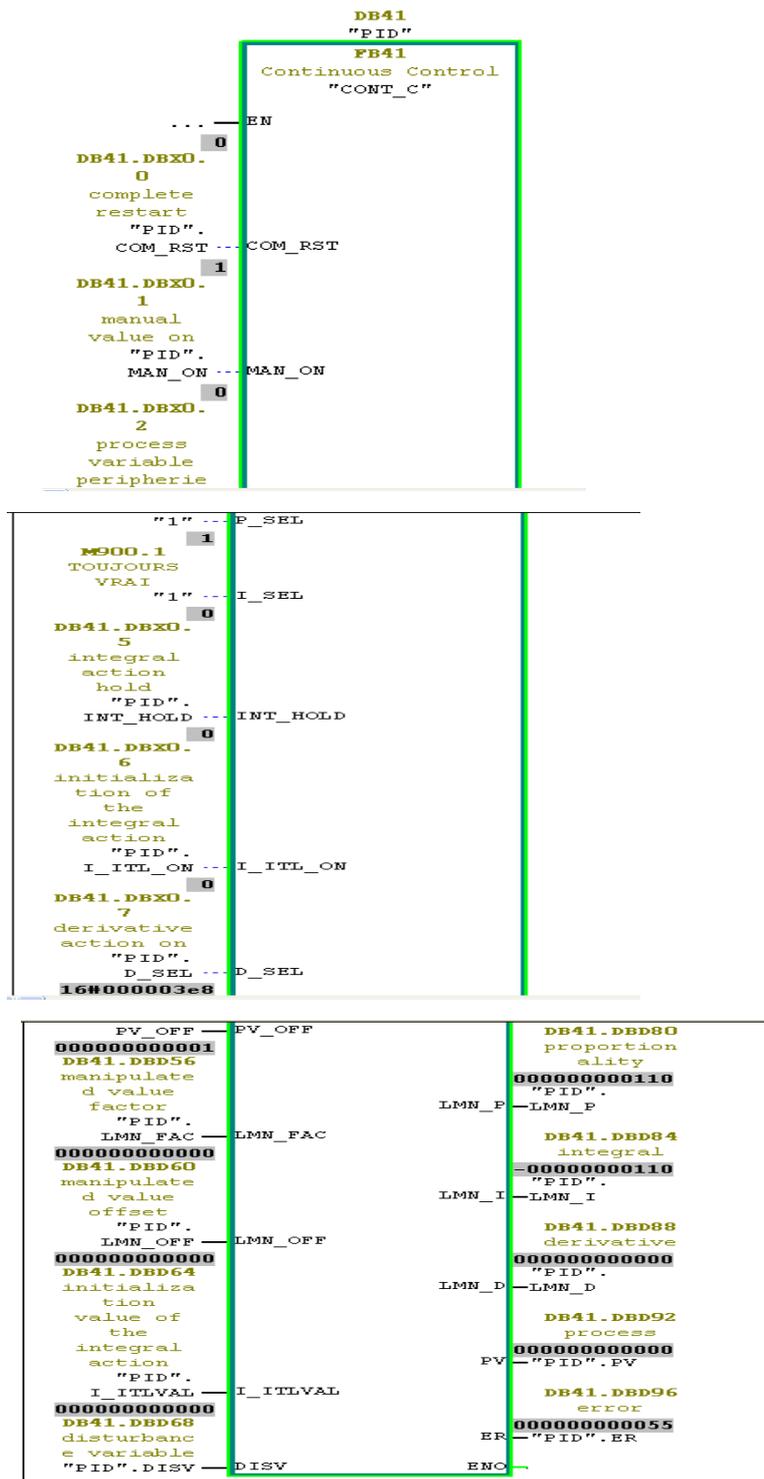


Figure 6.17: Bloc d'un régulateur classique

6.3.3.2 Régulation Floue

Dans cette partie, on a besoin d'un outil de la logique floue. Une Template (figure 6.18) nous a permis de lier les données et les blocs de STEP7 aux celles du programme Fuzzy Control++.



Figure 6.18: Template régulateur fluu

Le DB31 (figure 6.19) est le bloc de régulation floue sur STEP7.

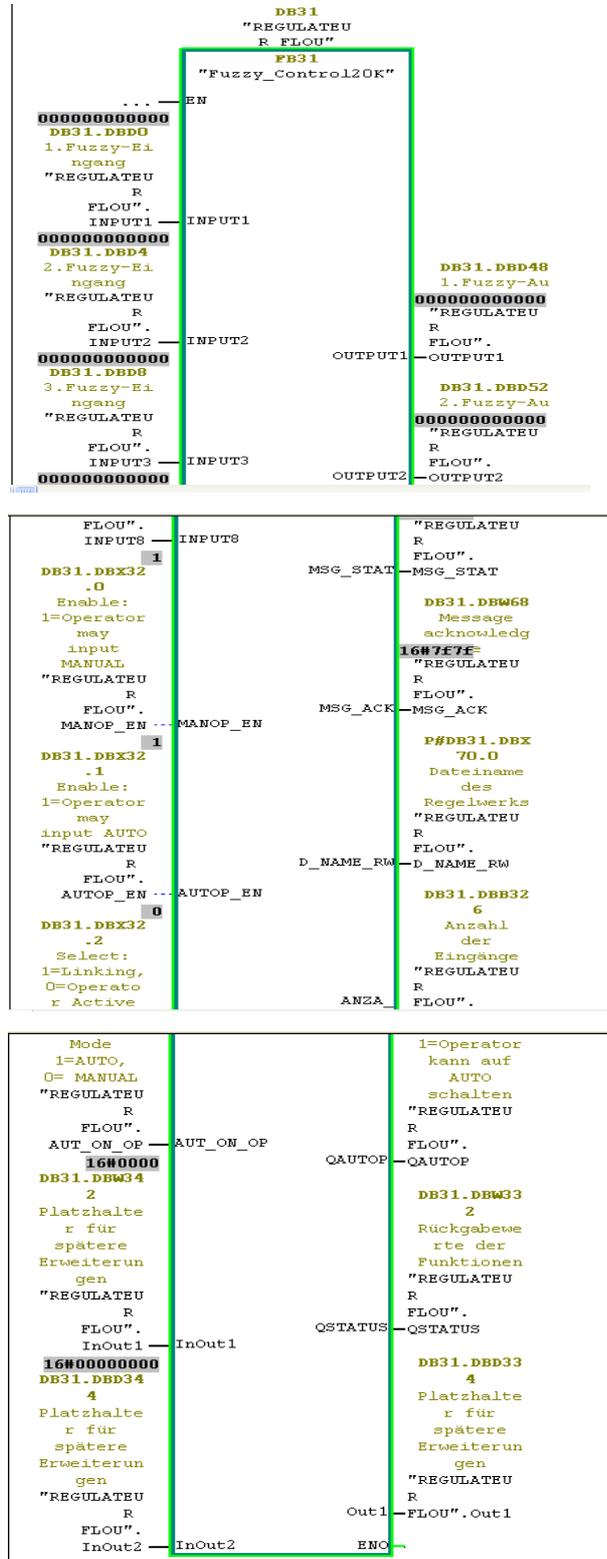


Figure 6.19: Bloc d'un régulateur Flu

Il faut vérifier la connexion entre l'outil de configuration FuzzyControl++ et le bloc dans le programme STEP 7. L'image intermédiaire est le bloc DB31. La figures 6.20 suivante donne lieu le DB31 'régulateur fluu'.

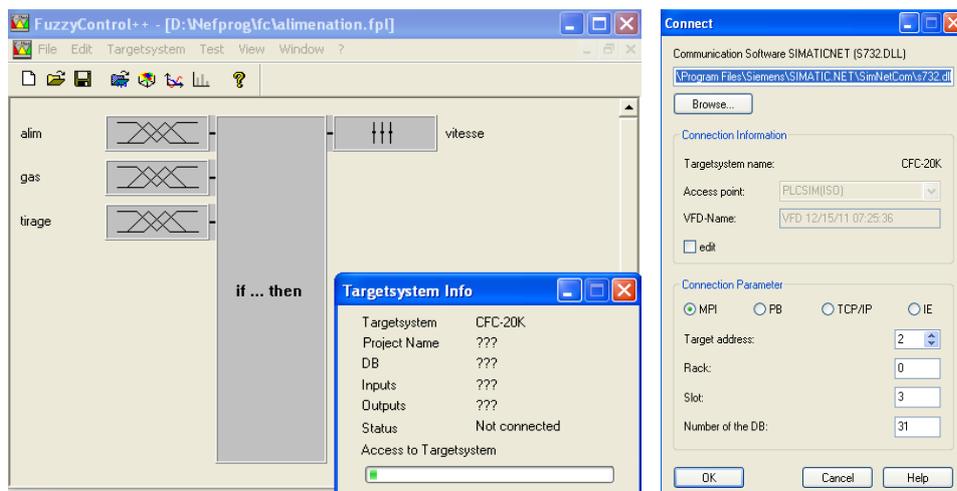


Figure 6.20: Connexion Fuzzy Control++ et DB31

Les deux figures 6.21 et 6.22 suivantes montrent les ensembles flous de l'entrée tirage et la sortie vitesse de four.

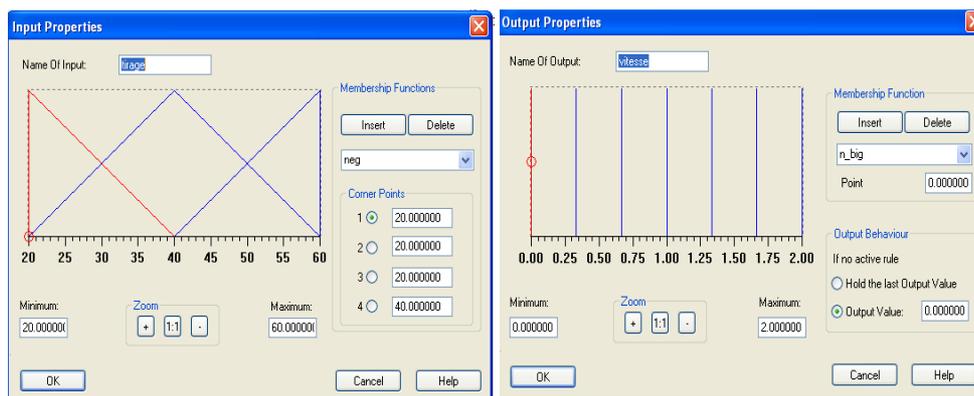


Figure 6.21: Ensemble flou Tirage

Figure 6.22: Ensemble flou Vitesse

La figure 6.23 montre la table des règles floues déjà citées :

	1	2	3	4	5	6	7	8
alim	p_small	n_big	n_mediu	n_big	n_mediu	n_small	n_small	n_mediu
gas								
tirage	null	pos	null	null	neg	pos	neg	pos
vitesse	n_small	n_small	null	null	n_small	n_mediu	n_mediu	n_small

Figure 6.23: Table Règles Floues

La courbe de régulation de vitesse selon l'alimentation et gaz apparaît sur la figure 6.24 par l'outil FuzzyControl++.



