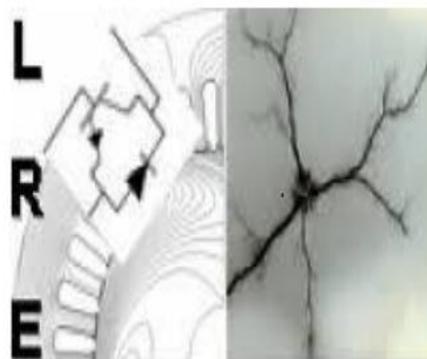


REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique



Département Électrotechnique

Laboratoire de Recherche en Électrotechnique

Mémoire de Projet de Fin d'Etude en vue de l'obtention du diplôme  
d'Ingénieur d'état en Électrotechnique

# Contrôleurs Neuro-flous et Hybrides dans l'AGC d'un Réseau Electrique

Présenté par

**Ramzi KECIR**

Sous la direction de : Pr. Abdelhafid HELLAL

Présenté et soutenu le : 22/06/2023

## Composition du jury :

Président	Pr. R. TAHMI	ENP
Examineur	Dr. R. BELKACEMI	ENP
Rapporteur	Pr. A. HELLAL	ENP

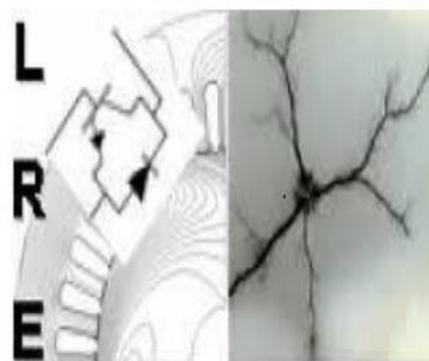
ENP 2023



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique



Département Électrotechnique  
Laboratoire de Recherche en Électrotechnique

Mémoire de Projet de Fin d'Etude en vue d'obtention du diplôme  
d'Ingénieur d'état en Électrotechnique

# Contrôleurs Neuro-flous et Hybrides dans l'AGC d'un Réseau Electrique

Présenté par

**Ramzi KECIR**

Sous la direction de : Pr. Abdelhafid HELLAL

Présenté et soutenu le : 22/06/2023

## Composition du jury :

Président	Pr. R. TAHMI	ENP
Examineur	Dr. R. BELKACEMI	ENP
Rapporteur	Pr. A. HELLAL	ENP

ENP 2023

## ملخص :

تقدم هذه المذكرة كيفية تصميم وتنفيذ وحدات تحكم ذكية باستخدام تقنية الأنظمة العصبية الضبابية التي تعد توليفة هجينة بين الشبكات العصبية والمنطق الضبابي في إطار التحكم التلقائي في التوليد لتحسين استقرار التردد. يعتبر استقرار التردد مسألة هامة جدًا وتقدم وحدات التحكم استقرار تردد أفضل من وحدات التحكم النسبية أو الضبابية في العديد من الحالات المدروسة في هذا العمل. تم إجراء المحاكاة باستخدام برنامج "ماتلاب" الذي يوفر أدوات قوية للعمل والتحليل. يتم شرح استخدام هذه الأدوات والتعديلات اللازمة على بيانات تدريب نموذج النظام العصبي الضبابي بالتفصيل في هذا العمل.

**الكلمات المفتاحية:** التحكم التلقائي في التوليد، استقرار التردد، المنطق الضبابي، وحدات التحكم النسبية، وحدات التحكم الضبابية، وحدات التحكم العصبية الضبابية، وحدات التحكم انفيس، بيانات التدريب.

## Abstract :

*This thesis presents the design and implementation of the intelligent neuro-fuzzy controllers (ANFIS), which are a hybrid of neural networks and fuzzy logic, for automatic generation control (AGC) to improve the frequency's stability (LFC) of a power grid. Frequency stability being crucial, the introduction of ANFIS controllers instead of regular PID or fuzzy controllers can give better performances in various scenarios explored in this study. The simulations were conducted using MATLAB/SIMULINK software, which provides powerful analysis tools. These tools handling and the necessary adjustments to the ANFIS model's training data are described in detail in this work.*

**Keywords:** Automatic generation control (AGC), Load frequency control (LFC), PID controller, Fuzzy logic, Fuzzy controller, neuro-fuzzy controller, ANFIS controller, Training data.

## Résumé :

*Ce mémoire présente la conception et l'implémentation des régulateurs intelligents neuro-flous (ANFIS), hybride entre les réseaux de neurones et la logique floue, dans le cadre du contrôle automatique des groupes de production (AGC) afin d'améliorer la stabilité de fréquence (LFC) du réseau électrique. La stabilité de fréquence étant une problématique très importante, l'introduction des régulateurs ANFIS à la place des régulateurs PID ou flous offre des résultats plus performants dans plusieurs cas de figures vus dans cette étude. Les simulations ont été réalisées sur le logiciel MATLAB/SIMULINK qui offre des outils de travail performants. La manipulation de ces outils et les corrections à apporter aux données d'entraînement du modèle ANFIS sont détaillées dans ce travail.*

**Mots clés :** Contrôle automatique de génération (AGC), Contrôle de fréquence (LFC), Régulateur PID, Logique floue, Régulateur flou, Régulateur neuro-flou, régulateur ANFIS, Données d'entraînement.

## ***Remerciements :***

*Mes remerciements vont en premier lieu à Allah le tout puissant qui m'a offert cette vie et toutes ces opportunités qui m'ont permis d'arriver à ce stade de ma vie.*

*Je remercie ensuite le Professeur Abdelhafid HELLAL qui a accepté de m'encadrer pour ce projet de fin d'étude. Je le remercie de m'avoir encouragé à travailler sérieusement, de m'avoir offert les outils nécessaires et de m'avoir soutenu avec rigueur.*

*Je tiens aussi à remercier tous mes professeurs qui pendant la spécialité ou les classes préparatoires m'ont enseigné l'amour de l'ingénierie, du travail et du sérieux.*

## ***Dédicaces :***

*Je dédie ce travail à toute ma famille et à tous mes amis, en particulier à :*

***Mes grands-parents Ahmed KECIR et Addra CHETTIBI***

*Qui représentent une grande partie de ma vie et auxquels je suis infiniment reconnaissant.*

*PS : je suis désolé si j'ai fait quelques fautes d'orthographes papi*

***Mon père Mohamed Riadh***

*Qui a toujours été présent. J'espère devenir un jour ne serait-ce qu'une fraction de l'homme que tu es.*

***Ma mère Aicha Naziha***

*Qui m'a permis de devenir l'homme que je suis.*

***Et à mes petites sœurs Rania et Rihem***

*Je vous souhaite de réussir encore plus que moi dans ce monde et dans l'au-delà.*

# Table des matières :

Liste des figures

Liste des tableaux

Acronymes

Introduction générale.....14

## Chapitre 1 : Modélisation du réseau électrique et régulation par PID

1. Modélisation du réseau électrique.....	18
1.1.Modélisation du générateur synchrone (GS).....	18
1.2.Modélisation de la charge.....	19
1.3.Modélisation de la turbine.....	20
1.4.Modélisation du régulateur de vitesse (Gouvernail).....	20
2. Régulation par PID.....	22
2.1.Cas d'une centrale isolée (région autonome).....	22
2.2.Cas de deux zones interconnectées.....	27
2.2.1. Modélisation de l'interconnexion.....	28
2.2.2. Réponse dynamique du système à deux zones.....	29
2.3.Cas de trois zones interconnectées.....	31
2.4.Conclusion.....	33

## Chapitre 2 : Contrôleur flou appliqué à l'AGC

1. Introduction à la logique floue.....	35
2. Application de la logique floue dans l'AGC.....	36
2.1.Conception d'un régulateur flou.....	36
2.2.Implémentation du régulateur flou.....	40
3. Conclusion.....	45

## Chapitre 3 : Contrôleur neuro-flou appliqué à l'AGC

1. Introduction aux systèmes neuro-flous.....	47
2. Théorie derrière la conception d'un ANFIS.....	47
2.1.Modèle de l'ANFIS.....	47
2.2.Entraînement du modèle ANFIS.....	50
3. Conception d'un régulateur ANFIS.....	50
4. Application du ANFIS sur 1 zone.....	54
5. Application du ANFIS sur 2 zones.....	59
5.1.Un régulateur PID (zone 1) et un régulateur ANFIS (zone 2).....	59
5.2.Deux régulateur ANFIS sur la zone 1 et la zone 2.....	65
5.3.Comparaison des méthodes utilisées sur 2 zones.....	69
5.4.Entraînement d'un modèle ANFIS à partir des données d'un régulateur PID.....	71
6. Application du ANFIS sur 3 zones.....	74
7. Conclusion.....	76

## **Chapitre 4 : Création d'une interface graphique**

1. Appdesigner.....78
2. Présentation de l'application.....79
3. Conclusion.....83

**Conclusion générale.....85**

**Bibliographie.....86**

## Liste des figures :

<b>Figure 1.1</b> : Modèle du Générateur Synchrones.....	18
<b>Figure 1.2</b> : Modèle du générateur avec charge.....	19
<b>Figure 1.3</b> : Schéma réduit du générateur avec charge.....	19
<b>Figure 1.4</b> : Schéma de la turbine.....	20
<b>Figure 1.5</b> : Caractéristique de vitesse du régulateur.....	21
<b>Figure 1.6</b> : Schéma du régulateur de vitesse. ....	21
<b>Figure 1.7</b> : Modèle d'un réseau primaire. ....	22
<b>Figure 1.8</b> : Modèle d'un réseau primaire pour la centrale de Tilghemt.....	23
<b>Figure 1.9</b> : Réponse du modèle suite à un changement de charge.....	23
<b>Figure 1.10</b> : Modèle d'un réseau avec réglage secondaire.....	24
<b>Figure 1.11</b> : Modèle d'un réseau avec réglage secondaire pour la centrale de Tilghemt.....	24
<b>Figure 1.12</b> : Effet du proportionnel sur l'erreur de fréquence.....	25
<b>Figure 1.13</b> : Effet de l'intégrale sur l'erreur de fréquence.....	26
<b>Figure 1.14</b> : Effet du dérivatif sur l'erreur de fréquence.....	26
<b>Figure 1.15</b> : Représentation simplifiée de deux zones interconnectées.....	27
<b>Figure 1.16</b> : Schéma de l'interconnexion interzones.....	28
<b>Figure 1.17</b> : Modèle de deux zones interconnectées avec réglage primaire.....	28
<b>Figure 1.18</b> : Modèle Simulink de deux zones interconnectées avec réglage primaire.....	29
<b>Figure 1.19</b> : Réponse du modèle de simulation à 2 zones avec réglage primaire.....	29
<b>Figure 1.20</b> : Modèle Simulink de deux zones interconnectées avec réglage secondaire.....	30
<b>Figure 1.21</b> : Réponse du modèle de simulation à 2 zones avec réglage secondaire.....	30
<b>Figure 1.22</b> : Représentation simplifiée de 3 zones interconnectées.....	31
<b>Figure 1.23</b> : Modèle Simulink de trois zones interconnectées avec réglage secondaire.....	32
<b>Figure 1.24</b> : Réponse du modèle de simulation à 3 zones avec réglage secondaire.....	32
<b>Figure 2.1</b> : Fonctions d'appartenance floues pour une variable de chaleur.....	35
<b>Figure 2.2</b> : Structure d'un régulateur flou classique.....	36
<b>Figure 2.3</b> : Fonctions d'appartenance de $\Delta f$ sur MATLAB.....	37
<b>Figure 2.4</b> : Fonctions d'appartenance de $\Delta \dot{f}$ sur MATLAB.....	37
<b>Figure 2.5</b> : Fonctions d'appartenance de $\Delta P_{ref}$ sur MATLAB.....	38

<b>Figure 2.6 :</b> Règles d'inférence du régulateur flou sur MATLAB.....	39
<b>Figure 2.7 :</b> Schéma Simulink d'un régulateur flou.....	40
<b>Figure 2.8 :</b> Schéma du calcul de la dérivée de l'erreur de fréquence.....	40
<b>Figure 2.9 :</b> Schéma du calcul de l'action intégrale de la puissance de référence.....	41
<b>Figure 2.10 :</b> Schéma Simulink pour 1 zone avec régulateur flou.....	42
<b>Figure 2.11 :</b> Comparaison entre la réponse du système pour un régulateur PID et un flou...42	42
<b>Figure 2.12 :</b> Fonctions d'appartenance de $\Delta f$ du régulateur flou2.....	42
<b>Figure 2.13 :</b> Fonctions d'appartenance de $\Delta \dot{f}$ du régulateur flou2.....	43
<b>Figure 2.14 :</b> Fonctions d'appartenance de $\Delta P_{ref}$ du régulateur flou2.....	43
<b>Figure 2.15 :</b> Réponse du système pour le régulateur flou2.....	44
<b>Figure 2.16 :</b> Réponse du système pour le régulateur flou2 pour dP=1.5%.....	44
<b>Figure 3.1 :</b> Représentation du réseau de neurone de l'ANFIS.....	48
<b>Figure 3.2 :</b> Raisonnement flou avec méthode du minimum.....	49
<b>Figure 3.3 :</b> Fenêtre de l'outil « Neuro-Fuzzy Designer ».....	50
<b>Figure 3.4 :</b> Chargement des données d'entraînement.....	51
<b>Figure 3.5 :</b> Dix premiers points d'une matrice de donnée.....	51
<b>Figure 3.6 :</b> Schéma d'entraînement et de simulation.....	52
<b>Figure 3.7 :</b> Choix relatifs aux fonctions d'appartenance.....	52
<b>Figure 3.8 :</b> Phase d'entraînement du modèle ANFIS.....	53
<b>Figure 3.9 :</b> Phase de test du modèle ANFIS.....	54
<b>Figure 3.10 :</b> Fenêtre principale du régulateur ANFIS.....	55
<b>Figure 3.11 :</b> Fonctions d'appartenance de $\Delta f$ du régulateur ANFIS.....	56
<b>Figure 3.12 :</b> Fonctions d'appartenance de $\Delta \dot{f}$ du régulateur ANFIS.....	56
<b>Figure 3.13 :</b> Fonctions conséquentes de la variable de sortie $\Delta P_{ref}$ du régulateur ANFIS...57	57
<b>Figure 3.14 :</b> Règles d'inférence du régulateur ANFIS.....	57
<b>Figure 3.15 :</b> Comparaison des réponses du système pour un régulateur flou et un régulateur ANFIS.....	58
<b>Figure 3.16 :</b> Réponse du système pour les différents régulateurs.....	58
<b>Figure 3.17 :</b> Schéma à deux zones avec régulation ANFIS dans la zone non perturbée.....	59
<b>Figure 3.18 :</b> Réponses du système pour une perturbation sur la zone 1.....	60
<b>Figure 3.19 :</b> Schéma à deux zones avec les deux zones perturbées.....	61