Figure 6.24: Courbe de régulation

La surface générée par la boucle de régulation floue de vitesse de la virole est représentée à la figure 6.25 suivante :

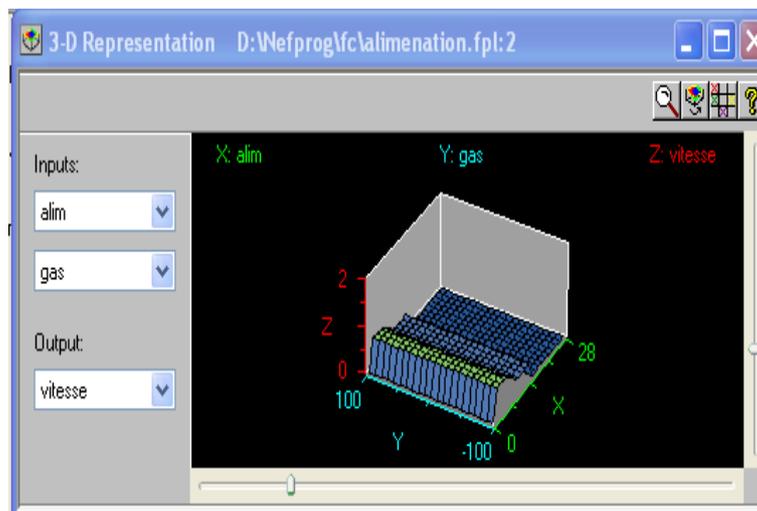


Figure 6.25: Surface générée par la boucle de régulation

6.3.3.3 Régulation neuro-floue

Dans cette partie, on a besoin d'un outil de la logique floue. Une Template (figure 6.26) nous a permis de lier les données et les blocs de STEP7 aux celles du programme NEURO SYSTEMS.



Figure 6.26: Template régulateur Neuro-flou

Le DB32 (figure 6.27) est le bloc de régulation neuro-floue sur STEP7.

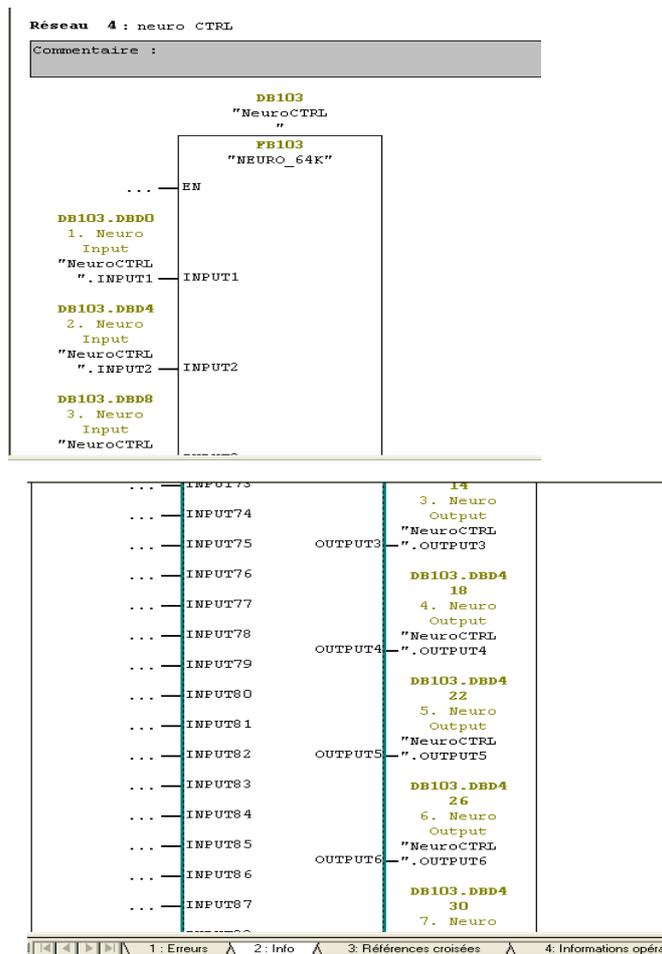


Figure 6.27: Bloc d'un régulateur Neuro-Flou

Il faut vérifier la connexion entre l'outil de configuration FuzzyControl++ et le bloc dans le programme STEP 7. L'image intermédiaire est le bloc DB32. La figure 6.28 suivante donne lieu le DB32 'régulateur Neuro-flou'.

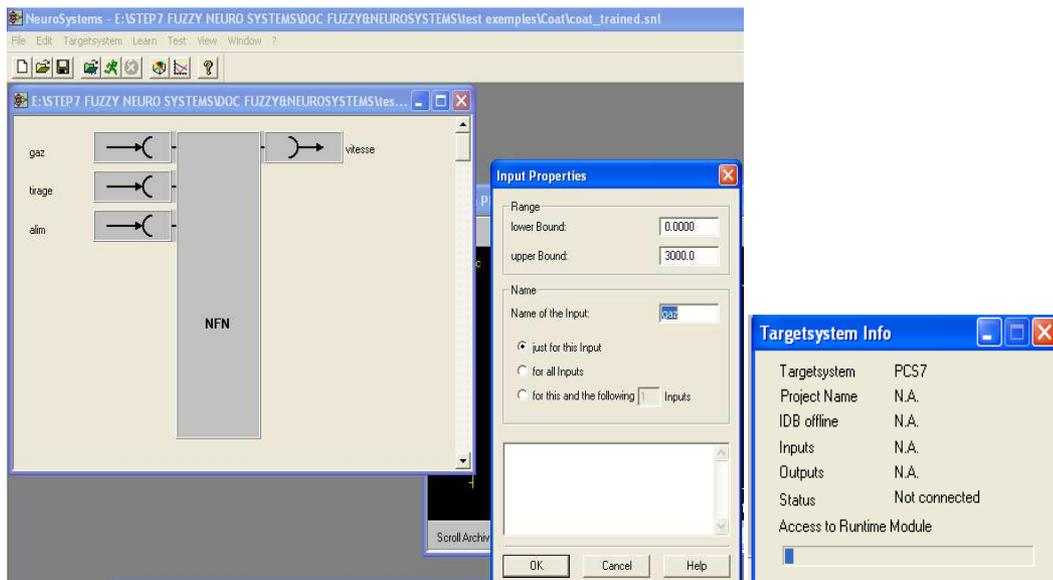


Figure 6.28: Connexion NeuroSystems et DB32

Principe de fonctionnement

Les réseaux de Neuro flou se composent de trois sous-réseaux neuronaux (NSN) (figure 2.29) qui le fuzzification, l'inférence et le défuzzification modèles.

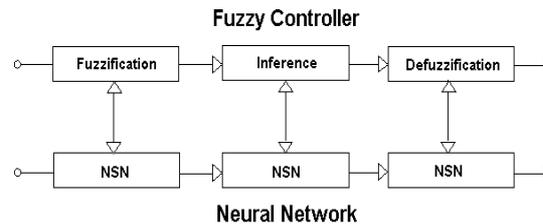


Figure 2.29: Connexion Fuzzy Control++ et NeuroSystems

Si vous sélectionnez ce type de réseau, les fonctions d'appartenance floues des entrées et des sorties sont automatiquement adaptées pendant l'apprentissage, et une base appropriée de règle est produite.

Pour chaque entrée et rendement (variables linguistiques), vous devez définir le nombre de fonctions d'appartenance (valeurs linguistiques, ensembles flous) qui représentent le degré d'appartenance des valeurs d'une gamme permise de valeur pour les valeurs linguistiques. Des fonctions trapézoïdales sont produites comme fonctions d'appartenance d'entrée.

La méthode d'inférence employée est l'inférence de Sugeno (inférence de Sum-Min). Avec l'importation, vous pouvez importer un système brouillé (.fpl) configuré par FuzzyControl. NeuroFuzzy (NFN) a une restriction d'un maximum de 8 entrées et de 4 sorties. Le produit de tout le nombre de fonctions d'appartenance d'entrée doit être moins ou égale à 2000.

Les figures 6.30 et 6.31 suivantes représentent les propriétés de l'entrée ainsi que le temps et nombre d'itérations de l'apprentissage.

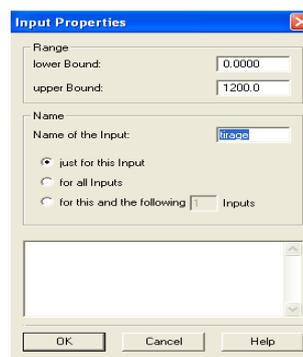


Figure 6.30: Propriétés de l'entrée du réseau

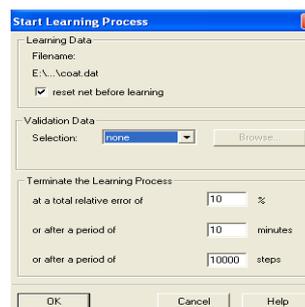


Figure 6.31: Temps et Nombre d'itérations de l'apprentissage

Après l'initialisation du réseau, une phase de l'apprentissage est introduite. Le fichier de l'apprentissage est enregistré, on peut le consulter et voir la trace de l'apprentissage utilisant le NOTPAD. La *figure 6.32* illustre la trace de l'apprentissage.

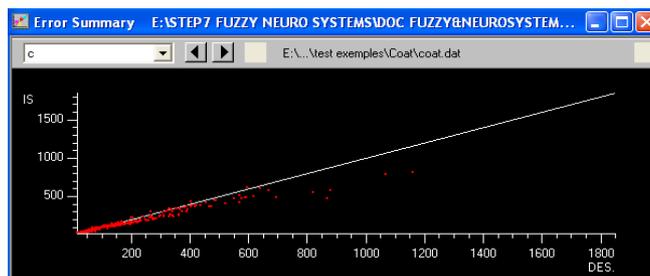


Figure 6.32: Trace de l'apprentissage

La surface générée par la boucle de régulation est représentée à la *figure 6.33* suivante :

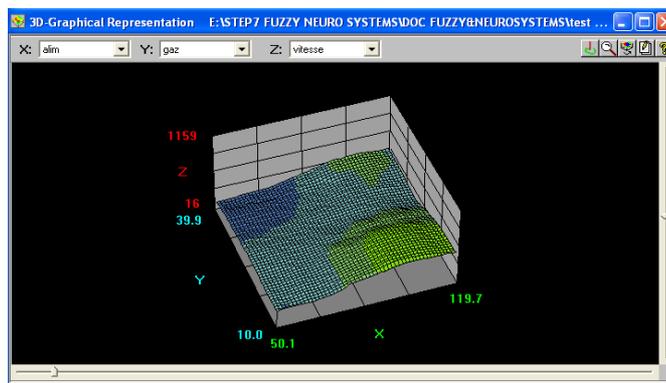


Figure 6.33: Surface générée

La courbe de régulation apparaît à la *figure 6.34* par l'outil NeuroSystems.

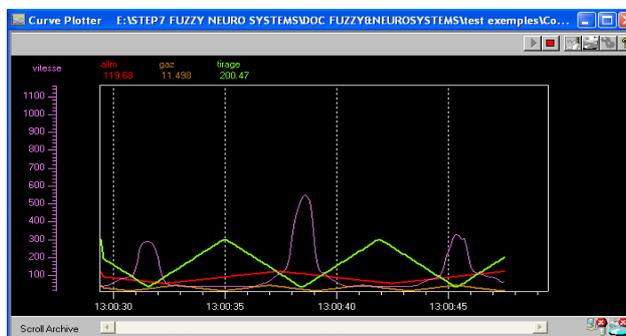


Figure 6.34: Courbe de régulation

La *figure 6.35* illustre le test de réseau, c'est la comparaison de fichier de l'apprentissage avec la base de test.

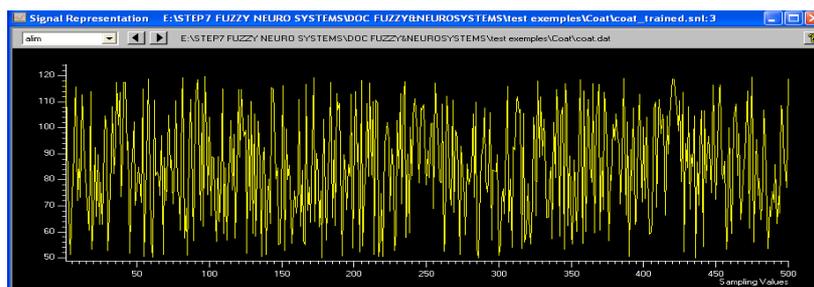


Figure 6.35: phase de test

6.3.3.4 Régulation par ANFIS

Un exemple de régulation neuro floue utilisant ANFIS de Matlab pour comparer les résultats obtenus par différents systèmes.

C'est la synchronisation de Schenck. Pour cela, on a trois paramètres d'entrée et une seule sortie, c'est la vitesse e la virole (max= 2 tour par minute).

Avant d'appliquer l'ANFIS, il faut créer l'ensemble flou par l'outil Fuzzy Control de Matlab.

La *figure 3.36* donne les différents paramètres utilisés (les entrées el la sortie).

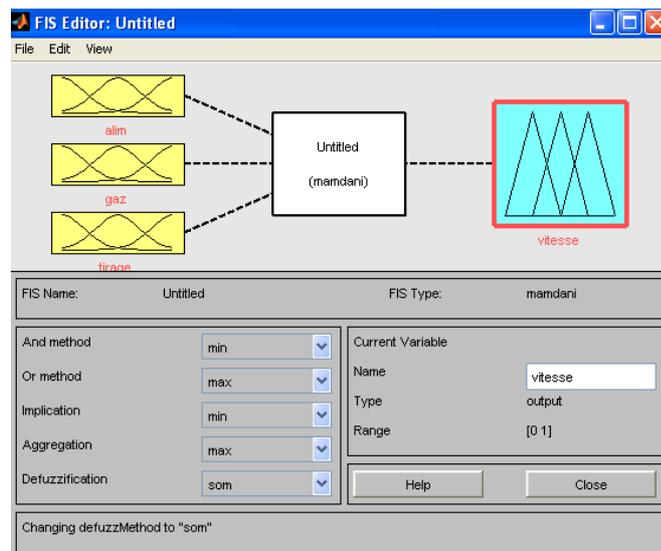


Figure 6.36: Les ensembles flous

Après la spécification de l'ensemble flou, on va créer les fonctions d'appartenance de chaque variable (*figure 6.37*).

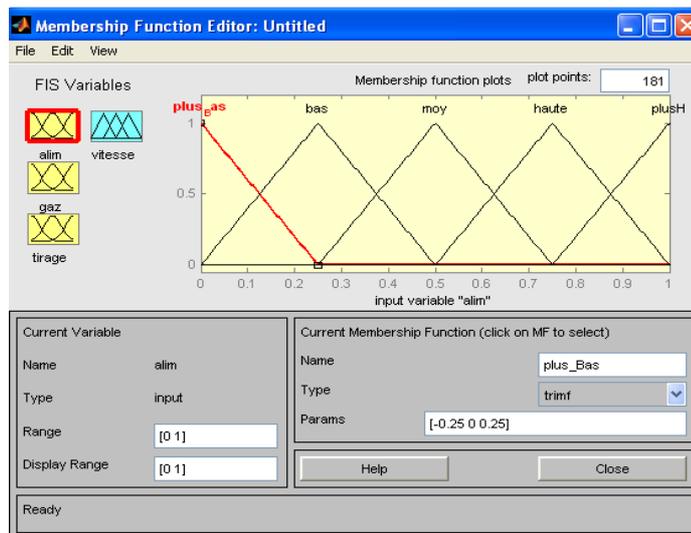


Figure 3.37: les fonctions d'appartenance

La table des règles floues est représentée par la *figure 6.38* suivante :

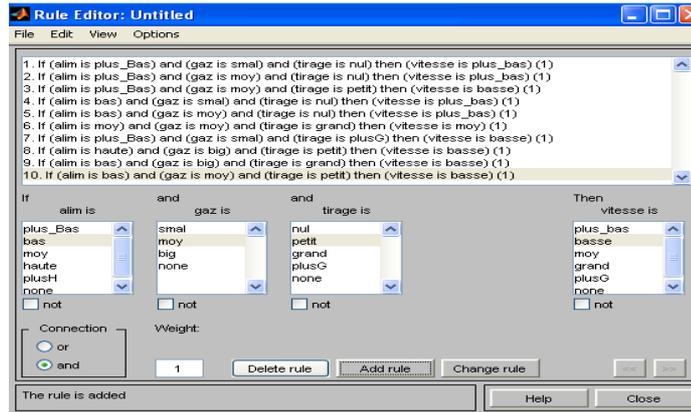


Figure 6.38: Table de règles floues

Ensuite, un chargement du fichier de la logique floue à l'aditeur ANFIS pour appliquer l'apprentissage. La *figure 3.39* représente l'interface de ANFIS.

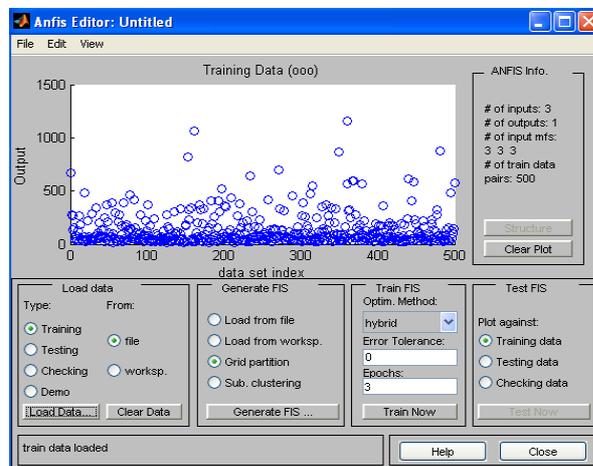


Figure 3.39: Chargement du fichier Fuzzy sur ANFIS Edit

L'étape suivante est l'apprentissage et le test des erreurs (*figure 3.40*).

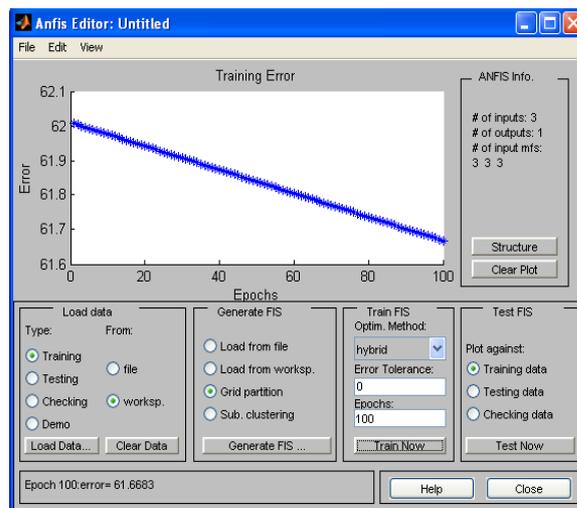


Figure 3.40: Apprentissage par ANFIS

6.4 Présentation NEFPROG

6.4.1 Interface générale

Notre système Neuro-Flou Temporel pour le pronostic de l'atelier de clinkérisation « *NEFPROG* » est lancé via une interface générale (figure 6.41) permettant l'accès à toutes les vues existant au niveau du système soit par des boutons affichés sur l'interface (fenêtre) ou à partir des images du process : Alimentation, Four, Refroidissement.