<b>Figure 3.20</b> : Réponse du système pour les régulateurs flous sur la zone 2.....	61
<b>Figure 3.21</b> : Réponse du système pour le régulateur flou2 dans la zone 2.....	62
<b>Figure 3.22</b> : Données d'entraînement du régulateur ANFIS.....	63
<b>Figure 3.23</b> : Résultats des tests du régulateur ANFIS.....	63
<b>Figure 3.24</b> : Réponse du système pour le régulateur ANFIS dans la zone 2.....	64
<b>Figure 3.25</b> : Réponse du système pour le régulateur ANFIS dans la zone 2 en échangeant dP1 avec dP2.....	64
<b>Figure 3.26</b> : Schéma à deux zones avec deux régulateurs ANFIS.....	65
<b>Figure 3.27</b> : Réponse du système pour 2 régulateurs ANFIS identiques.....	65
<b>Figure 3.28</b> : Réponse du système pour 2 régulateurs ANFIS identiques avec dP1=dP2.....	66
<b>Figure 3.29</b> : Réponse du système pour 1 régulateur ANFIS en mono-zone.....	67
<b>Figure 3.30</b> : Réponse du système pour le régulateur flou3 en mono-zone.....	67
<b>Figure 3.31</b> : Réponse du système pour le régulateur ANFIS tiré du flou3 en mono-zone....	68
<b>Figure 3.32</b> : Réponse du système pour 2 régulateurs ANFIS différents.....	69
<b>Figure 3.33</b> : Comparaison entre 2 PID et 2 ANFIS.....	69
<b>Figure 3.34</b> : Comparaison entre 1 PID + 1 ANFIS et 2 ANFIS.....	70
<b>Figure 3.35</b> : Comparaison entre 2 régulateurs flous et 2 ANFIS.....	70
<b>Figure 3.36</b> : Comparaison entre 1 PID optimisé + 1 ANFIS et 2 ANFIS.....	71
<b>Figure 3.37</b> : Schéma d'apprentissage et de simulation d'un ANFIS entraîné à partir d'un PID.....	72
<b>Figure 3.38</b> : Comparaison entre le PID dans la zone 1 et l'ANFIS qu'il a entraîné.....	72
<b>Figure 3.39</b> : Puissance de référence dans le cas du PID et de l'ANFIS.....	73
<b>Figure 3.40</b> : Schéma à 3 zones avec régulation ANFIS.....	74
<b>Figure 3.41</b> : Réponse du système à 3 zones pour différents régulateurs.....	75
<b>Figure 4.1</b> : Fenêtre de MATLAB GUI.....	78
<b>Figure 4.2</b> : Fenêtre du codage du MATLAB GUI.....	78
<b>Figure 4.3</b> : Fenêtre d'accueil de l'application.....	79
<b>Figure 4.4</b> : Fenêtre du cas mono-zone.....	79
<b>Figure 4.5</b> : Réponse du système mono-zone au régulateur ANFIS.....	80
<b>Figure 4.6</b> : Comparaison avec le régulateur flou pour le cas mono-zone.....	80
<b>Figure 4.7</b> : Comparaison entre les différents régulateurs pour le cas mono-zone.....	81

<b>Figure 4.8 :</b> Fenêtre du cas 2 zones.....	81
<b>Figure 4.9 :</b> Comparaison entre les différents régulateurs pour le cas 2 zones.....	82
<b>Figure 4.10 :</b> Fenêtre du cas 3 zones.....	82
<b>Figure 4.11 :</b> Comparaison entre les différents régulateurs pour le cas 3 zones.....	83

## **Liste des tableaux :**

<b>Tableau 2.1 :</b> Tables des règles du contrôleur flou.....	39
--	----

## Liste des acronymes :

$f$	fréquence
LFC	Load Frequency Control
AGC	Automatic Generation Control
IA	Intelligence Artificielle
PID	Proportionnel, Intégral, Dérivatif
$\Omega_m$	vitesse angulaire du rotor
$J$	moment d'inertie
$C_m$	couple mécanique
$C_r$	couple résistant
GS	générateur synchrone
$C_a$	couple d'accélération
$H$	constante d'inertie
$P_{méc}$	puissance mécanique
$P_{élec}$	puissance électrique
$P_L$	puissance de charge indépendante de la fréquence
$K_p$	gain de système électrique
$T_p$	constante de temps du système électrique
$T_1, T_2$	constante de temps de la turbine
$R$	constante de régulation de vitesse
$W$	énergie cinétique
Kp	gain du proportionnel
Ki	gain de l'intégral
Kd	gain du dérivatif
ACE	area control error
p.u	per unit
FIS	Fuzzy Inference System
FLC	Fuzzy Logic Controller
ANFIS	Adaptive Network-based Inference System/Adaptive Neuro-Fuzzy Inference System
GUI	Graphic User Interface

# **Introduction Générale**

## **Introduction :**

Les réseaux électriques font partie des infrastructures stratégiques de notre société moderne, fournissant l'énergie et l'électricité nécessaires à des millions de foyers, lieux publics et au secteur industriel.

Les réseaux électriques actuels sont complexes et vastes, s'étendant sur la majorité du territoire national pour répondre à une demande d'énergie en changement et croissance constants. Ceci a pour effet de rendre les réseaux vulnérables à différentes formes de perturbations électriques, telle la perturbation de tension ou de fréquence qui menacent la stabilité du réseau.

La perturbation de fréquence est directement liée au déséquilibre entre la production électrique et la consommation à un moment donné. Plus l'écart entre les deux est important et plus la fréquence s'éloigne de sa valeur nominale qui est de 50Hz en Algérie.

Passé une faible tolérance d'erreur (de l'ordre de 0.5Hz), la variation de fréquence produit plusieurs conséquences néfastes sur le réseau électrique et chez les consommateurs. Les centrales électriques sont constituées de turbines et d'alternateurs qui tournent à fréquence constante. Une chute de fréquence peut entraîner une désynchronisation des générateurs électriques entraînant un déséquilibre de la production d'électricité pouvant aboutir à des pannes de courants, voir un blackout dans les cas extrêmes.

La consommation d'énergie varie constamment à chaque instant rendant la régulation manuelle de la quantité de production très difficile et peu précise. Des prédictions de la variation de la consommation sont calculées au niveau des centres de dispatching, ce qui permet de connaître la production nécessaire à fournir pour chaque centrale à travers la journée. Néanmoins, les prédictions ne sont jamais parfaites et elles ne sont pas assez précises pour exercer une régulation instantanée de la variation de fréquence. C'est pour cela qu'il faut introduire des moyens de régulations autonomes au niveau des différentes centrales électriques pour ajuster les niveaux de production des générateurs, afin de régler la fréquence.

Le réglage de la fréquence de charge (Load Frequency Control-LFC) consiste comme son nom l'indique à régler la fréquence de charge. Les objectifs principaux de ce réglage sont :

- Assurer la sécurité du réseau électrique en préservant la fréquence à sa valeur nominale ou à des valeurs proches.
- Maintenir l'équilibre de puissance entre les zones interconnectées.
- Répartir la production de chaque centrale de manière économiquement optimale.

Une stratégie de commande est utilisée qui consiste à contrôler automatiquement les groupes de production (Automatic Generation Control-AGC). L'AGC ajuste les niveaux de production des générateurs pour les équilibrer avec la demande de la nouvelle charge.

Le réseau est divisé en plusieurs zones de contrôle qui sont interconnectées entre elles. Chaque zone de contrôle contient un nombre de générateurs qui subissent le même contrôle car ils doivent maintenir une variation de vitesse similaire entre eux.

L'AGC se divise sous 3 formes de réglages [1] :

- **Le réglage primaire** : se déroule pendant les premières secondes suivant la perturbation. La fréquence se stabilise à une valeur légèrement inférieure de sa valeur nominale.
- **Le réglage secondaire** : se déclenche après le réglage primaire et règle la fréquence à sa valeur nominale grâce à des régulateurs autonomes (PID, flou...).
- **Le réglage tertiaire** : a lieu en dernier et règle la puissance des différentes zones du réseau afin d'optimiser le coût de production

L'étape nous intéressant le plus est le réglage secondaire car c'est durant ce dernier que se produit le réglage autonome de la puissance des groupes asservis.

Le réglage autonome se fait généralement à l'aide de régulateurs PID qui sont privilégiés pour leur simplicité et leur robustesse.

Ce type de régulateurs atteint néanmoins rapidement ses limites car les paramètres du PID sont fixes. Ces paramètres sont généralement réglés pour un ensemble spécifique de conditions de fonctionnement. Lorsque ces conditions changent, tel que des variations de charge importantes ou des perturbations externes, les performances du PID peuvent se dégrader. Les charges électriques peuvent varier considérablement d'une saison à l'autre par exemple. Une combinaison de paramètres peut être satisfaisante pour l'hiver mais non pour l'été.

Il est donc important d'introduire des concepts d'intelligence artificielle (IA) pour obtenir des régulateurs plus dynamiques et plus performants.

Il y a deux méthodes différentes pour introduire l'IA dans l'AGC [2] :

- 1- Garder le régulateur PID et optimiser ses paramètres de manière dynamique grâce aux métaheuristiques. Les métaheuristiques sont des algorithmes intelligents d'optimisation tel les algorithmes génétiques ou l'optimisation par essaims de particules.
- 2- Se débarrasser complètement du régulateur PID et le remplacer par un régulateur intelligent tel un régulateur flou.

### **Objectif et plan de travail :**

L'objectif de ce travail est d'appliquer des régulateurs intelligents hybrides alliant la logique floue et les réseaux de neurones dans l'AGC d'un réseau, afin de remplacer les régulateurs PID et d'optimiser les régulateurs flous.

Le mémoire est divisé en 4 chapitres :

L'introduction générale a introduit la problématique du contrôle de la fréquence du réseau, la stratégie de commande AGC et les limites des régulateurs PID.

**Le chapitre 1** propose la modélisation du réseau électrique afin de le simuler sur les logiciels adéquats, ainsi que l'effet du régulateur PID sur la régulation de fréquence.

**Le chapitre 2** présente le principe de la logique floue et son implémentation dans la conception d'un régulateur flou intelligent.

**Le chapitre 3** détaille le principe de fonctionnement et la conception des régulateurs neuro-flous ainsi que leur implémentation dans l'AGC du réseau.

**Le chapitre 4** présente une application que nous avons créé sur MATLAB « appdesigner » afin de regrouper et de présenter les différentes simulations qui ont été proposé.

Le mémoire se clôturera par une conclusion générale qui résumera les résultats obtenus au cours de ce travail en plus de quelques perspectives de recherche qui pourront s'avérer intéressantes.

# **Chapitre 1 :**

**Modélisation du réseau électrique et  
régulation par PID**

## 1. Modélisation du réseau électrique :

Le réseau électrique est constitué de centrales de production (Générateurs Synchrones-GS entraînés par des turbines), de lignes de transport et de charges. Pour l'étude informatique de l'AGC, en considérant les composantes du réseau linéaires, nous les représentons par des modèles linéaires à travers leurs fonctions de transfert [3].

L'outil Simulink du logiciel Matlab peut être utilisé afin de modéliser et de simuler des schémas de réseaux électriques pour étudier la problématique du contrôle de fréquence.

### 1. 1. Modélisation du générateur synchrone (GS) :

En régime stable et permanent, le couple électrique appelé par le réseau (ou couple résistant  $C_r$ ) est égal au couple mécanique fourni par la turbine ( $C_m$ ).

Lors des perturbations, un déséquilibre se produit sous la forme d'un couple d'accélération ( $C_a$ ) qui résulte de la différence entre le couple électrique et le couple mécanique. Selon le principe de rotation des corps à inertie, nous obtenons :

$$C_a = C_m - C_r = J \frac{d^2 \delta_m}{dt^2} = J \frac{d\Omega_m}{dt} \quad (1.1)$$

$J$  [kg.m<sup>2</sup>] : Moment d'inertie de l'axe du GS avec la turbine.

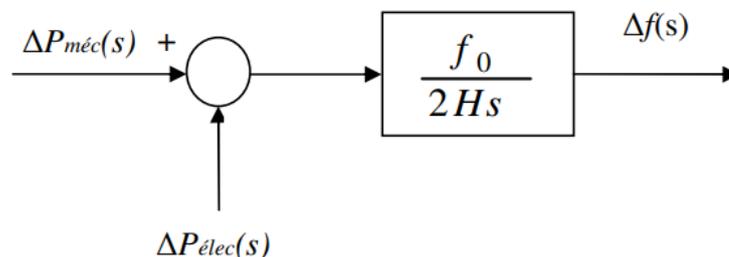
$\Omega_m$  [rad/s] : Vitesse angulaire du rotor.

$\delta_m$  [rad] : Position du rotor avant la perturbation.

En introduisant l'énergie cinétique du rotor et en développant les équations, nous aboutissons à l'expression de fréquence suivante :

$$\Delta f(s) = \frac{f_0}{2Hs} [\Delta P_{méc}(s) - \Delta P_{élec}(s)] (Hz) \quad (1.2)$$

Nous pouvons traduire l'équation (2.2) par le schéma bloc suivant :



**Figure 1.1** : Modèle du Générateur Synchrone [3]

## 1. 2. Modélisation de la charge :

Par rapport au problème d'AGC, les charges électriques peuvent être divisés en deux groupes :

- Charges indépendantes de la fréquence du réseau : se sont les charges résistives tel l'éclairage électrique.
- Charges dépendantes de la fréquence du réseau : appareils avec une grande sensibilité à la fréquence comme les moteurs électriques.

Le modèle de la charge sera donc la somme des deux types de charges, tel que :

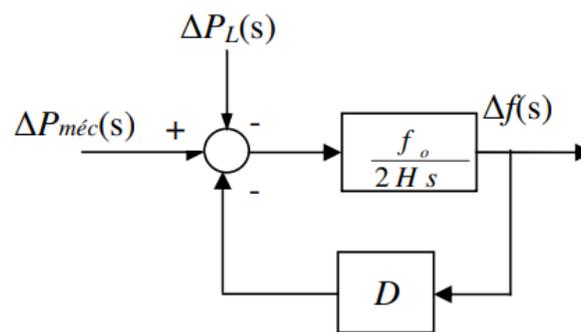
$$\Delta P_{elec}(pu) = \Delta P_L(pu) + D\Delta f(pu) \quad (1.3)$$

$\Delta P_L$ : Charge non sensible à la fréquence.

$D\Delta f$ : Charge sensible à la fréquence.

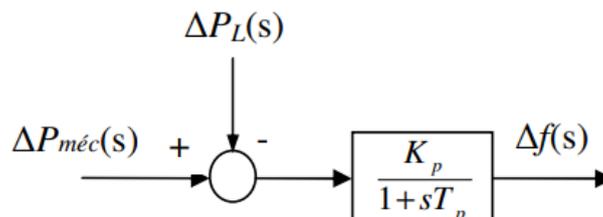
$D$ : Rapport entre le pourcentage de changement dans la charge et le pourcentage de changement dans la fréquence.