La figure ci-dessous illustre l'interface générale de notre système « *NEFPROG* » :

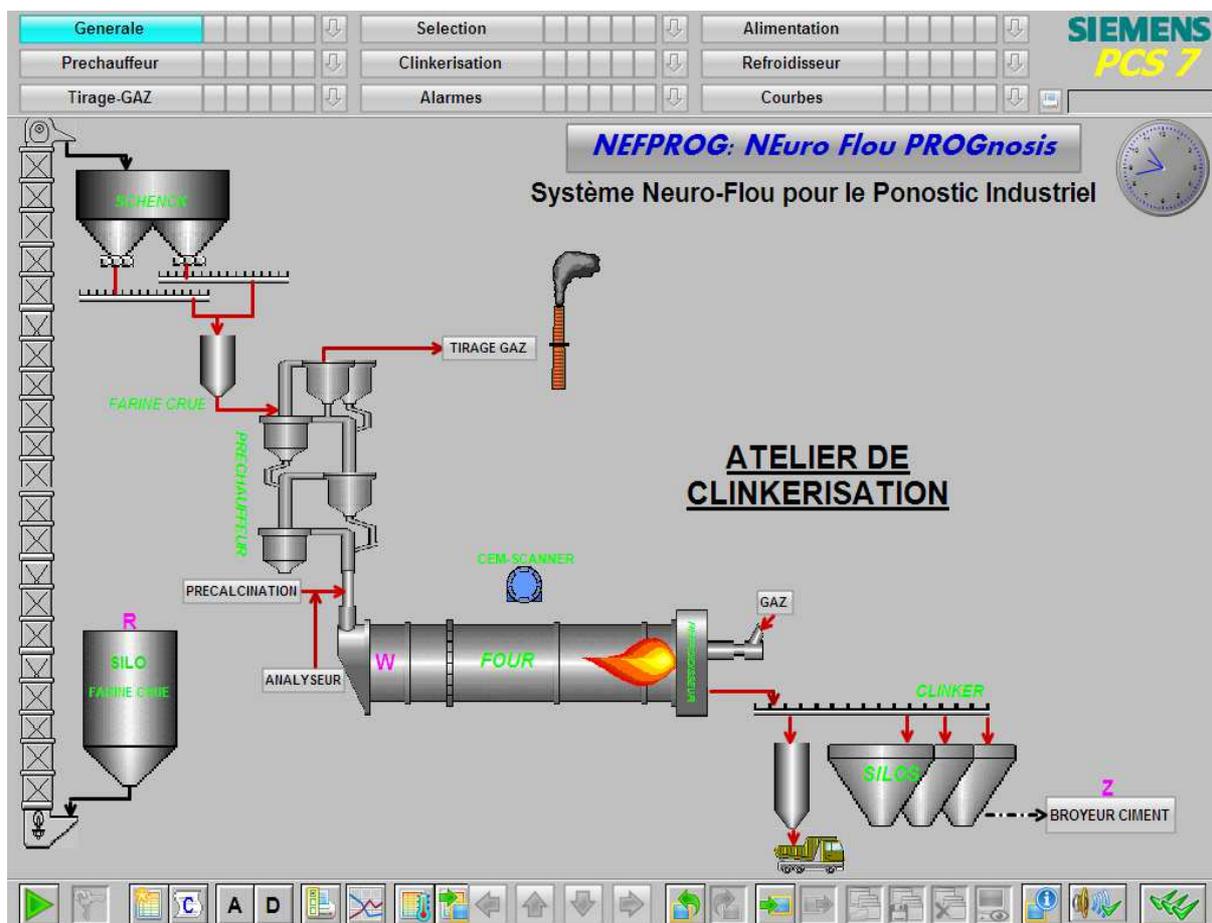


Figure 6.41: Interface générale du NEFPROG

L'opérateur peut agir sur plusieurs vues, il permet d'afficher les alarmes, les courbes et les archives. Pour quitter l'interface complètement, un clic sur le bouton  ensuite le bouton .

6.4.2 Sélection des équipements

Un clic sur le bouton Sélection, une fenêtre de sélection des groupes et des équipements s'affiche (figure 6.42). Elle est réalisée par la sélection des groupes, chaque groupe contient un nombre d'équipements ordonnés selon la séquence de démarrage. Le démarrage du groupe signifie le démarrage des équipements inclus.

Donc, la marche du four suit un démarrage séquentiel des équipements inversement de leurs marche. C'est-à-dire, le dernier dans le process démarre le premier pour assurer son fonctionnement.



Figure 6.42: Sélection des équipements

6.4.3 Alimentation Four

La figure 6.43 représente l'alimentation du four. Une synchronisation de débit de matière première (farine crue) et le tirage gaz avec le nombre de tour du four, selon les données du tableau de synchronisation (cahier des charges).

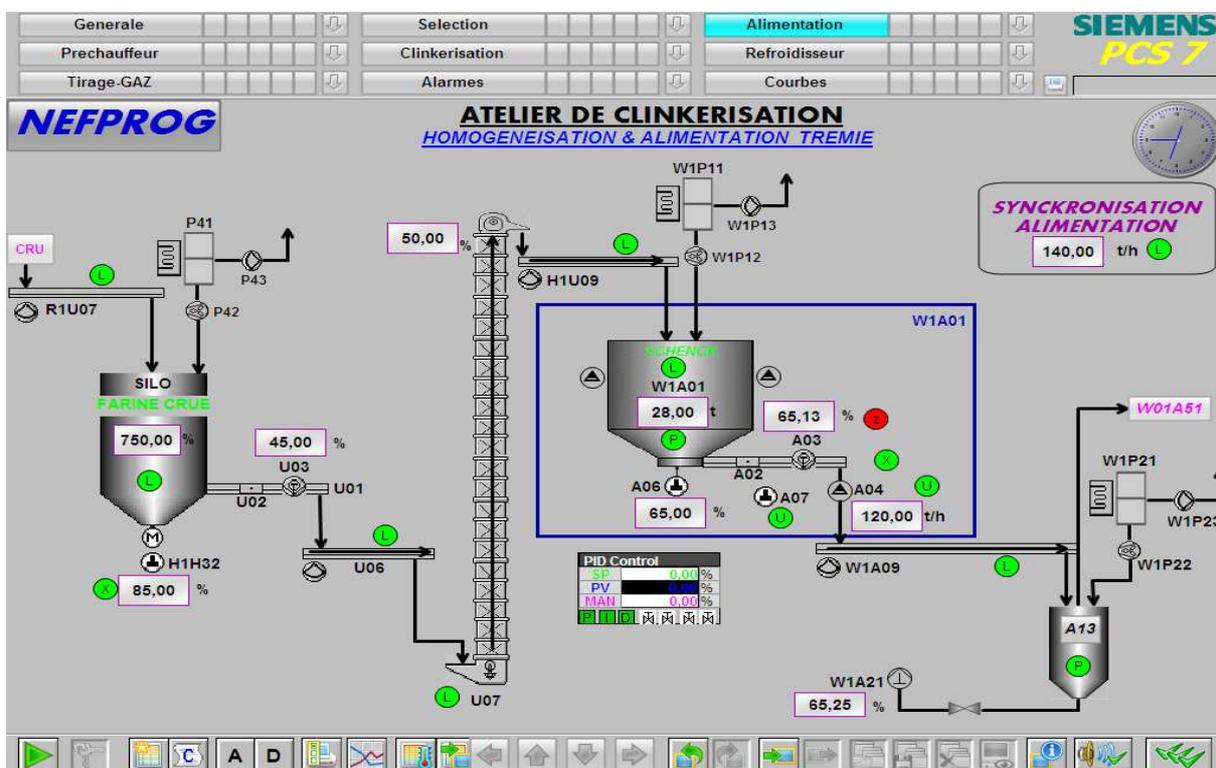


Figure 6.43: Alimentation four

L'opérateur peut modifier le point de consigne voir les courbes. Et aussi, il peut modifier le type de régulation (figure 6.44).

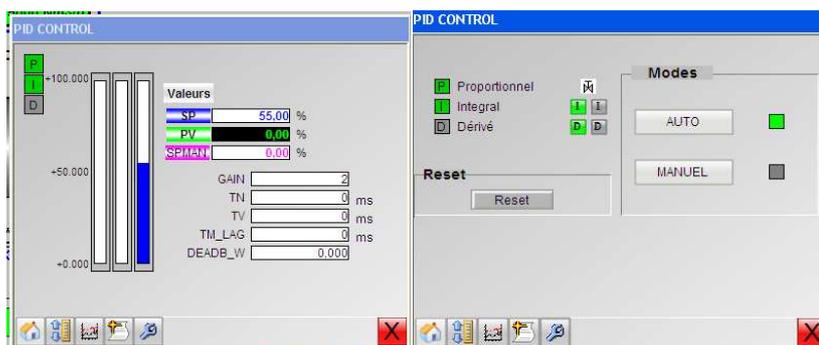


Figure 6.44: Interface régulateur

6.4.4 Préchauffement du four

C'est une étape importante pour la chauffe du four. L'opérateur peut voir les différents paramètres du préchauffement (figure 6.45).

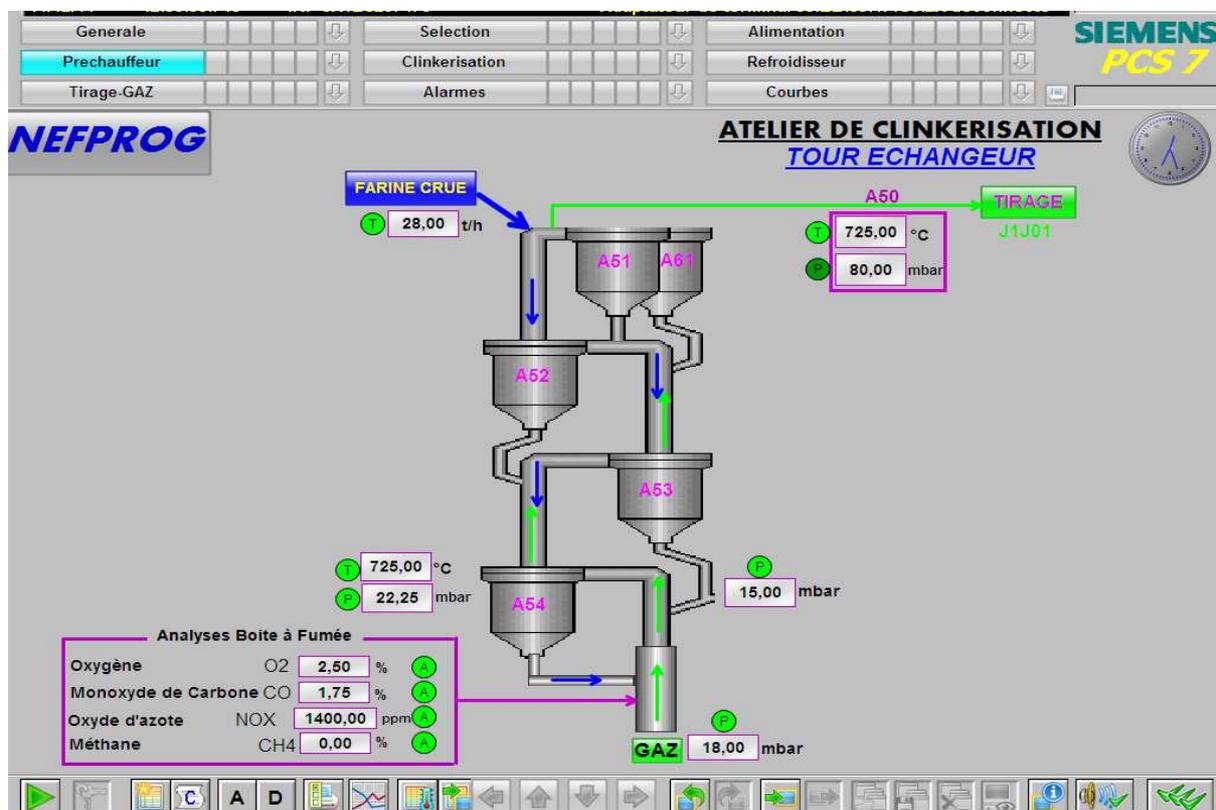


Figure 6.45: Tour de préchauffement

6.4.5 Cuisson

La vue e la cuisson est présentée sur la figure 6.46. Plusieurs paramètres sont affichés, soit ces derniers sont entrés manuellement, soit sont calculés à partir d'autres paramètres.