A partir de la modélisation des GS, nous pouvons créer une boucle de feedback qui modélise la charge électrique.



**Figure 1.2** : Modèle du générateur avec charge [3]

Nous pouvons simplifier cette boucle en introduisant les blocs dans une seule fonction de transfert comme le montre le **Figure 1.3** :



**Figure 1.3** : Schéma réduit du générateur avec charge [3]

$K_p = 1/D$  : Gain du système électrique.

$T_p = 2H/Df_0$  : Constante de temps du système électrique.

### 1. 3. Modélisation de la turbine :

La turbine d'une centrale électrique confère de la puissance mécanique au générateur synchrone. La turbine peut être alimentée par des chutes d'eau dans le cas d'une turbine hydraulique, par de la vapeur d'eau chauffée par la combustion de ressources fossiles pour les turbines à vapeur ou par la propulsion de gaz dans les turbines à gaz.

Dans le cas des centrales thermiques, la turbine fournit sa puissance mécanique ( $\Delta P_{méc}$ ) en fonction des positions des valves de vapeur ( $\Delta P_v$ ). Une façon simple de modéliser la turbine est de l'approximer par une fonction de transfert avec une constante de temps  $T_t$ , comme le montre la formule (2.4) :

$$G_T(s) = \frac{\Delta P_{méc}(s)}{\Delta P_v(s)} = \frac{1}{1+T_t s} \quad (1.4)$$

Cette fonction de transfert est traduite sous la forme de modèle ci-dessous :

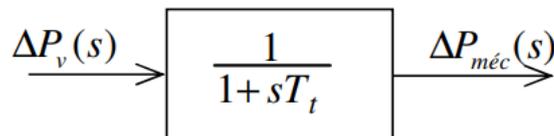


Figure 1.4 : Schéma de la turbine [3]

### 1. 4. Modélisation du régulateur de vitesse (Gouvernail) :

Les déséquilibres entre la puissance mécanique et électrique du système GS-Turbine provoquent des chutes de vitesse de la turbine qui font chuter à leur tour la fréquence du générateur.

Le rôle du régulateur de vitesse est de mesurer la vitesse de rotation de la turbine afin d'ajuster l'admission de vapeur.

La régulation de vitesse ( $R$ ) représente le rapport entre la variation de la vitesse et la variation de la puissance électrique.

$$R = \frac{\Delta w}{\Delta P_{elec}} \quad (1.5)$$

Le régulateur de vitesse est désigné pour maintenir la vitesse de rotation à sa valeur nominale. La caractéristique de régime établi du gouvernail est représentée dans la **Figure 1.5** :

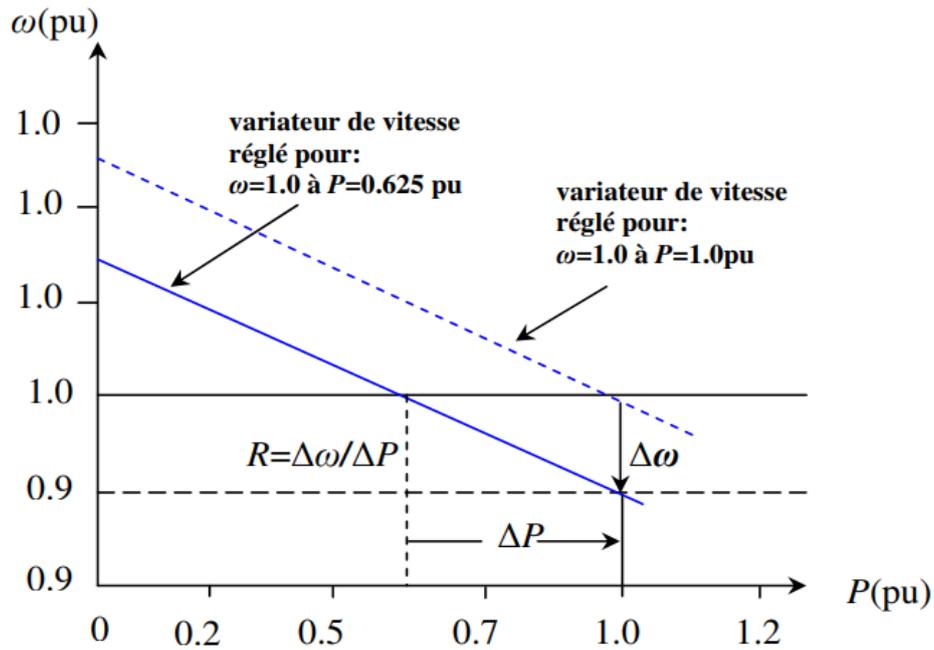


Figure 1.5 : Caractéristique de vitesse du régulateur [3]

Le régulateur de vitesse fonctionne comme un comparateur en calculant la différence entre la puissance de consigne  $\Delta P_{ref}$  et la puissance électrique  $\Delta P_{elec} = \frac{\Delta w}{R}$ .

$$\Delta P_g = \Delta P_{ref} - \frac{1}{R} \Delta w \tag{1.6}$$

Le passage par la transformée de Laplace nous donne l'expression :

$$\Delta P_g(s) = \Delta P_{ref}(s) - \frac{1}{R} \Delta \Omega(s) \tag{1.7}$$

Pour terminer, la commande  $\Delta P_g(s)$  est transformée en la commande de la position des valves  $\Delta P_v$  avec un retard entre les deux commandes qui est traduit par l'introduction de la constante de temps  $T_g$ , nous donnant la fonction de transfert :

$$\Delta P_v(s) = \frac{1}{1+T_g s} \Delta P_g(s) \tag{1.8}$$

Le schéma final qui représente l'action du régulateur de vitesse est représenté par la **Figure 1.6** :

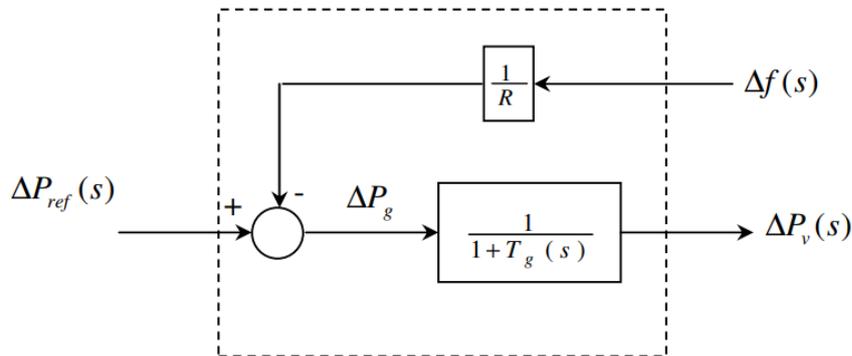


Figure 1.6 : Schéma du régulateur de vitesse [3]

En assemblant les modèles des différentes parties du réseau, nous obtenons le schéma simple et linéaire d'un réseau à une zone :

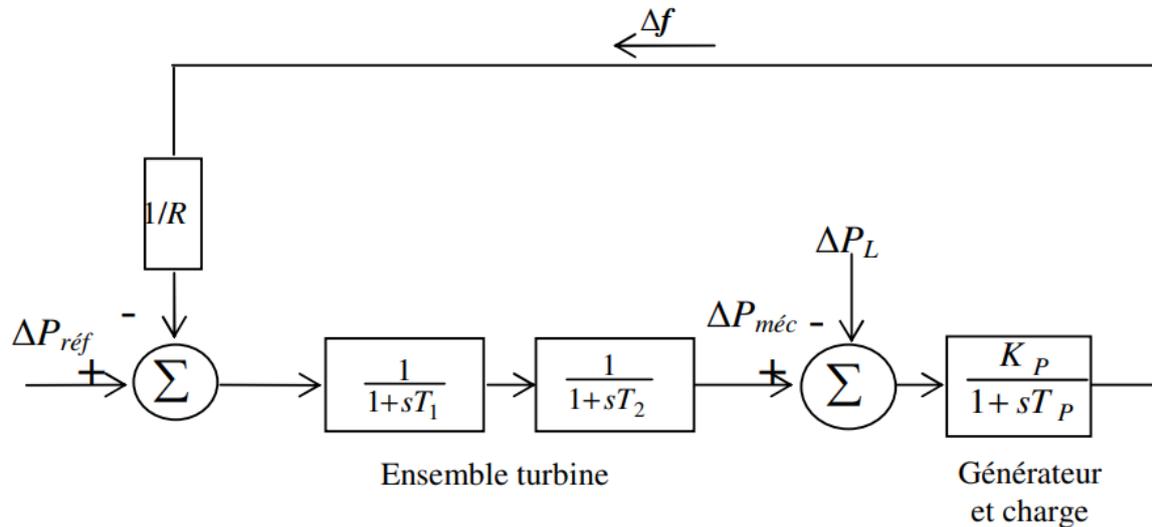


Figure 1.7 : Modèle d'un réseau primaire [3]

avec :

$T_1 = T_g$  : la constante de régulateur de vitesse (gouverneur/gouvernail).

$T_2 = T_t$  : la constante de la turbine.

## 2. Régulation par PID :

### 2. 1. Cas d'une région autonome :

Dans le cas d'une région autonome, le groupe électrique équivalent règle seul la fréquence du réseau. C'est un cas rare où la zone ou région autonome n'est reliée à aucune autre zone. Le groupe sera chargé de maintenir sa puissance égale à celle du réseau lors des variations de charges pour maintenir la stabilité de fréquence.

#### - Réglage primaire :

Le réglage primaire représente la première étape du réglage et se produit lors des premières secondes post-perturbation. L'équilibre offre-demande se rétablit grâce à la réserve d'inertie mécanique stockée dans les rotors des générateurs, ce réglage s'applique sur tous les générateurs de manière individuelle [1].

Dans cette étude, le modèle théorique sera basé sur la centrale thermique de Tilghemt « MS9001E » dotée d'une turbine à gaz de puissance unitaire. Ses constantes caractéristiques sont les suivantes [3] :

$$T_1 = 0.05 \text{ s}; \quad T_2 = 2 \text{ Hz pu ou } 0.4 \text{ s}; \quad R = 4 \%; \quad K_p = 100 \text{ Hz pu MW}; \quad T_p = 35.2 \text{ s}; \quad D = 1\%.$$

L'alternateur a une inertie de  $17850 \text{ kg.m}^2$  et une vitesse de rotation nominale de  $3000 \text{ tr/mn}$  ou  $314.15 \text{ rad/s}$ . Son énergie cinétique est calculée par :

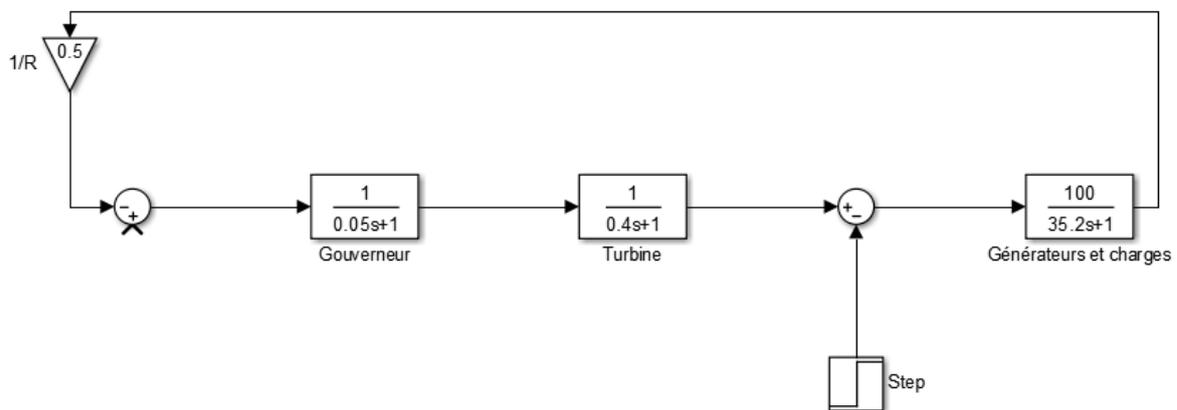
$$W = 0.5 \cdot J \cdot \Omega^2 = 0.5 \cdot 17850 \cdot (314.15)^2 = 880810235.81 \text{ J}$$

En supposant que l'énergie cinétique est celle de tout le système électrique à la puissance de base de  $100 \text{ MW}$ , la constante d'inertie sera :

$$H = W/100 = 880810235.81/100 \cdot 10^6 = 8.8 \text{ s}$$

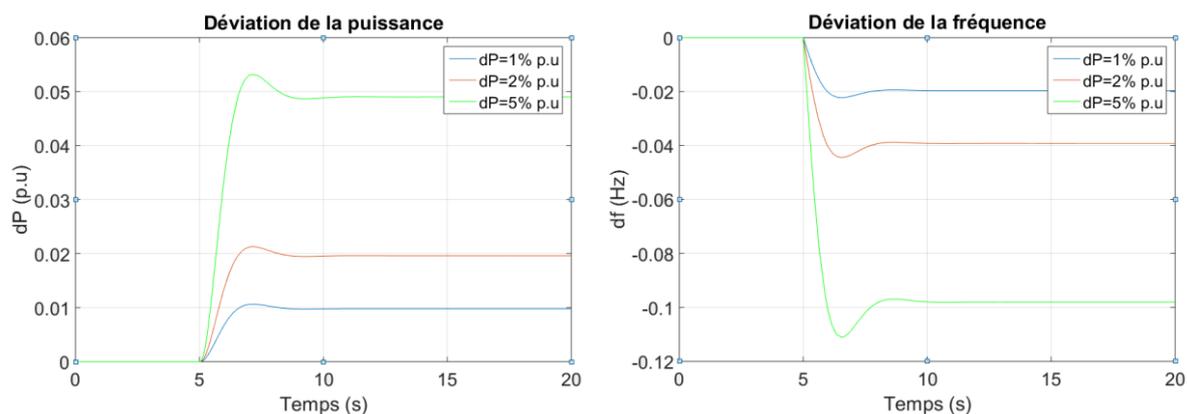
Le schéma Simulink général d'un réseau à une zone est représenté par la **Figure 1.7**.

A partir de ces données, nous pouvons construire le schéma de la centrale de Tilghemt.