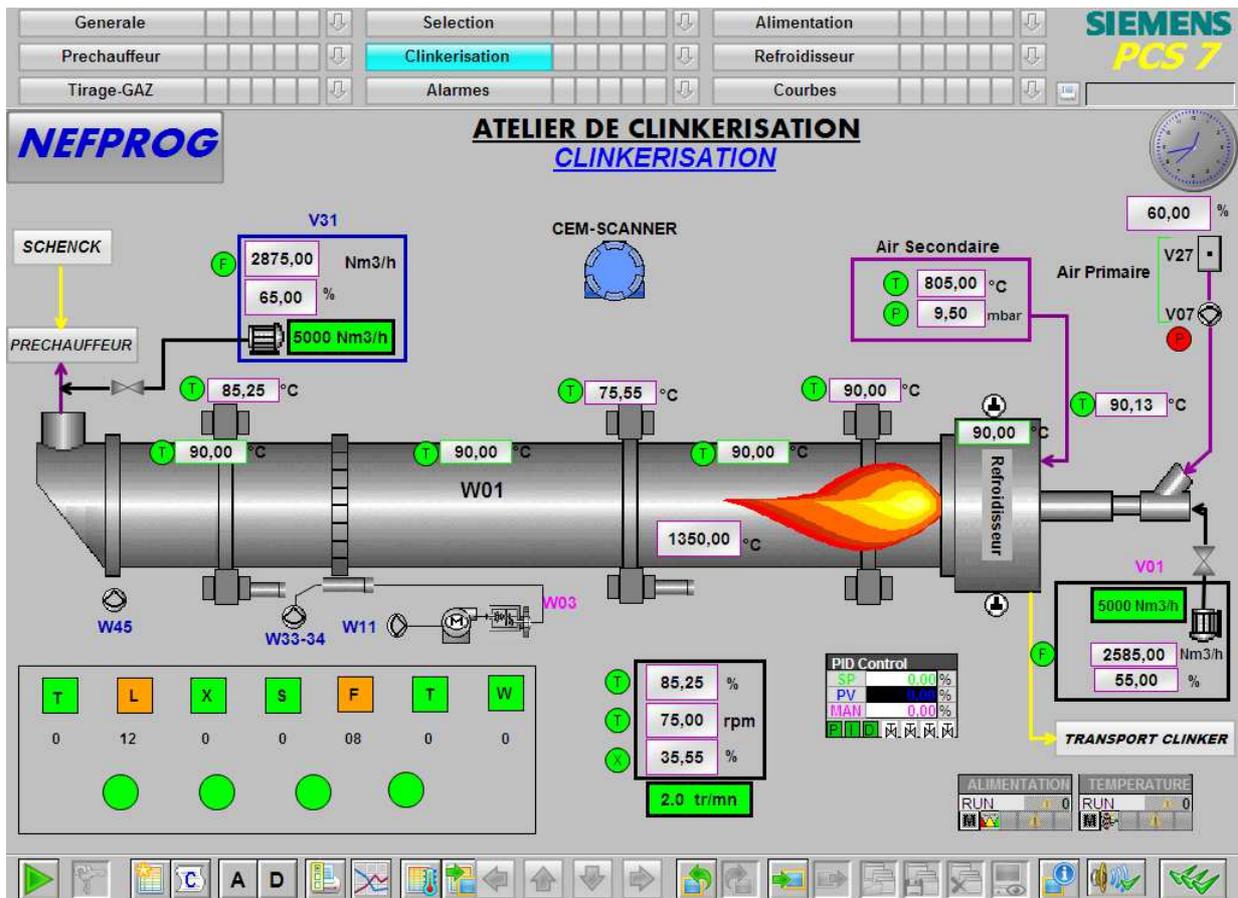


Figure 6.46: Zone de cuisson

Les paramètres sont contrôlés par des régulateurs, des pré-alarmes sont affichées s'il y a une ou plusieurs dégradations du système.

Après un avertissement à l'opérateur par une pré-alarme, avec une action préventive proposée pour régler le problème, l'opérateur peut intervenir pour un diagnostic prédictif pour éliminer cette pré-alarme, sinon, une alarme sera enregistrée et affichée au tableau des alarmes et dans ce cas, un diagnostic est obligatoire.

6.4.6 Refroidissement et stockage Clinker

La dernière étape du process de la clinkérisation est le refroidissement, ensuite le transport du produit fini (Clinker). La figure 6.47 représente le refroidisseur à ballonnets et le stockage du clinker dans des silos.

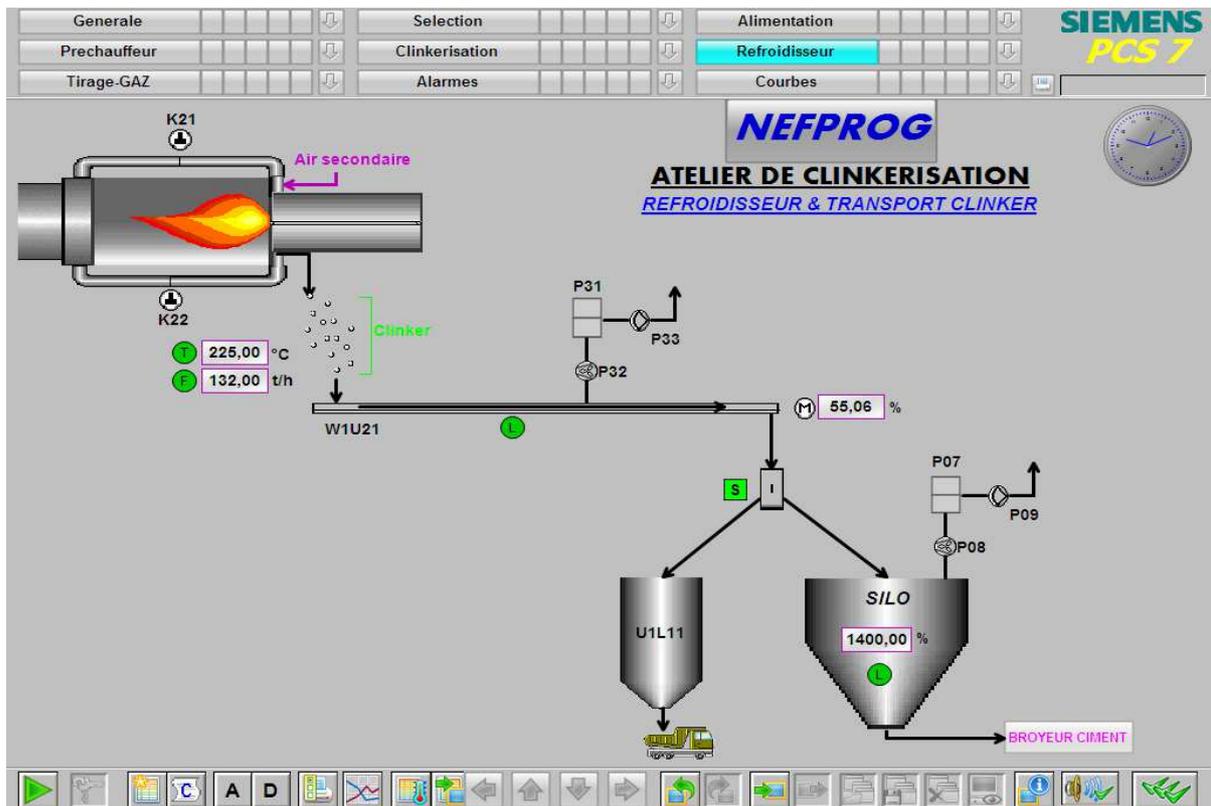


Figure 6.47: Refroidissement et stockage Clinker

6.4.7 Tirage Gaz

Le tirage à gaz déroule dans le sens inverse de la matière. Il permet de tirer les gaz et récupérer la matière première via un filtre (figure 6.48).