**Figure 1.8** : Modèle d'un réseau primaire pour la centrale de Tilghemt

En appliquant diverses perturbations au niveau de la puissance de charge (Step) nous obtenons les résultats représentés sur la figure suivante :



**Figure 1.9** : Réponse du modèle suite à un changement de charge

A  $t=5 \text{ s}$ , nous appliquons une déviation de puissance de 1%, 2% et 5% ( $0.01 \text{ p.u.}$ ,  $0.02 \text{ p.u.}$  et  $0.05 \text{ p.u.}$ ) qui représentent l'apparition d'une charge électrique à cet instant. La puissance mécanique du réseau augmente et tend vers la valeur de la déviation de puissance sans l'égaliser tandis que la fréquence chute brusquement avant d'être stabilisée par le réglage primaire. Néanmoins, la fréquence ne revient pas à sa valeur nominale. De faibles valeurs d'erreurs

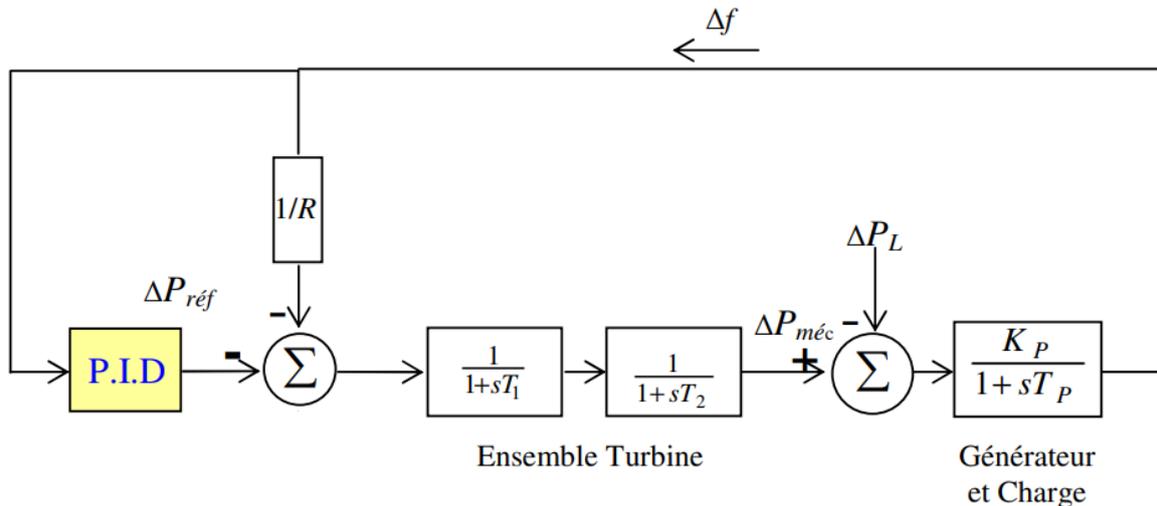
subsistent suite au réglage primaire, ce qui peut entraîner des complications si la valeur moyenne des erreurs de la fréquence au niveau des différents groupes n'est pas nulle.

Il est donc essentiel d'introduire une étape de réglage supplémentaire pour résoudre ce problème.

**- Réglage secondaire :**

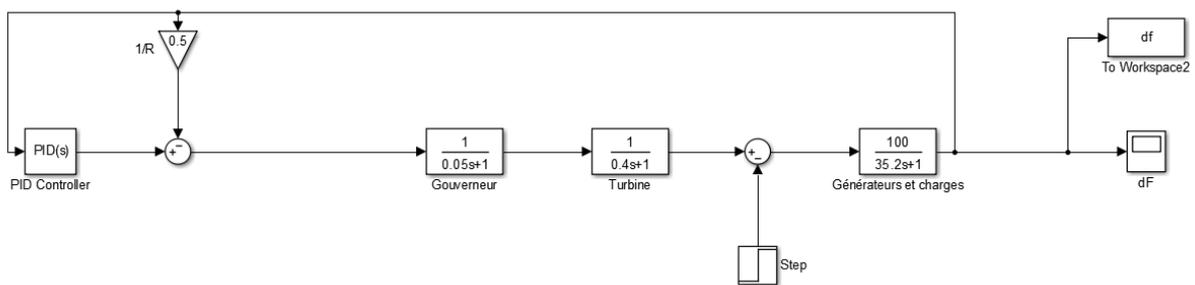
Suite à la l'écart de fréquence subsistant à la fin du réglage primaire, une autre forme de réglage s'impose pour le corriger. Le réglage secondaire opère durant les premières minutes du réglage et consiste à rétablir la fréquence en régulant la puissance de référence des groupes asservis, ce réglage s'applique donc uniquement au niveau des générateurs de la zone source de la perturbation. Une telle manœuvre est conduite par des régulateurs Proportionnels, Intégraux et Dérivés (PID) ou intelligents. Une zone constituée de plusieurs centrales peut être considérée comme une machine équivalente car le contrôle de la vitesse doit être unifier en son sein [1].

Pour obtenir le réglage secondaire au niveau d'un générateur, nous relient le régulateur à la sortie de l'erreur de fréquence afin d'obtenir la puissance de référence à travers une boucle fermée.



**Figure 1.10 :** Modèle d'un réseau avec réglage secondaire [3]

En introduisons les valeurs de la centrale de Tilghemt nous obtenons le schéma suivant :



**Figure 1.11 :** Modèle d'un réseau avec réglage secondaire pour la centrale de Tilghemt

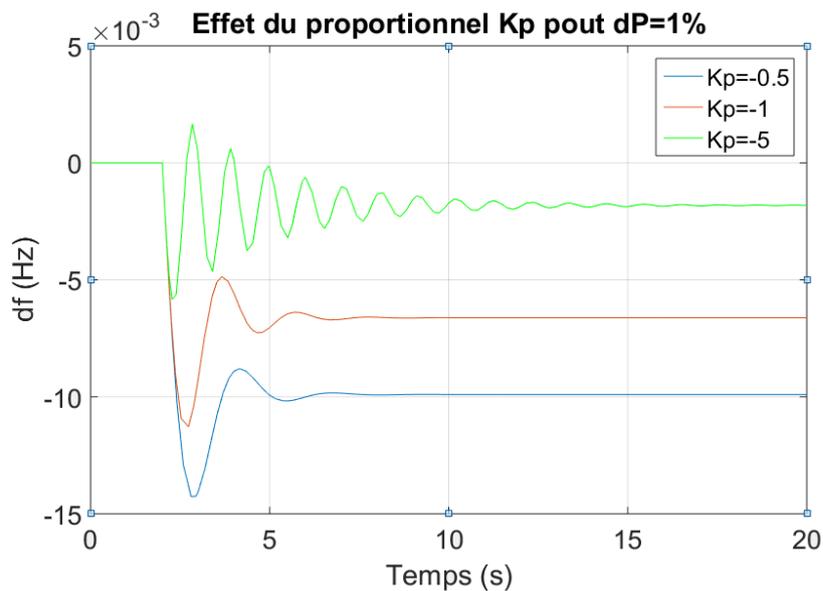
Dans plusieurs applications de l'ingénierie dont la régulation de fréquence, le régulateur PID est utilisé pour sa robustesse et sa simplicité d'implémentation.

Le régulateur PID est constitué de trois blocs opérateurs : le Proportionnel, l'Intégrateur et le Dérivatif, chacun ayant un comportement différent.

### L'action proportionnel P :

Elle s'écrit sous la forme d'une constante  $K_p$  qui est multiplié par l'erreur du système  $e(t)$  donnant l'expression :  $K_p * e(t)$ .

Avec l'augmentation de la valeur du correcteur  $K_p$ , l'erreur diminuera sans jamais s'annuler. Cela est dû au fait que plus le système se régule, plus l'erreur diminue et plus la valeur de régulation  $K_p * e(t)$  diminue aussi résultant ainsi en un cercle vicieux où l'erreur tendra vers zéro sans y arriver.



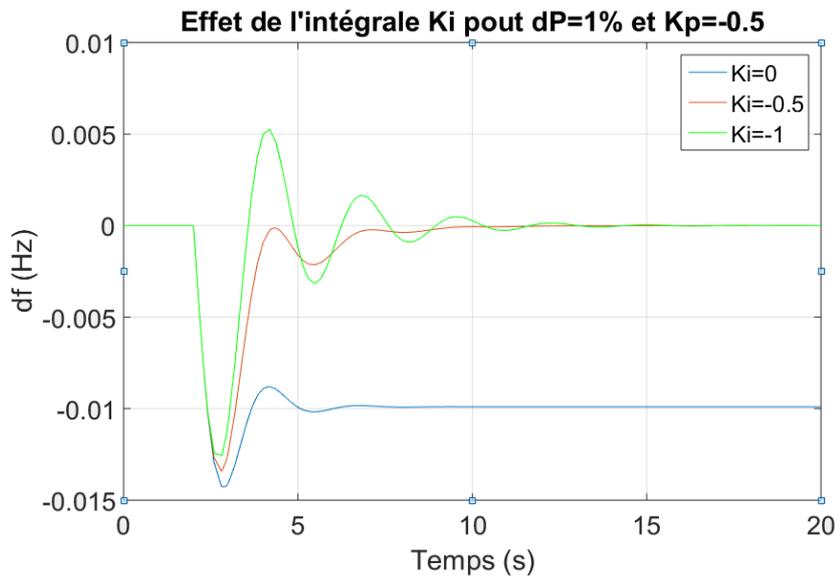
**Figure 1.12 :** Effet du proportionnel sur l'erreur de fréquence

La **Figure 1.12** montre que l'erreur diminue avec l'augmentation de  $K_p$  en valeur absolue. Des valeurs trop importantes de  $K_p$  peuvent néanmoins perturber le système comme le montre le cas  $K_p = -5$ .

### L'action intégrale I :

L'action intégrale est également écrite sous la forme d'une constante  $K_i$  mais contrairement au proportionnel, le  $K_i$  est multiplié par l'intégrale de l'erreur du système  $\int_0^t e(t) dt$ .

L'introduction de l'intégrale nous permet de faire intervenir la composante passée de la régulation. L'intégral agit comme une sorte de mémoire en prenant compte de la somme accumulée de l'erreur au fil du temps. Lorsque l'erreur converge vers zéro et donc que l'action proportionnelle atteint ses limites, l'action intégrale offrira la valeur de régulation nécessaire à la stabilisation du système à travers une régulation linéaire (contrairement au P qui est constante) grâce à l'intégrale  $K_i * \int_0^t e(t) dt$ .



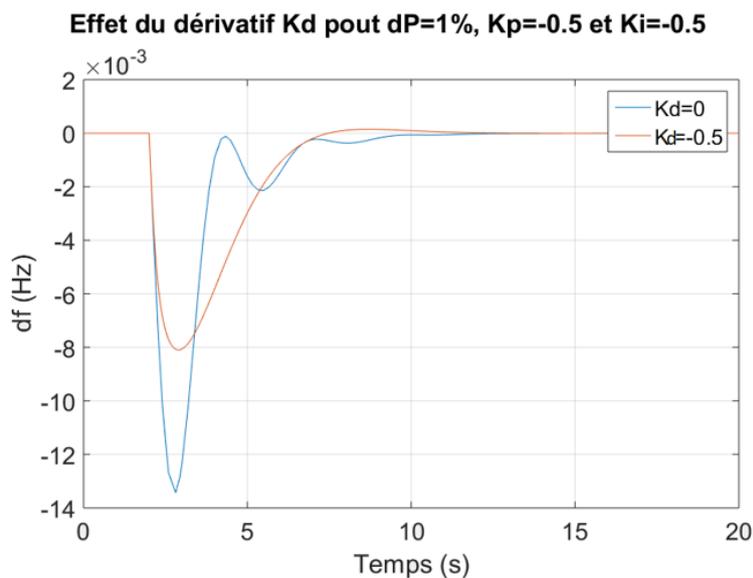
**Figure 1.13 :** Effet de l'intégrale sur l'erreur de fréquence

La **Figure 1.13** montre que l'action intégrale aide bien à atteindre la convergence nulle. Néanmoins, trop augmenter la valeur de  $K_i$  augmente les oscillations du système.

### L'action dérivative D :

Si l'action intégrale sert à modéliser le passé du système, la composante dérivative représente le futur. La constante de l'action dérivative  $K_d$  est multiplié par la dérivée de l'erreur pour obtenir la valeur de régulation  $K_d * \frac{de(t)}{dt}$ .

L'action intégrale résulte en un certain nombre d'oscillation et de dépassements selon ses valeurs, perturbant la stabilité du système. Le rôle de la composante  $K_d$  est d'étudier le comportement de la variable régulée à travers sa dérivée pour savoir quand freiner la régulation lorsque le correcteur se rapproche trop vite de la valeur voulue.



**Figure 1.14 :** Effet du dérivatif sur l'erreur de fréquence

Selon la **Figure 1.14** la chute de fréquence a diminué et les oscillations ont été atténuées grâce à l'intervention de l'action dérivative.

Le meilleur type de régulateur dans ce cas d'AGC est donc le régulateur PID complet.

**Remarque :** Chaque centrale de la zone fonctionne à une puissance inférieure à sa capacité maximale, laissant ainsi une marge de puissance dans le cas de besoins imprévus ou d'urgences. Lorsqu'une zone ne peut plus satisfaire la demande après avoir utilisé toute sa marge de puissance, nous rentrons dans le cadre du réglage tertiaire.

#### - Réglage tertiaire :

Le réglage tertiaire utilise les liaisons interzones. Un ajustement de la production au niveau de différentes zones est appliqué pour subvenir à la demande actuelle tout en prenant en considération les coûts de production [1].

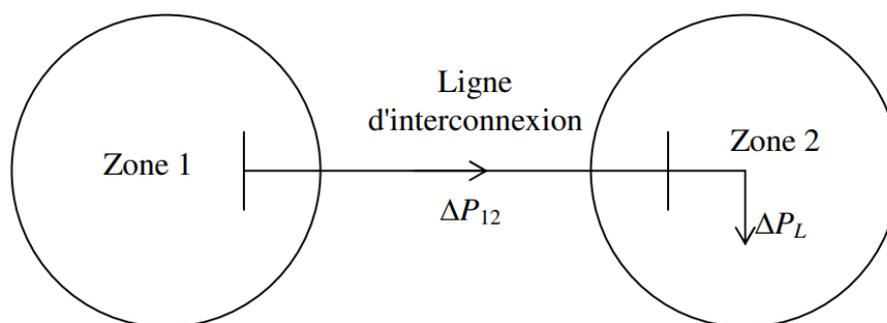
Ce réglage se fait manuellement, il ne sera donc pas traité dans ce travail.

## **2. 2. Cas de deux zones interconnectées :**

Les réseaux électriques sont divisés en plusieurs zones de propriétés et dimensionnement différents. Les zones sont reliées entre elles par des interconnexions qui leur permettent de se soutenir mutuellement en échangeant de la puissance.

Les zones autonomes ont besoins de réserves tournantes ou froides d'énergie pour satisfaire les heures de pointe. Ce besoin est réduit dans le cas de deux zones ou plus car chaque zone peut tirer de la puissance des autres, à travers leurs interconnexions. De plus, les zones n'ont généralement pas les mêmes heures de pointe ce qui leur permet de combler ou de se faire combler leurs manques de puissance.