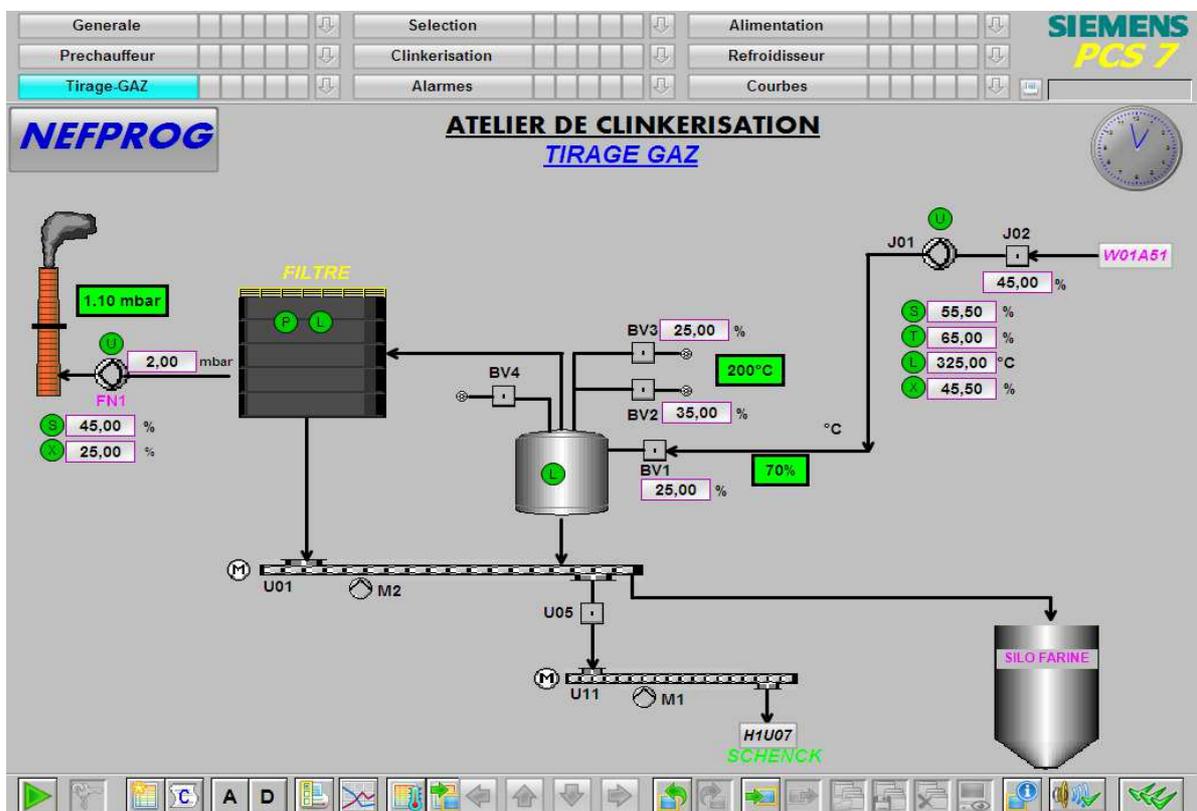


Figure 6.48: Tirage gaz

6.5 Conclusion

Le but de ce chapitre est de présenter l'implémentation d'un Progiciel « NEFPROG ». C'est une application au sein de la cimenterie d'Ain-Touta-Batna. Nous avons choisi l'atelier de clinkérisation comme un champ d'étude puisque c'est la zone sensible aux défaillances, et aussi à condition de l'amélioration d'une approche déjà étudiée.

Cette application est basée sur une nouvelle approche neuro-floue temporelle, c'est-à-dire l'intersection de réseau de neurones et de la logique floue pour le suivi du comportement du procédé industriel et la détection et l'estimation du temps avant défaillance d'un équipement.

CONCLUSION
GENERALE

L'activité de pronostic devient un élément clef des stratégies de maintenance industrielle. Il a la capacité à détecter et à isoler les défauts imminents, ainsi qu'à prévoir la situation future d'un système. Donc, l'utilité d'un système de pronostic est liée à prédire l'évolution de la dégradation d'un équipement.

Le travail apporté dans cette thèse vise la spécification d'un cadre applicatif pour l'implémentation d'un système de pronostic capable de prédire le comportement d'un système tout en tenant compte de la dynamique des équipements réels.

Dans ce cadre, les principales contributions de ce mémoire sont regroupées en trois parties :

La *première partie* est un état de l'art de diagnostic et de pronostic industriel. Les points essentiels de cette partie sont :

- Une terminologie de diagnostic et de pronostic industriel.
- Le diagnostic industriel par RdF
- Les différentes approches de pronostic industriel.

Dans la *deuxième partie*, une présentation des outils de la Reconnaissance des formes pour le diagnostic industriel y a compris :

- La logique floue,
- Les réseaux de neurones classiques,
- Les réseaux de neurones dynamiques,
- Les réseaux Neuro-flous.

La *troisième partie* a consacré à l'application des outils précédents pour le suivi du comportement d'un système de production et prédire les situations anormales, avant que le système ait provoqué une panne. Alors, cette partie concerne l'exploitation industrielle, c'est l'utilisation de l'approche neuro-floue temporel.

Dans le cas des processus industriels, une grande quantité des pannes et une diminution de la production sont en général le résultat d'une défaillance dans le procédé et la solution est de la *corrigée à temps*.

Pour cela, une nouvelle approche Neuro-floue temporelle pour le pronostic industriel est réalisée : NEFPROG. Elle permet de prédire les dégradations ou les défaillances futures, d'un équipement en temps réel au sein du LAP-Batna et appliqué à l'atelier de clinkérisation à condition de l'élévation de nombres de pannes de la zone du four par rapport aux autres zones.

Ce travail a permis d'ouvrir les perspectives suivantes :

- 🚧 Un premier point concerne l'amélioration des performances de notre système neuro-flou, par l'ajout du concept réseau distribué. L'interface de NEFPROG nous a permis d'ajouter des stations clients pour donner la main à plusieurs opérateurs du suivi du procédé.
- 🚧 D'un autre part, on peut généraliser notre approche à tous les ateliers de la SCIMAT (broyage cru et broyage à ciment).

- ✚ Ensuite, faire une étude des réseaux récurrents : comparaison de l'utilisation d'une architecture neuro-floue de type Elman et celle de type Jordan.
- ✚ Enfin, on peut intégrer ce système au niveau de la cimenterie à cause de l'utilisation de mêmes équipements, mêmes types d'automates programmables utilisés au niveau de la cimenterie d'Ain-Touta.
- ✚ Interfaçage avec les outils industriels de gestion de maintenance (GMAO).
- ✚ Suivi d'une démarche pour la conception : UML.

Bibliographie

1. **GUENOUNOU, Ouahib.** *Méthodologie de conception de contrôleurs intelligents.* Université Toulouse III - Paul Sabatier : Thèse de Doctorat, 2009.
2. **Azeddine.Chaiba.** *Commande de la machine asynchrone à double alimentation par des techniques de l'intelligence artificielle.* Université de Batna : Thèse de Doctorat, 2010.
3. **Kempowsky.Tatiana.** *Surveillance de procédés à bases de méthodes de classification : conception d'un outil d'aide pour la détection et le diagnostic des défaillances.* institut national des sciences appliqués- Toulouse : Thèse de doctorat, 2004.
4. **Koujok.Mohamed.** *Contribution au pronostic industriel : intégration de la confiance à un modèle prédictif.* Université Franche-Comité : Thèse de Doctorat, 2010.
5. **Marc.Parizeau.** *Le perceptron multicouches et son algorithme de rétropropagation des erreurs.* Université LAVAL , 2004.
6. **Racoceanu.Daniel.** *Contribution à la surveillance des systèmes de production en utilisant les techniques de l'intelligence artificielle.* Université de franche-comté de Besançon : Thèse DEA, 2006.
7. **Elie.Franck.** *Conception et réalisation d'un système utilisant des réseaux de neurones pour l'identification et la caractérisation, à bord de satellites, de signaux transitoires de type sifflement.* Université d'Orléan : Thèse de Doctorat, 1997.
8. **Mouchaweh.M.S.** *Conception d'un système de diagnostic adaptatif et prédictif basé sur la méthode fuzzy pattern matching pour la surveillance en ligne des systèmes évolutifs application à la supervision et au diagnostic d'une ligne de peinture au trempé.* Université de REIMS Champagne-Ardenne : Thèse de Doctorat, 2002.
9. **Zerhouni.N, Nicolas.P.** *Utilisation des réseaux de neurones temporels pour le pronostic et la surveillance dynamique.* laboratoire d'automatique de Besançon-LAP, 2005.
10. **Zemouri.M.R.** *Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : application à la e-maintenance.* Université de Franche-comte : Thèse de Doctorat, 2003.
11. **Ricardo.H.H.** *Supervision et diagnostic des procédés de production d'eau potable.* Institut national des sciences appliquées de Toulouse : Thèse de Doctorat, 2006.
12. **Zerhouni.N, Ryad.Z.** *Réseaux de neurones récurrents à fonctions de base radiales :RRFR - Application à la surveillance dynamique.* Laboratoire d'Automatique de Besançon, France, 2002.
13. **Soufyane.Chekroun.** *commande neuro-floue sans capteur de vitesse d'une machine asynchrone TRIPHASEE.* Université d'Oran-Algérie : Thèse de Magister, 2009.
14. **Palluat.Nicolas.** *Méthodologie de surveillance dynamique à l'aide des réseaux neuro-flous temporels.* l'URF des sciences et techniques en automatique : Thèse de doctorat, 2006.
15. **Amine.M, Kamal.M.** *Diagnostic et pronostic de défaillances par réseaux bayésiens.* Cedex. France , 2008.
16. **Mahadouï.R.** *Diagnostic industriel par neuro-flou -application à un système de production.* Laboratoire d'Automatique et Productique-Université de Batna.Algérie : Thèse de Magister, 2008.

17. **Mouss.M.D.** *Diagnostic et conduite des systèmes de production par approche à base de connaissances.* Laboratoire d'Automatique et Productique.Université de Batna-Algérie : Thèse de Doctrat, 2005.
18. **Nezar.Malika.** *Diagnostic des associations convertisseurs statiques-machines asynchrones en utilisant les techniques de l'intelligence artificielle.* Laboratoire de recherche LEB.Université de Batna-Algérie : Thèse de Doctorat, 2006.
19. **Larabi.M.S, Rouidi.B.** *Diagnostic neuro-flou-application à une machine asynchrone.* 2005.
20. **Matthieu.Desinde.** *Contribution à la mise au point d'une approche intégrée analyse diagnostique/analyse de risques.* Laboratoire d'Automatique de Grenoble, Université JOSEPH FOURIER – GRENOBLE : Thèse de Doctorat, 2006.
21. **Rayhane.Hassan.** *Surveillance des systèmes de production automatisés : détection et diagnostic.* Institut National Polytechnique de Grenoble : Thèse de Doctorat, 2004.
22. **Vasile.O.E.** *Contribution au pronostic de défaillances par réseau neuro-flou : maîtrise de l'erreur de prédiction.* URF des sciences et techniques de l'université de France- comte : Thèse de Doctorat, 2008.
23. **Claudia.N, Victoria.I.** *Diagnostic par techniques d'apprentissage floues : conception d'une methode de validation et d'optimisation des partitions.* Université de Toulouse : Thèse de Doctorat, 2007.
24. **Yacine.Bourek.** *Détermination des conditions d'ionisation caractérisant le seuil de calquage de l'air par la logique floue.* Universite de Batna-Algérie : Thèse de Magister, 2005.
25. **SIEMENS.** *SIMATIC Programmer avec STEP 7- Manuel 6ES7810-4CA08-8CW1.* Allemagne, 2006.
26. **SCIMAT.** *Fiches descriptives de l'historique de la mise en marche du système Fuzzy-Logic du FOUR.* 1997.
27. **ERCE-SCIMAT.** *Revue technique de l'entreprise.* Algérie, 1998.
28. **Vania.Andronova.** *Utilisation de données météo et des réseaux de neurones pour la prédiction de vitesses de vent.* université technique de Sofia, 2006.
29. **SIEMENS-SIMATIC.** *Système de conduite de processus PCS 7- OS Conduite du process (V7.1)- Instructions de service.* Allemagne, 2009.
30. **SIEMENS.** *SIMATIC Système de conduite de process PCS 7 Configuration PC et autorisation (V7.1)- Manuel d'utilisation.* Allemagne, 2006.
31. **SIEMENS-.** *SIMATIC S7-SCL V5.3 pour S7-300/400- Manuel.* Allemagne : s.n., 2005.
32. **F.L.SMIDTH.** *Fours à voie sèche.* Danemark, 1996.
33. **SCIMAT.** *Taux des pannes des équipements de la chaine technologique.* Batna-Algérie, 2005.
34. **SIEMENS_SIMATIC.** *Système de conduite de processus- PCS 7- SFC pour SIMATIC S7 Manuel de programmation et d'utilisation.* Allemagne, 2009.