Une interconnexion à deux zones est généralement représentée comme le montre la **Figure 1.15** :



**Figure 1.15 :** Représentation simplifiée de deux zones interconnectées [3]

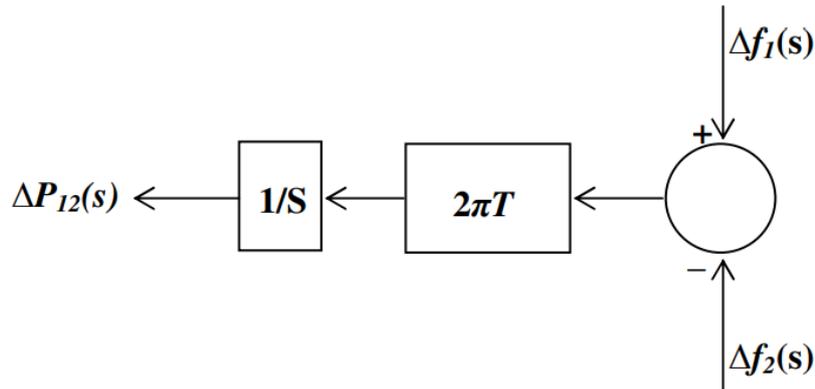
Les zones possèdent des fréquences différentes pour une liaison faible, mais les liaisons reliant les différentes zones sont généralement assez fortes pour faire en sorte que toutes les zones aient la même fréquence.

**2. 2. 1. Modélisation de l'interconnexion :**

La formule décrivant la puissance de transit passant par l'interconnexion en fonction des erreurs de fréquence des deux zones s'écrit comme suit [2] :

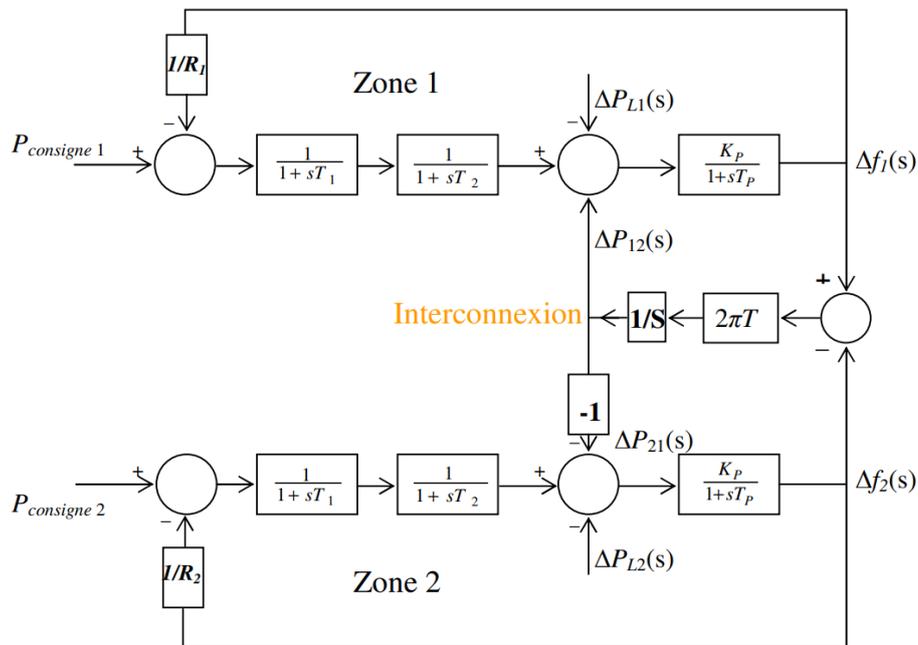
$$\Delta P_{12}(s) = \frac{2\pi T}{s} (\Delta f_1(s) - \Delta f_2(s)) \tag{1.9}$$

Nous pouvons traduire l'équation (2.9) en un schéma linéaire donné dans la figure suivante :



**Figure 1.16 :** Schéma de l'interconnexion interzones [3]

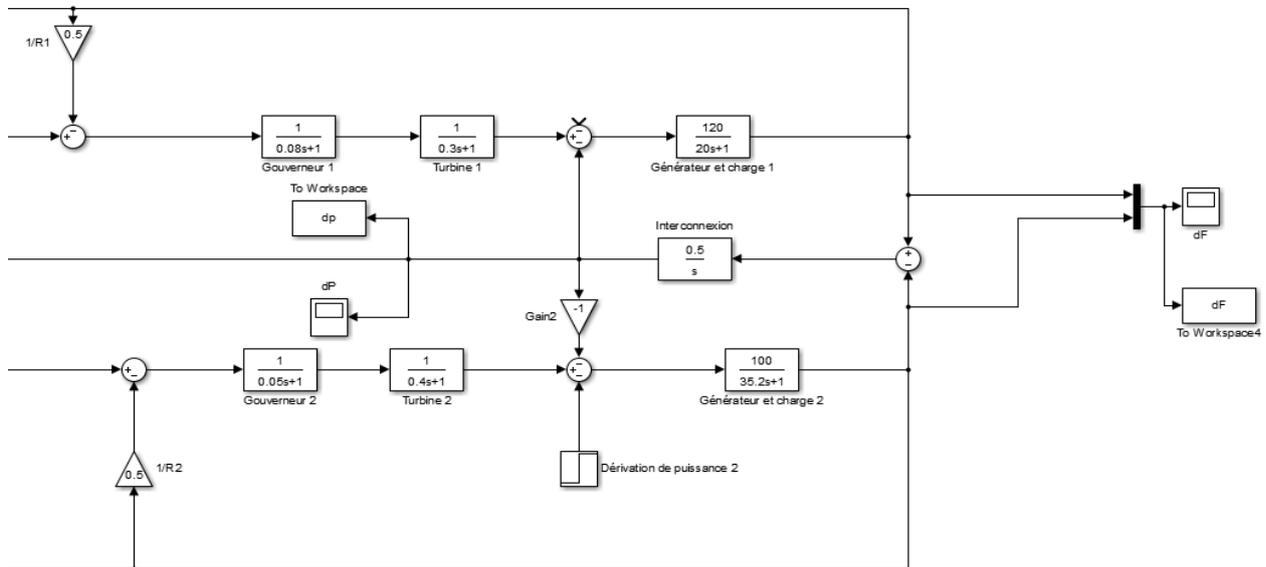
Pour modéliser les deux zones, il nous suffit de modéliser chaque zone et de les relier grâce à l'interconnexion où la puissance de transit sera introduite dans chacune des zones comme suit :



**Figure 1.17 :** Modèle de deux zones interconnectées avec réglage primaire [3]

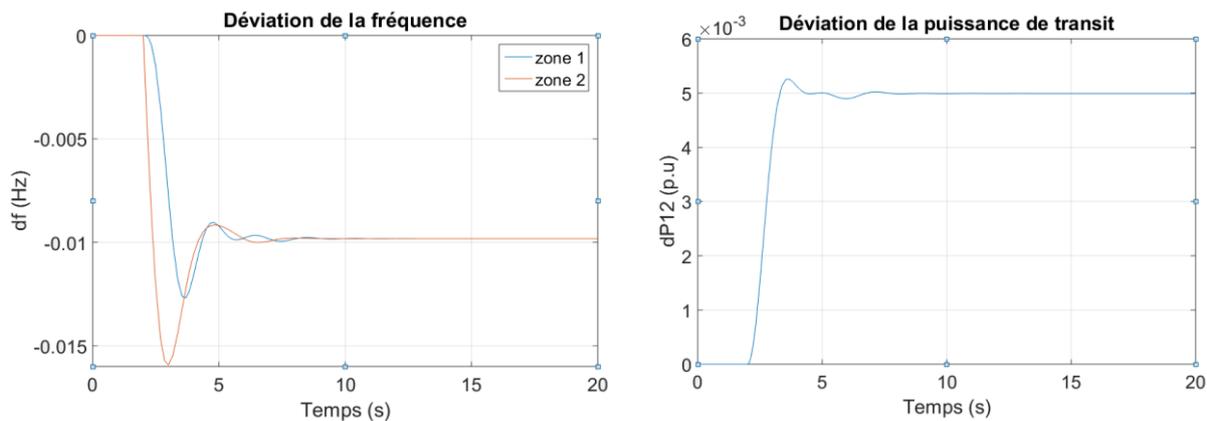
**2. 2. 2. Réponse dynamique du système à deux zones :**

La **Figure 2.18** représente le modèle Simulink d'un cas deux zones avec une perturbation de puissance de 1% (0.01p.u) au niveau de la zone 2.



**Figure 1.18 :** Modèle Simulink de deux zones interconnectées avec réglage primaire

Les résultats de la simulation sont illustrés dans la figure suivante :



**Figure 1.19 :** Réponse du modèle de simulation à 2 zones avec réglage primaire

De la **Figure 1.19** nous remarquons que la fréquence ne converge pas vers sa valeur nominale, il faut donc appliquer un réglage secondaire pour éliminer l'erreur subsistante. Nous remarquons aussi l'augmentation de la puissance de transit parcourant l'interconnexion entre les deux zones  $\Delta P_{12}$ . La puissance de transit est censée être nulle lorsque le réseau est en état d'équilibre, il faut donc également trouver un moyen d'annuler cette puissance.

C'est là que l'Area Control Error (ACE) entre en jeu. L'ACE représente l'erreur régulée dans le réglage secondaire pour les réseaux à plusieurs zones. Contrairement au cas mono-zone, le régulateur ne peut pas se contenter de l'erreur de fréquence comme unique entrée car la

puissance de transit doit également être annulée. L'ACE représente donc la somme de l'erreur de fréquence et de la puissance de transit qui sera corrigée par un régulateur PID afin de fournir une puissance de référence qui éliminera ces deux erreurs [2].

$$\begin{aligned}
 ACE_1 &= \Delta P_{12} + B_1 \Delta f_1 \\
 ACE_2 &= \Delta P_{21} + B_2 \Delta f_2
 \end{aligned}
 \tag{1.10}$$

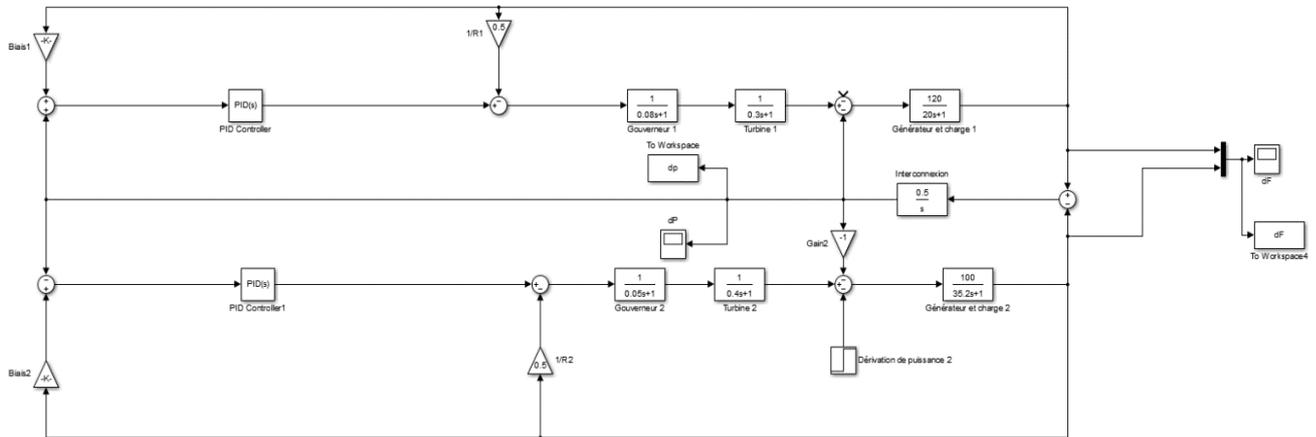


Figure 1.20 : Modèle Simulink de deux zones interconnectées avec réglage secondaire

Les deux paramètres  $B_1$  et  $B_2$  représentent les biais qui multiplient les erreurs de fréquence. Les biais servent à atteindre un fonctionnement satisfaisant du système, ils sont calculés selon la formule :

$$B_i = D_i + \frac{1}{R_i}
 \tag{1.11}$$

Les résultats de la simulation avec l'ACE sont représentés dans la figure ci-dessous :

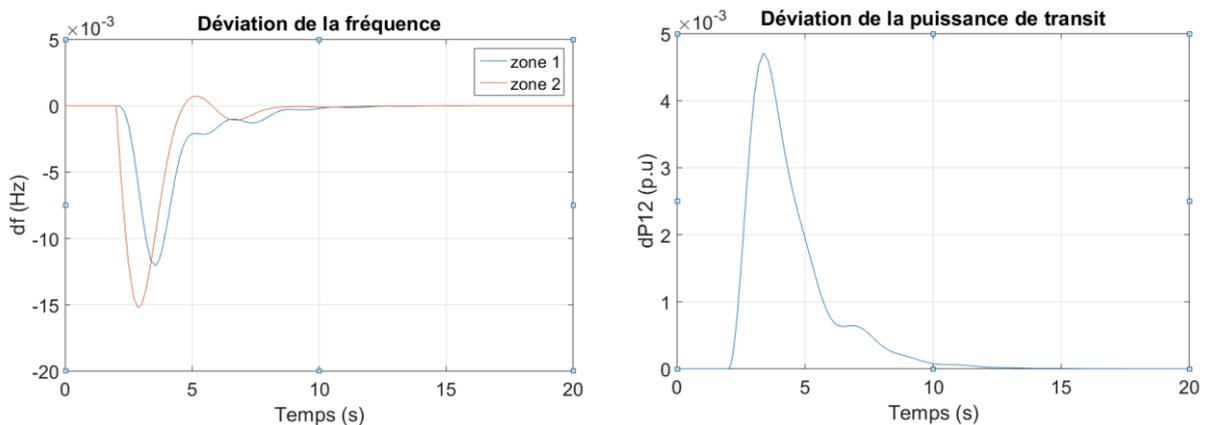
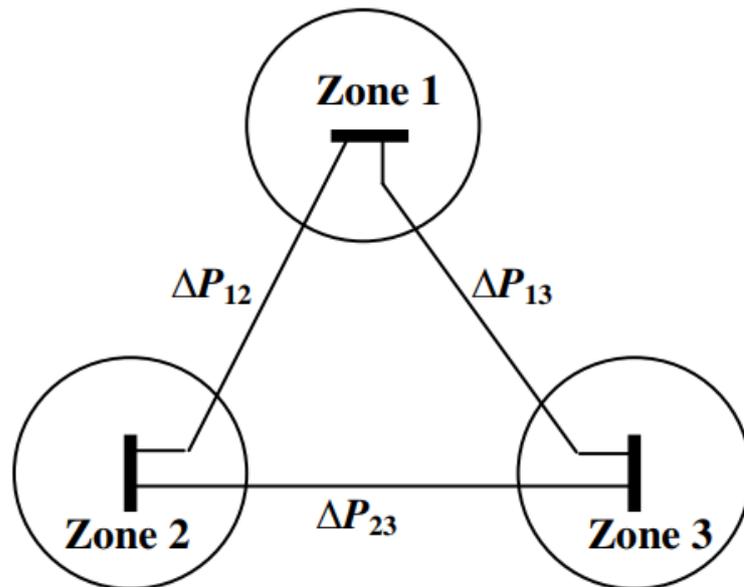


Figure 1.21 : Réponse du modèle de simulation à 2 zones avec réglage secondaire

La Figure 1.21 montre la convergence de l'erreur de fréquence pour les deux zones et de la puissance de transit vers zéro. L'erreur ACE devient donc nulle à l'équilibre statique.

### 2. 3. Cas de trois zones interconnectées :

Dans la pratique, une zone est connectée à plusieurs autres zones formant ainsi un réseau électrique dense et complet. Pour se rapprocher de la réalité nous pouvons simuler le cas de trois zones toutes interconnectées entre elles.



**Figure 1.22 :** Représentation simplifiée de 3 zones interconnectées [3]

Comme pour le cas à deux zones, l'erreur ACE est utilisée pour la régulation du système. L'ACE d'une seule zone devra prendre en compte les puissances des différentes liaisons et les erreurs de fréquence des différentes zones pour fournir une information complète.

$$ACE_i = \sum_1^3 \Delta P_{ij} + B_i * \Delta f_i \quad (1.12)$$

Le schéma Simulink du cas trois zones régulés par trois régulateurs PID est représenté par la figure suivante :

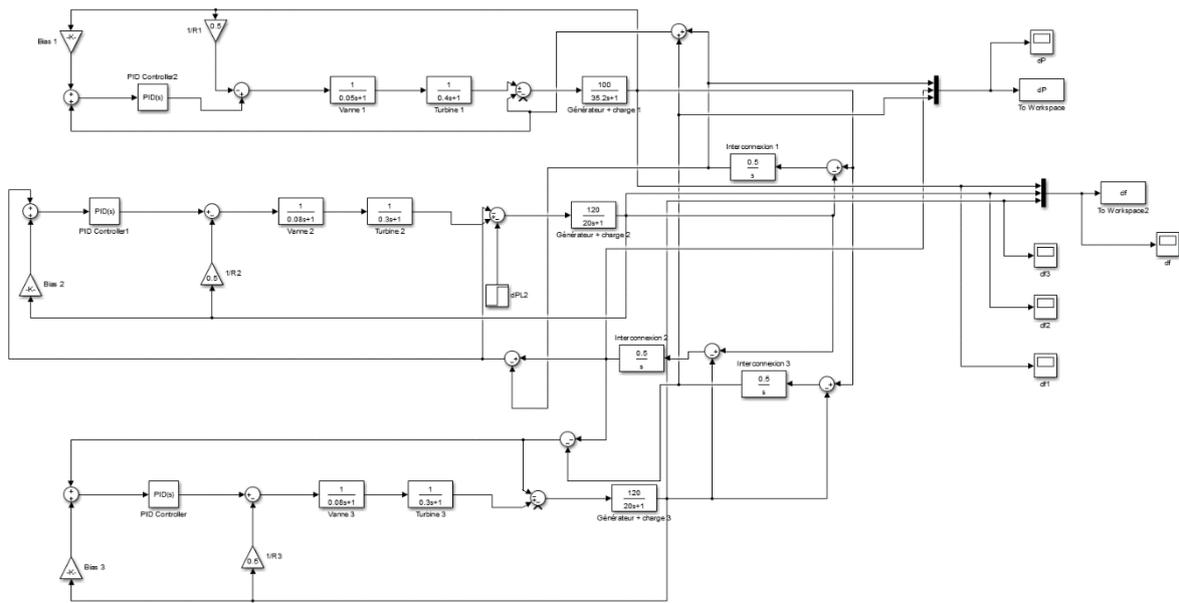


Figure 1.23 : Modèle Simulink de trois zones interconnectées avec réglage secondaire

Après simulation d'une déviation de puissance de 1% dans la zone 2, nous obtenons les résultats :

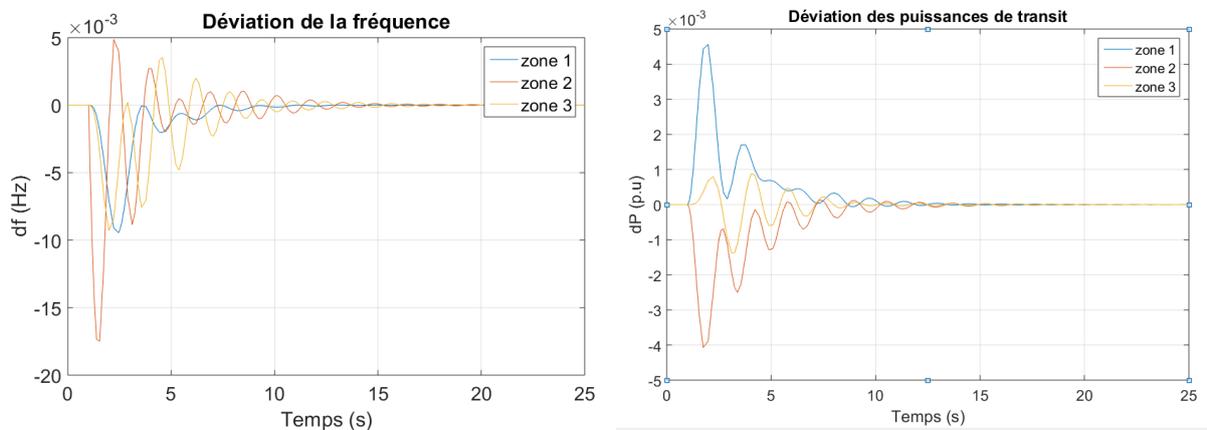


Figure 1.24 : Réponse du modèle de simulation à 3 zones avec réglage secondaire

D'après la Figure 1.24, nous voyons qu'il y a des déviations de fréquence au niveau de chaque zone et des déviations de puissance de transit au niveau de chaque interconnexion. Cela montre l'implication de chaque zone dans la régulation du système.

Grâce aux trois correcteurs PID et aux ACE au niveau de chaque zone, les erreurs de fréquence et les puissances de transit tendent toutes vers zéro pour atteindre un état stable.

## **2. 4. Conclusion :**

Le régulateur PID représente une solution à la perturbation de fréquence et ce même pour des cas multizones. La composante intégrale du PID est la plus importante car le réglage primaire agit plus ou moins comme l'action proportionnelle, laissant l'erreur relativement faible et constante. Les composants P et D apportent quant à eux une amélioration des performances du PID si tous les paramètres sont choisis de manière adéquate.

Malgré les performances acceptables du PID, le temps de réponse du régulateur et la forte chute de fréquence dont souffre le système en sa présence font de lui un correcteur non optimal pour l'AGC. C'est pour cela que l'on penche vers l'introduction de l'intelligence artificielle pour essayer d'obtenir des résultats plus satisfaisants.

# **Chapitre 2 :**

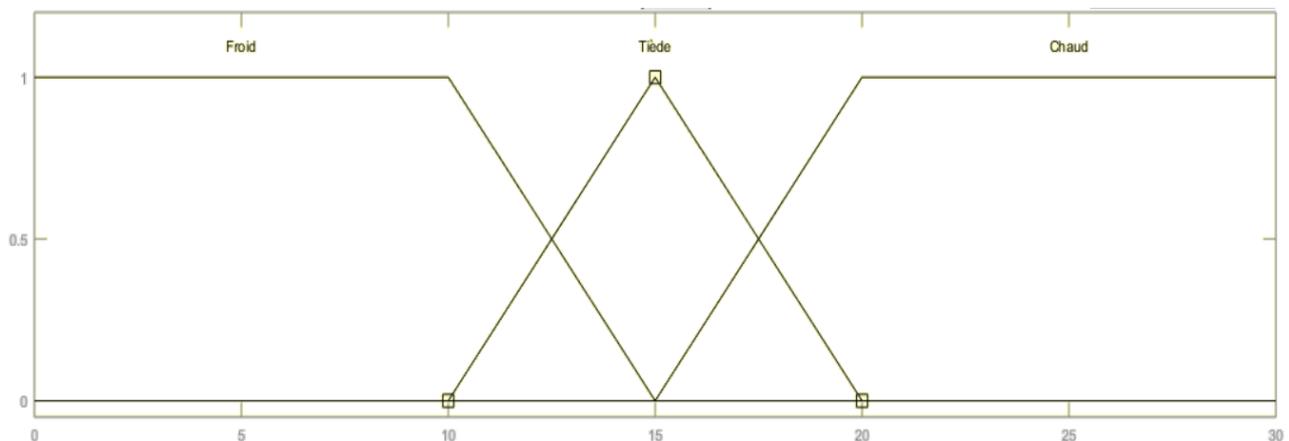
**Contrôleur flou appliqué à l'AGC**

## 1. Introduction à la logique floue :

La logique floue est un outil du domaine de l'intelligence artificiel. Formalisée par le mathématicien et informaticien iranien Lotfi Zadeh en 1960. La logique floue est ce qu'on appelle une logique polyvalente qui contrairement à la logique classique binaire (vrai ou faux) propose plus de nuances à travers un raisonnement « humain ». Les grandeurs numériques appartiennent à différents concepts vagues et incertains (chaud, froid...) avec des valeurs de vérité partielles, c'est-à-dire entre 0 et 1. Une variable de température peut par exemple être froide à une valeur de 0.3 (soit 30%) et en même temps chaude à une valeur de 0.7 (soit 70%).

La logique floue se montre très efficace dans des applications qui demandent un raisonnement humain, là où l'expertise et l'expérience de l'homme font une différence [4].

Nous pouvons illustrer cela à travers un exemple traitant des grandeurs de température par exemple :



**Figure 2.1 :** Fonctions d'appartenance floues pour une variable de chaleur

Selon ce modèle, la température est considérée comme étant froide en dessous de 10°, tiède à 15° et chaude au-dessus de 20°.

Il y'a néanmoins des intervalles flous où les fonctions d'appartenance (les ensembles : Froid, Tiède et Chaud) se rencontrent. Par exemple, de 10° à 15°, la température peut être considérée comme froide ou tiède. Une valeur numérique tombant sous cet intervalle appartiendra à plusieurs ensembles en même temps avec un certain pourcentage ou degrés d'appartenance pour chaque ensemble.

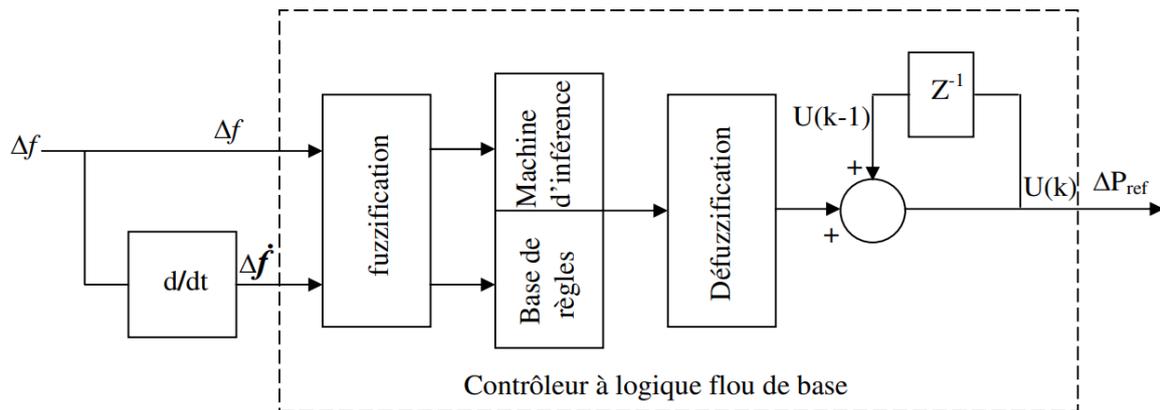
Le température  $T=12^\circ$  par exemple est froide à 40% (degrés d'appartenance=0.4) et tiède à 60% (degrés d'appartenance=0.6). Cela sert à représenter le doute et l'ambiguïté humains car chaque personne à un ressenti différent envers la température selon leur position géographique, et la limite entre les 2 concepts de « chaud » et « froid » reste floue et non absolue.

## 2. Application de la logique floue dans l'AGC :

### 2.1. Conception d'un régulateur flou :

Il existe plusieurs approches différentes pour modéliser un système flou. Les deux méthodes les plus utilisées sont la méthode de Mamdani et la méthode de Sugéno. Pour les régulateurs flous, le modèle présenté ci-dessous sera celui de Mamdani.

Le régulateur flou se met à la place du PID, le remplaçant complètement. La figure ci-dessous représente la structure d'un régulateur classique [3].



**Figure 2.2 :** Structure d'un régulateur flou classique [2]

Un régulateur flou peut contenir une ou plusieurs entrées. Pour l'AGC, 2 entrées sont généralement utilisées, l'erreur de fréquence et sa dérivée dans le cas mono-zone, l'ACE et sa dérivée dans le cas de 2 zones ou plus. La sortie quant à elle, est la puissance de référence  $\Delta P_{ref}$  qui sera appliquée au gouverneur.

Le contrôle à base de logique floue se fait sur 3 étapes [5].

#### - La fuzzification :

Comme vu précédemment, les valeurs numériques de l'erreur de fréquence  $\Delta f$  et de sa dérivée  $\Delta \dot{f}$  sont traduites sous la forme de fonctions d'appartenance floues décrivant les différents états des variables d'entrées. Dans ce cas, les fonctions d'appartenance possèdent 5 ensembles [6].

NG (Négative Grande) : valeur négative grande.

NM (Négative Moyenne) : valeur négative moyenne.

EZ (Environ Zéro) : valeur nulle ou presque nulle.

PM (Positive Moyenne) : valeur positive moyenne.

PG (Positive Grande) : valeur positive grande.