Annexe A

PRINCIPAUX TERMES DU PROCESS DE CLINKERISATION

Allumeur à gaz

L'allumeur à gaz sert à l'allumage et à la surveillance du bruleur principal. Il est équipé d'une sécurité de flamme- électrode combinée d'allumage et de détection d'ions incorporée dans le bruleur- qui ferme instantanément la soupape d'arrêt de sureté et l'accès de gaz au bruleur dès que l'allumeur s'éteint ou devient instable. L'allumage de l'allumeur est commandé par un programmeur, et un pressostat veille à ce que l'apport d'air soit satisfaisant avant que ne commence l'allumage.

Bandage

Les bandages sont montés sur la longueur du tube de four tout en tenant compte de la flèche maximale permise entre les trains porteurs.

Bruleur à gaz

Le bruleur à gaz est conçu de manière de permettre un réglage rapide et précis de la flamme en vue de l'obtention d'une transmission de chaleur optimale. Dans la buse de bruleur, le courant de gaz est divisé en un courant principal passant par une ouverture centrale et en un courant. Le combustible –soit que ce soit le poussier de charbon, le gaz ou le fuel- est mélangé avec l'air primaire dans le tuyau de bruleur et est ensuite insufflé dans le four. Le bruleur comprend un nombre de tuyaux concentriques tenus en position l'un par rapport à l'autre par des empennages soudés. La quantité d'air primaire est à déterminer durant la mise au point, et elle est dès lors à régler en cas des changements de la quantité du combustible.

Chauffe au gaz

L'installation de chauffe au gaz comprend un bruleur à gaz avec allumeur à gaz, une unité de contrôle électronique, un poste de chauffe et un poste de détente. Le bruleur est conçu de manière à combiner une grande capacité à un contrôle précis du processus de chauffe. La dispersion de la flamme est réglée au moyen d'un cône incorporé dans la buse du bruleur. Le réglage du débit de gaz se fait manuellement ou par commande automatique d'une vanne située au poste de chauffe.

Chute de Croutage

Si un important fragment de croûtage se détache des briques du four, faisant que le matériel descende alors précipitamment dans le four, il est alors nécessaire, dans certain cas, de ralentir la vitesse du four pour éviter que la zone de cuisson se refroidisse en raison de l'avalanche de matériel. Cette situation est appelée « chute croûtage ».

Galets porteurs

L'élasticité du four et un réchauffement irrégulier du tube provoquent des déformations successives, qui ne permettent pas de conserver le parallélisme, donc un contact uniforme entre bandage et galets porteurs.

Point de Consigne

Régulateur (point réglé dorénavant)

Poste de chauffe

La poste de chauffe avec l'armoire de commande est placée près du bruleur. Le fuel passe par l'électrovanne et l'indicateur de quantité, car la soupape est fermée durant la marche. Elle est raccordée au bruleur. La température de fuel, la pression et le débit sont enregistrés sur le panneau de contrôle. Le voyant 1 s'allume lorsque la pression de fuel au bruleur est suffisante et le voyant 2 s'allume lorsque la pression de vapeur est suffisante. Le voyant 3 s'allume, lorsque le groupe est prêt au démarrage, c.-à-d. lorsque le ventilateur primaire et celui à filtre marche et lorsque les conditions de pression et de température susmentionnées sont présentes. En cas de défaillance des conditions de démarrage, le clapet ferme automatiquement et un autre ouvre à la vapeur et à l'air comprimé pour le nettoyage du bruleur.

Refroidissement du clinker

Le clinker quitte le four avec une température de 1200-1300°C, et il est nécessaire de refroidir, d'une part pour pouvoir le transposer et d'autre part en vue du broyage dans le broyeur à ciment. Donc, le clinker est refroidi dans un courant d'air froid. L'air de refroidissement est chauffé en mesure avec le refroidissement du clinker, et l'air chauffé est par la suite utilisé comme air comburant (air secondaire) à la cuisson.

Revêtement

Le four rotatif est revêtu à l'intérieur de briques réfractaires biseautées et assujetties par coincement. Le revêtement est constamment soumis, à la flexion du four et son ovalisation, en plus de l'usure mécanique, ainsi qu'aux effets thermiques et aux variations de température.

Station de commande

La station de commande est installée près d'un train porteur, là où le four est calé par le bandage. La vitesse de rotation du four est de 2 tr/mn. La rotation lui est transmise par la couronne dentée, le pignon et le réducteur. Le four peut être viré indépendamment du moteur principal par réducteur de virage et moteur auxiliaire.

Annexe B

PROPORTION DES MATIERES PREMIERES

Pour avoir une bonne composition chimique du cru dépendant de la composition chimique de chaque matière première et pour limiter les quantités des impuretés on fait une optimisation de la composition chimique à l'aide de 3 équations.

Équation 1: Taux de saturation en chaux

Cette équation détermine la quantité de chaux nécessaire pour saturer les trois autres oxydes (SiO₂, Al₂O₃, Fe₂O₃).

LSF : Lime Saturation Factor (taux de saturation en chaux)

$$LSF = \frac{CaO \times 100\%}{28SiO_2 + 1,18Al_2O_3 + 0,65Fe_2O_3}$$

Équation 2: Module silicique

La deuxième équation nécessaire est celle qui détermine la relation entre la quantité de SiO₂ d'une part et la quantité d'Al₂O₃ et Fe₂O₃ d'autre part.

MS : Module Silicique.

$$MS = \frac{SiO_2}{Al_2O_3 + Fe_2O_3} = \frac{S}{AF} = 1,932$$

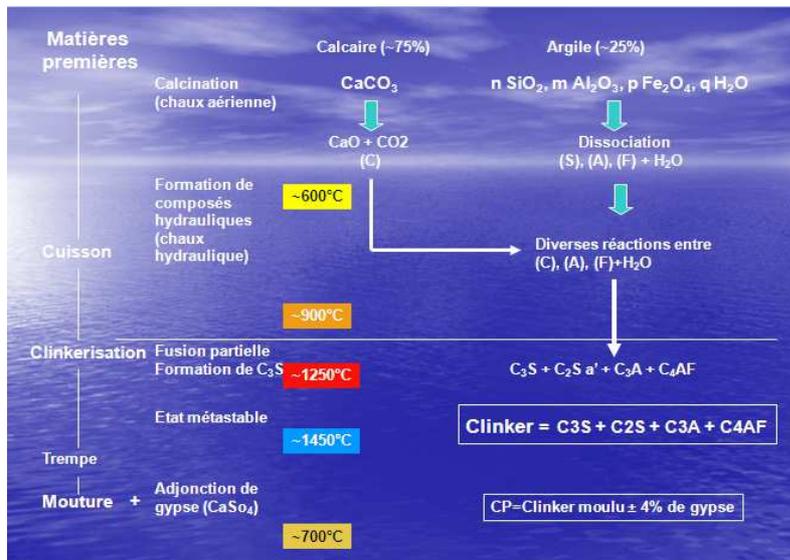
Équation 3: Module Aluminoferrique

La troisième équation est celle qui détermine la relation entre la quantité d'Al₂O₃ et de Fe₂O₃.

MA : Module Aluminoferrique.

$$MA = \frac{Al_2O_3}{Fe_2O_3} = \frac{A}{F} = 1,525$$

Paramètre	Valeur	Actions
LSF	>99	Cuisson difficile, chaux libre élevée, consommation énergétique excessive
	<90	Cru facile à cuire, faible consommation calorifique, phase liquide élevée, tendance à la formation d'anneaux dans le four, chute des résistances mécaniques à 28 jours
MS [2.3] Module Silicique	<2	Cuisson facile, phase liquide en excès, briques attaquées, revêtement lavé, clinker dur, CCS basse
	>3	Cuisson difficile, phase liquide faible, charge thermique élevée, pas de croustage, KK poussiéreux, CaO libre élevée, four instable
MAF [1.3-2.5] Module Aluminoferrique	<2	Phase liquide visqueuse, résistances initiales élevées
	>3	Phase liquide fluide, résistances initiales faibles



COMPOSITION MINÉRALOGIQUE DU CLINKER(%)

C3S	57,44
C2S	17,36
C3A	09,52
C4AF	09,75

CARACTÉRISTIQUES CHIMIQUES (%)

CaO	60,41
Al_2O_3	05,19
SiO_2	21,91
Fe_2O_3	02,94
MgO	01,60
Na_2O	00,16
K_2O	00,54
Cl^-	00,02
SO_3	02,19
P.A.F	03,83
Résidus Insolubles	03,57
CaO libre	01,01

Annexe C

LES ALGORITHMES D'APPRENTISSAGE DU SYSTEME NEURO-FLOU

Dans cette annexe nous présentons quatre algorithmes d'apprentissages en utilisant les notions suivantes:

- Γ : un ensemble de données d'apprentissage avec $|\Gamma| = s$, qui représente un problème de classification de défaillances, tel que la forme $p \in \mathfrak{R}^n$ qui doivent être classées dans m mode de défaillances, tel que $C_i \subseteq \mathfrak{R}^n$.
- $(p, t) \in \Gamma$: une forme d'apprentissage consiste d'un vecteur d'entrée $p \in \mathfrak{R}^n$ et un vecteur de sortie (désiré) $t \in [0,1]^m$. L'index de la classe de \mathbf{P} est calculé par le grand valeur de \mathbf{t} : classe (p) = $\text{argmax}_j \{t_j\}$.
- $R = (A, C)$: une règle floue de classification avec un antécédent $\text{ant}(R) = A$ et conséquent $\text{con}(R) = C$, avec $A = (\mu_{j1}^{(1)}, \dots, \mu_{jn}^{(n)})$ et C est un classe (mode de défaillance). Le degré d'accomplissement d'une règle R pour une forme p est

$$R(p) = A(p) = \min\{\mu_{j1}^{(1)}(p_1), \dots, \mu_{jn}^{(n)}(p_n)\}.$$

- $\mu_j^{(i)}$ est la j^{eme} ensemble floue de partition floue initial du variable d'entée x_i .