Les fonctions d'appartenance des variables floues sur l'outil « fuzzy » de MATLAB sont représentés comme suit [2] :

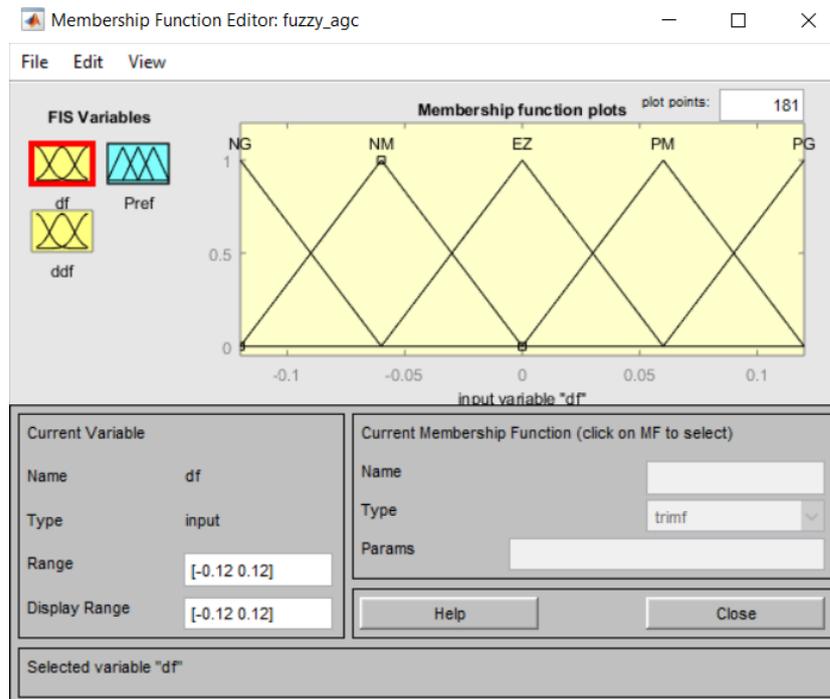


Figure 2.3 : Fonctions d'appartenance de  $\Delta f$  sur MATLAB [2]

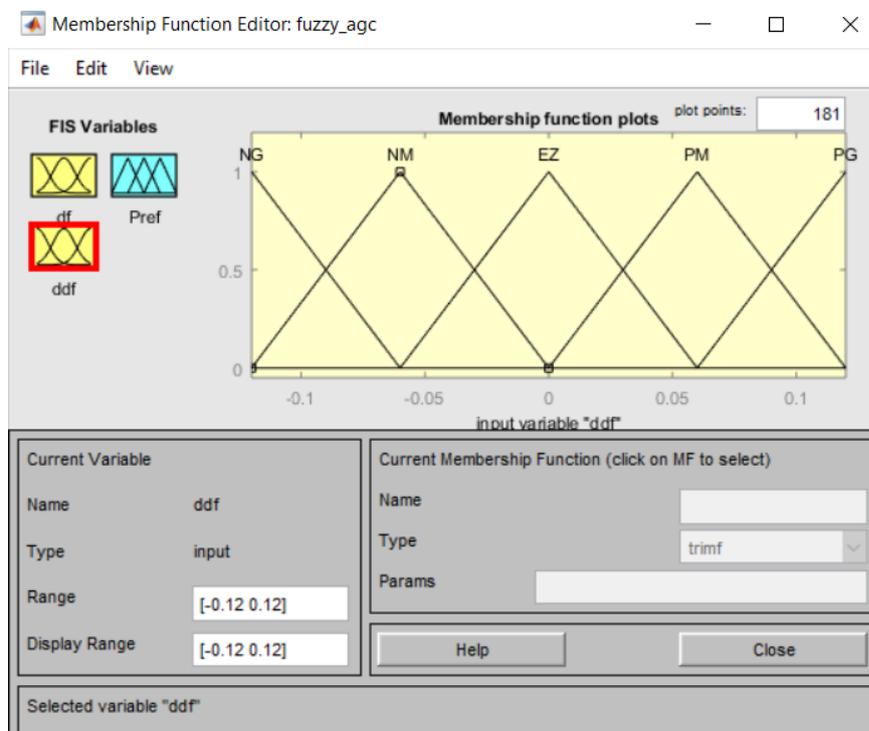


Figure 2.4 : Fonctions d'appartenance de  $\Delta \dot{f}$  sur MATLAB [2]

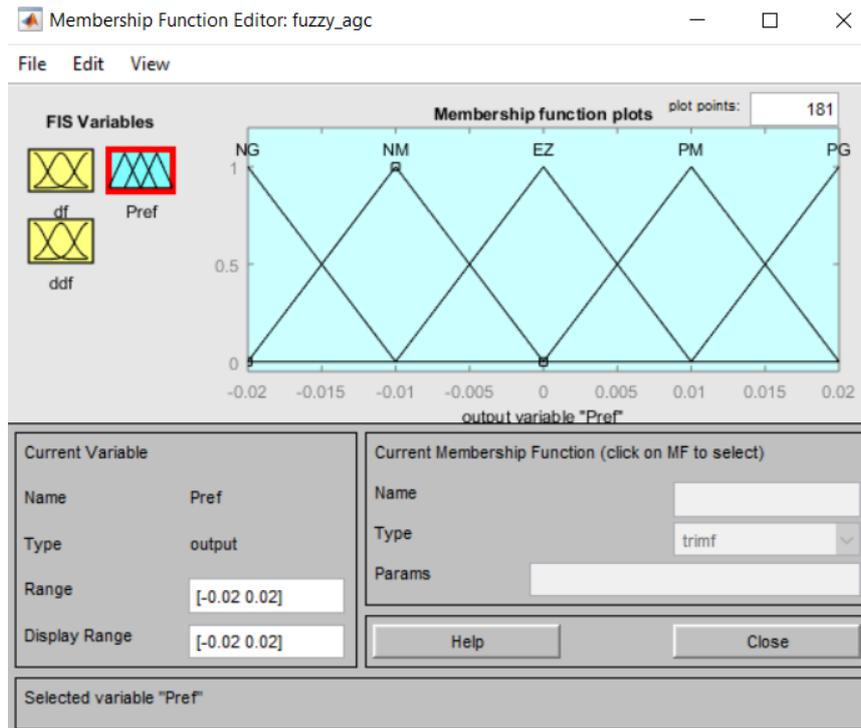


Figure 2.5 : Fonctions d'appartenance de  $\Delta P_{ref}$  sur MATLAB [2]

#### - Détermination des jeux de règles :

Aussi appelé mécanisme d'inférence, la conception des règles d'un contrôleur flou s'écrit sous la forme de conditions sur les variables d'entrées accompagnées d'opérations logiques qui déterminent le résultat de la variable de sortie [7].

**Opération** = Si [condition (variable 1) **et/ou** condition (variable 2)] **alors** [résultat (variable de sortie)].

Exemple :

Si ( $\Delta f$  est PG et  $\Delta \dot{f}$  est PG) alors ( $\Delta P_{ref}$  sera NG).

Cette opération signifie que si l'erreur de fréquence et sa dérivée ont toutes les deux des valeurs positives grandes, la puissance de régulation devra avoir une valeur négative grande pour faire baisser l'erreur.

En traitant tous les cas de figures, soit toutes les combinaisons entre les 5 ensembles de la variable 1 et les 5 ensembles de la variable 2, nous obtenons 25 règles qui sont représentés dans le tableau suivant :

$\Delta f$ \ $\Delta f$	<b>NG</b>	<b>NM</b>	<b>EZ</b>	<b>PM</b>	<b>PG</b>
<b>NG</b>	PG	PG	PM	PM	EZ
<b>NM</b>	PG	PM	PM	EZ	EZ
<b>EZ</b>	PM	PM	EZ	EZ	NM
<b>PM</b>	PM	EZ	EZ	PM	NG
<b>PG</b>	EZ	EZ	NM	PG	NG

Tableau 2.1 : Tables des règles du contrôleur flou [2]

Représentation des règles d'inférence sur l'outil « fuzzy » de Matlab :

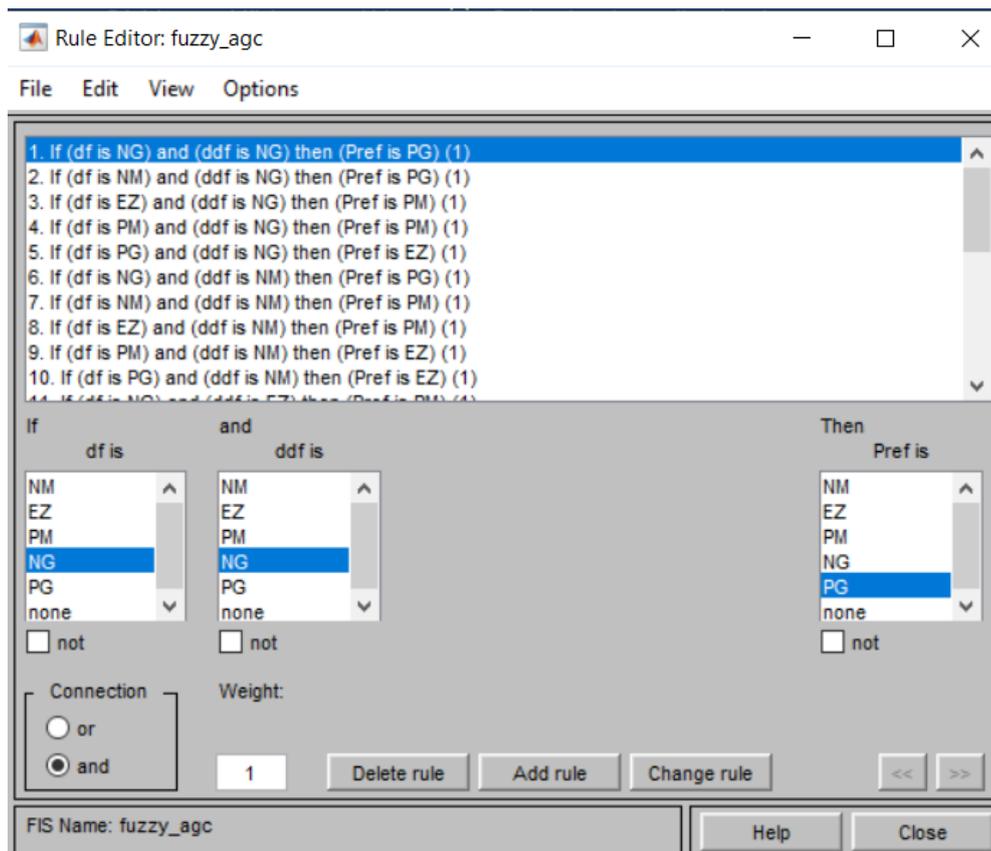


Figure 2.6 : Règles d'inférence du régulateur flou sur MATLAB

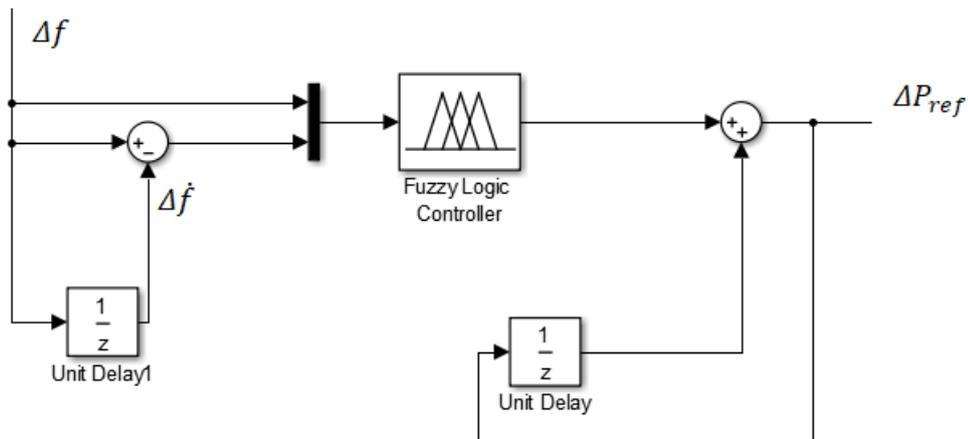
### - Défuzzification :

Puisque la variable de sortie est floue à la fin de l'inférence, il faut traduire le résultat en une valeur numérique pour l'implémentation dans le réseau.

Dans notre cas, c'est le contrôleur de Mamdani qui est utilisé. Sur Matlab, ce contrôleur utilise la méthode du centre de gravité qui consiste à obtenir l'abscisse du centre de gravité de la fonction d'appartenance obtenue post-inférence [4].

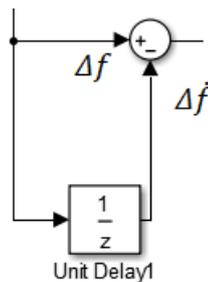
## 2.2. Implémentation du régulateur flou :

Sur Simulink, le régulateur flou se présente sous la forme du schéma suivant :



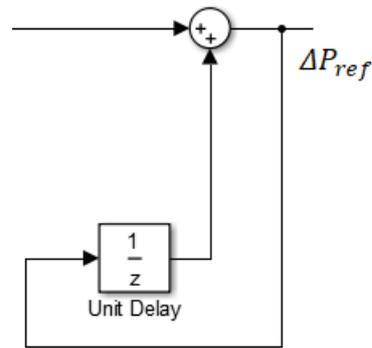
**Figure 2.7 :** Schéma Simulink d'un régulateur flou

Les fonctions « unit delay » engendrent un délai temporel discret. À tout moment, leur sortie est la valeur de leur entrée pour l'itération précédente.



**Figure 2.8 :** Schéma du calcul de la dérivée de l'erreur de fréquence

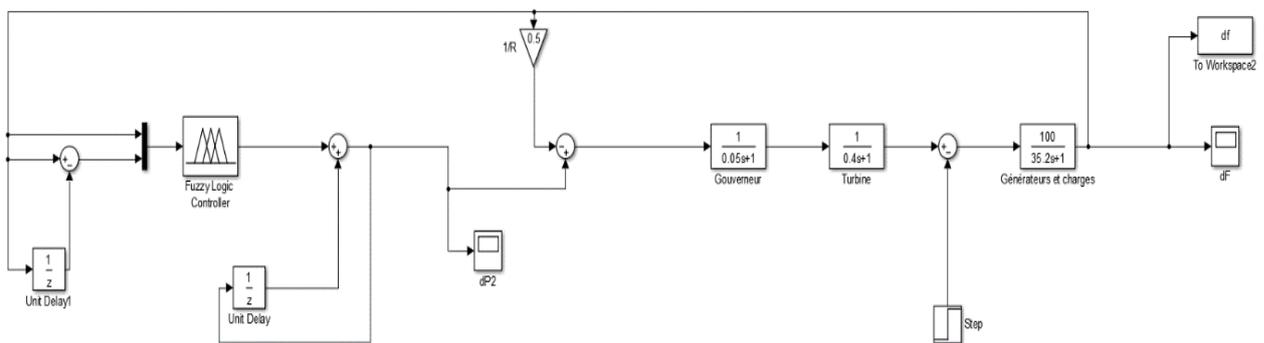
La **Figure 2.8** représente le calcul de la dérivée de l'erreur de fréquence  $\Delta \dot{f}$ ,  $(\Delta f_1 - \Delta f_0)$  de manière discrète.



**Figure 2.9 :** Schéma du calcul de l'action intégrale de la puissance de référence

Le « unit delay » de la **Figure 2.9** sert à incorporer une action intégrale à la puissance de référence  $\Delta P_{ref}$  ( $\Delta P_{ref}(1) + \Delta P_{ref}(0)$ ). Cette méthode, comme avec l'action intégrale, additionne la variable sur des moments consécutifs. Mais contrairement à l'action intégrale qui est continue, ici le calcul est discret.