Il existe q_i ensemble flou pour le variable x_i .

- \mathbf{c}_A : un vecteur de m entrées pour présenter les degrés d'appartenance cumulées pour chaque classe de toutes les formes avec $A(p) > 0$; $\mathbf{c}_A[j]$ est le j^{eme} entrée de \mathbf{c}_A .
- $P_R \in [-1,1]$ une valeur qui représente la performance de règle R :

$$P_R = \frac{1}{s} \sum_{(p,t)} (-1)^c R(p), \text{ avec } c = \begin{cases} 0 & \text{si } \text{class}(p) = \text{con}(R), \\ 1 & \text{si autrement} \end{cases}$$

- $\nu_r^{(j)}$: l'ensemble floue du variable de sortie y_j ($j \in \{1, \dots, m\}$) qui est incluse dans le conséquent de règle floue R_r .
- C_j : le vecteur de sortie qui enregistre le degré d'appartenance de forme d'entrée pour la classe j .

Algorithme 1. Algorithme d'apprentissage de règles**Début algorithme****Pour** toutes forme (p,t) **faire****Pour** toutes entrée x_i **faire**

$$\mu_{ji}^{(i)} = \arg \max_{\mu_j^{(i)}, j \in \{1, \dots, q_i\}} \{\mu_j^{(i)}(p_i)\}$$

Fin pourCréer l'antécédent $A = (\mu_{j1}^{(1)}, \dots, \mu_{jn}^{(n)})$;**Si** ($A \notin$ liste des antécédents) **alors**

Ajouter A dans la liste des antécédents ;

Finsi**Finpour****Pour** toutes forme (p,t) **faire****Pour toutes** $A \in$ liste d'antécédents **faire**

$$c_A[\text{classe}(p)] = c_A[\text{classe}(p)] + A(p) ;$$

Finpour**Finpour****Pour toutes** $A \in$ liste d'antécédents **faire**

$$j = \arg \max_{i \in \{1, \dots, m\}} \{c_A[i]\} ;$$

Créer une règle R avec l'antécédent A et conséquent C_j ;

Ajouter R dans la base de règle candidates ;

$$P_R = \frac{1}{s} c_A[j] - \sum_{i \in \{1, \dots, m\}, i \neq j} c_A[i] ;$$

Finpour

Raffiner la base de règles candidates

Fin algorithme**Algorithme 2. Sélectionner les meilleures règles pour la base de règles****Début Algorithme** $K = 0$; stop = false ;**Répéter** $R' = \arg \max \{P_R\}$;**Si** taille base de règles fixées **alors****Si** $k < k_{max}$ **alors**

Ajouter R' dans la base de règle ;

Effacer R' de liste des règles candidates ;

 $K := k + 1$;**Sinon**

Stop = true ;

Finsi**Fin algorithme**

Algorithme 3. Apprentissage des ensembles flous**Début d'algorithme**

1- sélectionner la forme suivante (p,t) de Γ , propager la, par NEFDIAG et détermine le vecteur de sortie C .

2- pour chaque unité de sortie C_i ; déterminer la valeur de delta

$$\delta C_i = t_i - O C_i$$

3- pour chaque unité de règle R avec $O_R > 0$

a- déterminer la valeur de δ

$$\delta_R = O_R (1 - O_R) \sum_{C \in U_3} W(R, C) \delta_C \cdot$$

b- trouver x' tel que : $w(x', R)(o_{x'}) = \min_{x \in U_1} \{w(x, R)(\delta_C)\}$

c -pour l'ensemble flou $w(x', R)$ déterminer les valeurs de δ pour les paramètres a, b, c utilisant le pas d'apprentissage $\sigma > 0$.

$$\delta_b = \sigma \cdot \delta_R \cdot (c - a) \operatorname{sgn}(O_{x'} - b),$$

$$\delta a = -\sigma \delta_R \cdot (c - a) + \delta_b,$$

$$\delta c = \sigma \delta_R \cdot (c - a) + \delta_b.$$

Appliqué le changement aux $w(x', R)$.

4- Si l'itération est terminée, ou les critères de fin sont validés ; alors stop

Sinon aller à (1).

Fin d'algorithme**Algorithme 4. Mises à jour des ensembles flous de conséquents**

{ Les paramètres d'entrées suivantes de cet algorithme sont

V : L'ensemble flou qui doit calculer ses paramètres.

e : Valeur d'erreur.

τ : Degré d'accomplissement pour la règle qui utilise V dans ça conséquent.

t : Valeur de vecteur de sortie désiré pour le domaine de V

a, b, c, d Sont les paramètres de V }

Début Algorithme

Si V est triangulaire **alors**

$$\text{Shift} = \sigma \cdot \tau \cdot (c - a) \cdot \tau \cdot (1 - v(t)) ;$$

$$\Delta b = \Delta b \text{ shift} ;$$

Si $v(t) > 0$ **alors**

$$\Delta a = \Delta a + \sigma \cdot \tau \cdot (b - a) + \text{shift}$$

$$\Delta c = \Delta c - \sigma \cdot \tau \cdot (b - a) + \text{shift}$$

Sinon

$$\Delta a = \Delta a + \operatorname{sgn}(t - b) \sigma \cdot \tau \cdot (b - a) + \text{shift}$$

$$\Delta c = \Delta c + \operatorname{sgn}(\sigma \cdot \tau \cdot (b - a)) + \text{shift}$$

Finsi

Fin Algorithme

Annexe D

SYSTEME SDR/FUZZY LOGIC

F.L. SMIDTH a mis au point un système de contrôle de four basé sur un mini-ordinateur. Système qui simule directement le mode d'évaluation et de décision d'un opérateur expérimenté. À cet effet il est utilisé un outil mathématique, dénommé « FUZZY LOGIC ».

➤ Paramètres de Réglage du Four

Le contrôle d'un four présente certains problèmes en raison de la complexité du processus, des temps de réponse ainsi que des possibilités de mesure limitées. Le degré d'automatisation du contrôle d'un four a été relativement faible, et c'est pourquoi la marche satisfaisante du four dépendait en grande partie de l'expérience et de la compétence des opérateurs. Les paramètres du procédé d'un four contrôlés par le FUZZY sont :

- ✓ L'alimentation du four,
- ✓ La vitesse du four (rotation par minute),
- ✓ Le débit du gaz du four (combustion),
- ✓ Vitesse du ventilateur de tirage principal.

Chacun de ces réglages a une influence sur les différentes phases de préchauffage, calcination, clinkérisation et refroidissement, cela semble défier toute loi d'équilibre que de vouloir maintenir une marche stable du four à une production maximale de clinker, une consommation minimale de combustible et à une charge admissible du matériel.

Ainsi, ces paramètres sont contrôlés à partir d'indications de procédé, tels que couple d'entraînement du four, oxygène et température dans le préchauffeur. De plus les opérateurs devront introduire les valeurs d'échantillons de chaux libre et poids litre dans le système FUZZY/QCX, de façon à permettre au FUZZY de contrôler la qualité du clinker.

➤ Contrôle de la Marche du Four

Le système FUZZY établit en ligne la surveillance et le contrôle des principaux paramètres de conduite, par quoi il se substitue à l'opérateur en assurant ses fonctions de coordination. En conséquence, le système élimine les différences entre les méthodes de contrôle de chaque opérateur : il en résulte une marche plus uniforme au cours des 24 heures.

D'après les documents historiques du système FUZZY, les conditions de marche/arrêt du FUZZY en date : 30/10/1996 sont les suivantes :

- **Conditions d'enclenchement du FUZZY :**
 - ✓ Si le débit de gaz V01 $\in [4500, 6500]$ Nm³/h,
 - ✓ Si la température après cyclones $\in [315, 380]$ °C,
 - ✓ Si l'oxygène après cyclones $\in [3.20\%, 5\%]$,
 - ✓ Si la consommation calorifique spécifique $\in [900, 1000]$ Kcal (kg clinker),
 - ✓ Si vitesse de rotation du four = 2t/mn,
 - ✓ Si la vitesse de ventilateur de tirage $\in [68\%, 72\%]$,
 - ✓ Si le débit de gaz bruleur secondaire réduit entre [850, 900] Nm³/h.

- **Conditions de déclenchement du FUZZY :**

- ✓ Si la chute de croûtage engendrant l'augmentation de l'ampérage du transporteur à auget plus de 80%,
- ✓ Si la chaux libre (FCAO) > 03 % durant 3 heures successives,
- ✓ Si la chaux libre (FCAO) < 0.6 % durant 3 heures successives,
- ✓ Si le poids au litre (LTW) > 1400 g/l durant 3 heures successives,
- ✓ Si le poids au litre (LTW) < 1050 g/l durant 3 heures successives,
- ✓ Si le débit du gaz engendre des fluctuations importantes.

- **Programme de Conduite du Four par "FUZZY LOGIC"**

La conduite du four par le système de "FUZZY" est commandée par le mini-ordinateur. Les règles de commande sont enregistrées via une console de visualisation. Et constituent, dans leur ensemble, une stratégie de conduite qui décrit ce qu'il faut faire dans chaque situation. Les règles de conduite du four sont écrites en langage clair, en utilisant des expressions courantes comme « haut, bas, normal, beaucoup, peu ».

Un programme traite les données et les transforme pour appliquer les réglages demandés. L'ordinateur enregistre automatiquement, au moyen de mesures, toutes les Informations. En appliquant la technique de logique floue, le programme décide ensuite, sous une forme entièrement numérique, dans quelle mesure l'état de marche du moment correspond effectivement aux conditions d'application de chaque règle de conduite et calcule le meilleur compromis entre les différentes propositions de corrections de la marche du four. Ensuite, l'ordinateur apportera automatiquement les corrections de conduite ou présentera, sur un écran de visualisation, le résultat de ses calculs pour contrôle et approbation de l'opérateur.

Ce procédé est typiquement utilisé au cours de la période de mise en marche du four durant laquelle l'opérateur suit attentivement le déroulement de la conduite automatique, et en ajoutant, retirant ou modifiant les règles de conduite au moyen du pupitre de commande, il « apprend » à l'ordinateur la conduite correcte du four. Lorsque l'opérateur est convaincu que la conduite du four se fait de manière adéquate, il peut le laisser fonctionner en automatique.