Le « Fuzzy Logic Controller-FLC » quant à lui représente le cœur du contrôleur flou, étant l'outil qui applique la logique du régulateur flou avec ses 3 étapes (fuzzification, inférence, défuzzification) en temps réel.



**Figure 2.10 :** Schéma Simulink pour 1 zone avec régulateur flou

Les fonctions d'appartenance montrés dans les figures (**Figure 2.3**, **figure 2.4** et **Figure 2.5**) et les règles de la **Figure 2.6** constituent un régulateur flou que nous nommerons « flou1 ».

Les résultats obtenus pour différentes valeurs de variation de puissance sont illustrés sur la figure suivante :

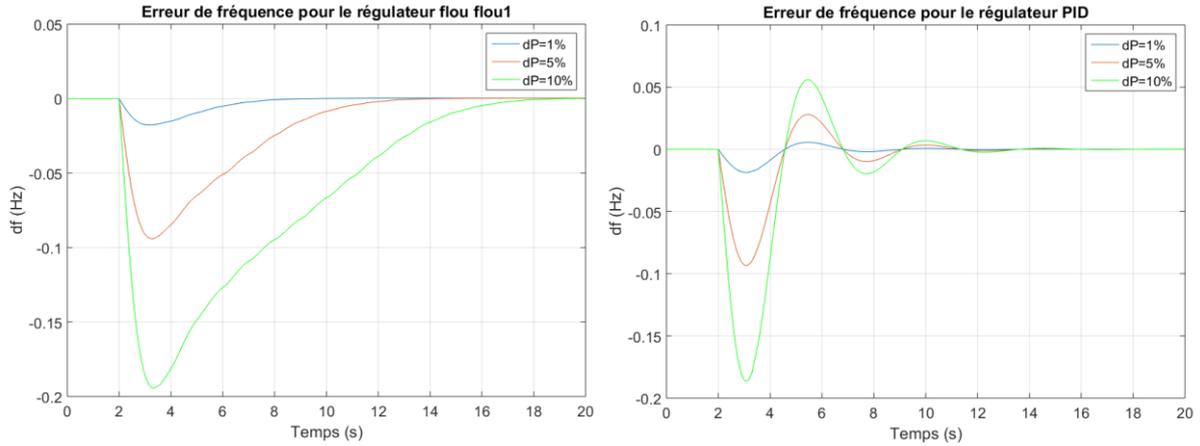


Figure 2.11 : Comparaison entre la réponse du système pour un régulateur PID et un fluu

Le régulateur fluu « flou1 » est bien fonctionnel vu sa convergence pour différentes valeurs de variation de puissance. Ce régulateur fluu offre une régulation beaucoup plus lisse avec une absence d'oscillations.

Néanmoins, les chutes de fréquence sont similaires au cas du PID et le temps de convergence est extrêmement lent atteignant les 18 secondes pour une déviation de puissance de 10%.

Nous voyons bien que ce régulateur fluu n'est pas assez intéressant. En modifiant les intervalles des fonctions d'appartenance et de leurs domaines, nous pouvons obtenir un régulateur avec les mêmes entrées et règles fondamentales qui offre une régulation tout à fait différente [3].

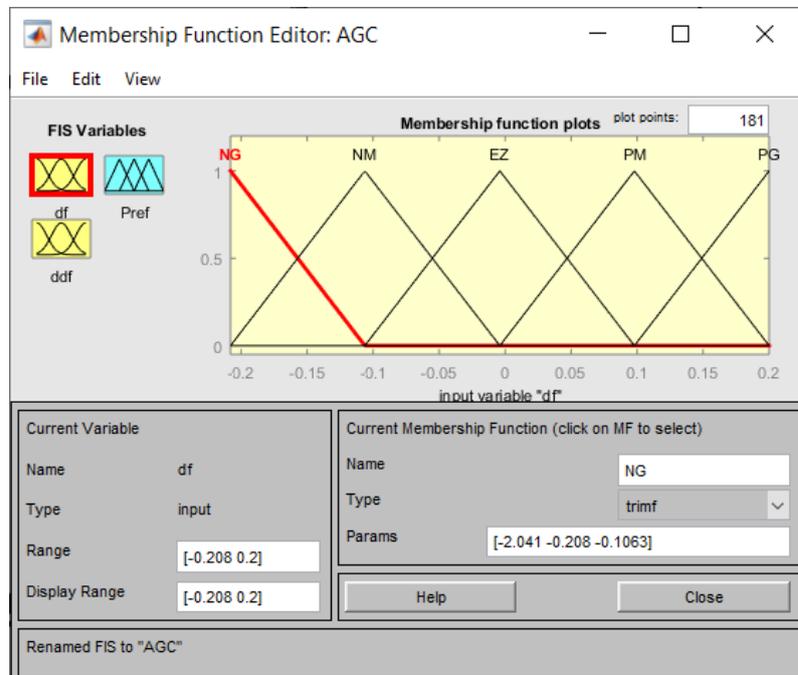


Figure 2.12 : Fonctions d'appartenance de  $\Delta f$  du régulateur fluu2

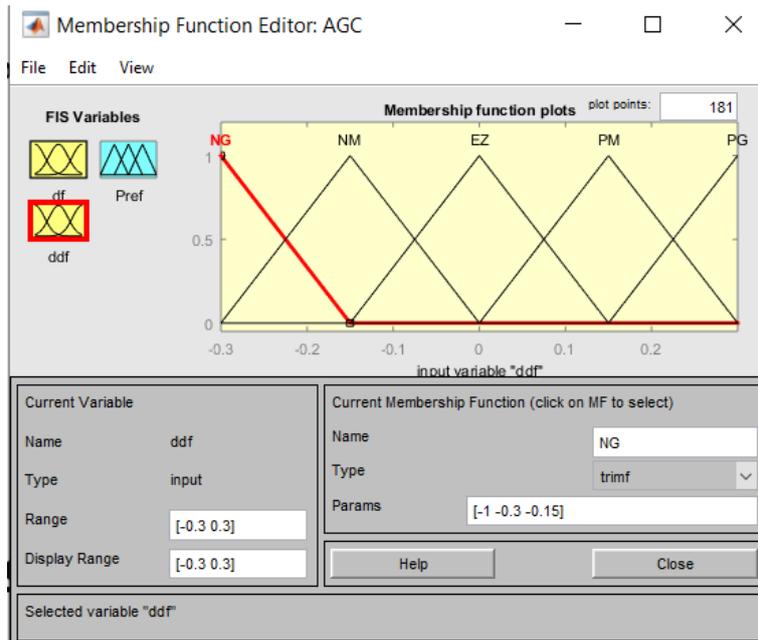


Figure 2.13 : Fonctions d'appartenance de  $\Delta \dot{f}$  du régulateur flou2

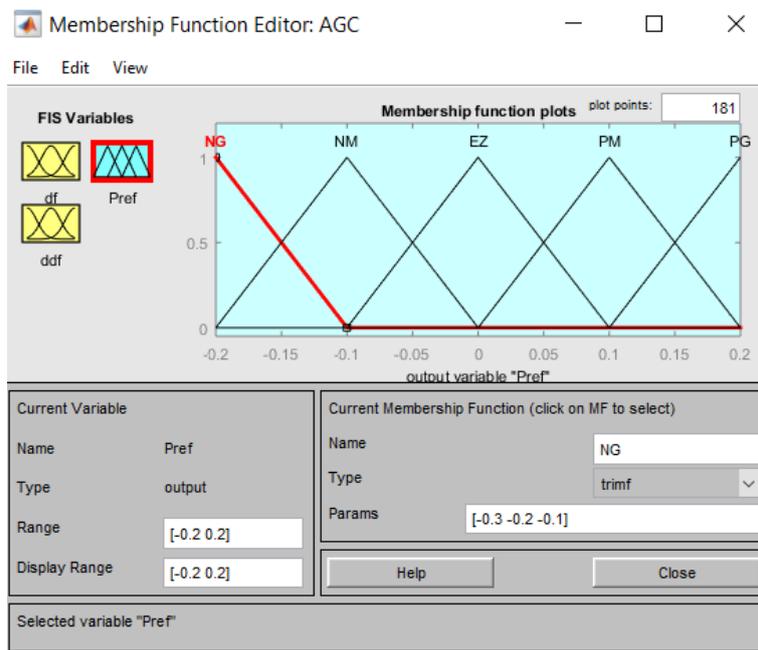
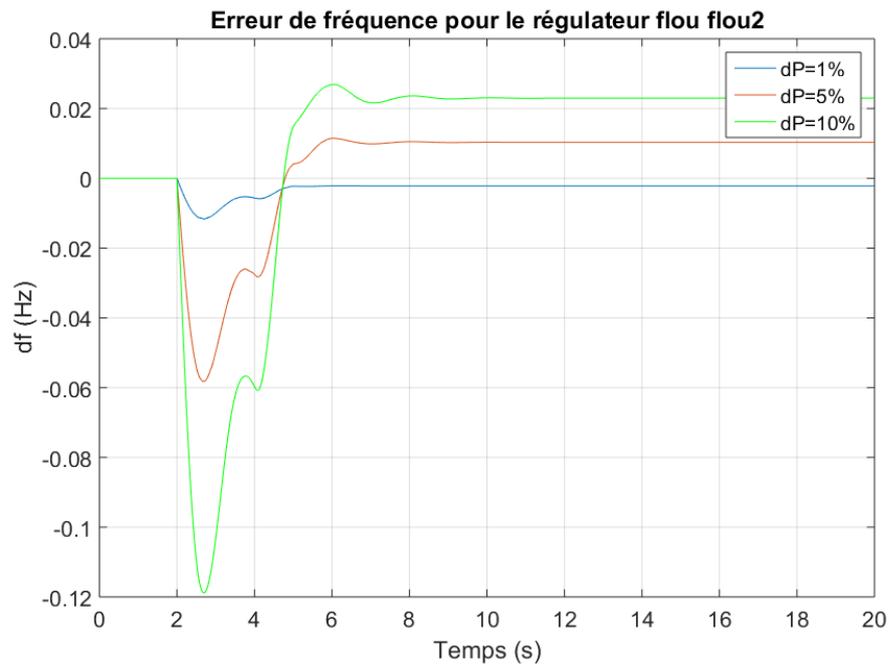


Figure 2.14 : Fonctions d'appartenance de  $\Delta P_{ref}$  du régulateur flou2

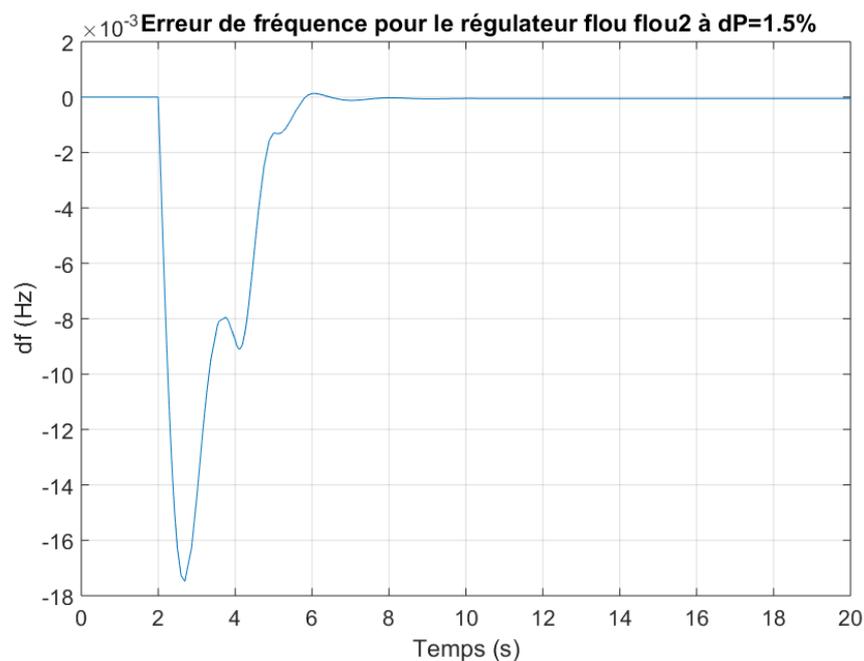
Test du régulateur « flou2 » :



**Figure 2.15 :** Réponse du système pour le régulateur flou2

Ce régulateur offre des chutes de fréquence bien plus faibles et des temps de convergence plus rapides par rapport aux régulateurs précédents. Néanmoins, il a un comportement un peu plus oscillatoire que celui du « flou1 » et il ne converge pas vers une valeur nulle pour la plupart des valeurs de variation de puissance.

Nous obtenons régulation qui tend vers des valeurs nulles dans le cas où la variation de puissance de charge ( $dP$ ) est égale à 1.5%.



**Figure 2.16 :** Réponse du système pour le régulateur flou2 pour  $dP=1.5\%$

A travers ces exemples, nous voyons l'important impact qu'a le choix des valeurs des intervalles des fonctions d'appartenance du régulateur flou sur son fonctionnement.

### **3. Conclusion :**

Les régulateurs flous offrent une alternative aux régulateurs PID. Néanmoins, l'une des forces du FLC qui est sa forte dépendance du jugement humain s'avère être également une faiblesse dans la régulation de la fréquence.

Le nombre des variables d'entrées, le nombre des fonctions d'appartenance de chaque variable, la forme des fonctions d'appartenance, les règles d'inférence et les valeurs des intervalles des fonctions d'appartenance dépendent tous directement des choix du concepteur du régulateur. Tandis que la plupart de ces paramètres possèdent des possibilités de changement assez restreintes (ex : nombres de variables d'entrées, entre 1 et 3 pour la plupart des cas) ou des conséquences facilement identifiables et comparables entre elles (ex : type et nombre des fonctions d'appartenance), le choix des intervalles se montre bien plus complexe car il offre un très grand nombre de combinaisons possibles et un impact important sur la réponse du régulateur.

La solution est donc d'utiliser des méthodes d'optimisation pour améliorer les paramètres du FLC. L'une des méthodes les plus efficaces est celle d'utiliser des réseaux de neurones pour la partie optimisation à travers les régulateurs neuro-flous. En sachant que l'une de leurs fonctions premières est l'optimisation des intervalles des fonctions d'appartenance des FLC.

# **Chapitre 3 :**

Contrôleur Neuro-flou appliqué à  
l'AGC

## **1. Introduction aux systèmes neuro-flous :**

Les systèmes neuro-flous sont des systèmes intelligents fusionnant la logique floue et les réseaux de neurones. Le concept a été introduit par J.R Jang en 1993 [8]. Il n'existe pas de méthodes standards pour transformer le savoir et l'expérience humains en règles d'inférences flous, il faut donc trouver une approche efficace pour régler les paramètres des fonctions d'appartenance flous pour maximiser les performances du régulateur.

Il y'a deux approches principales concernant les systèmes neuro-flous :

- Réseaux neuro-flous : incorporation de la logique floue dans les réseaux de neurones.
- Systèmes d'inférence neuro-flous : incorporation de l'entraînement et l'apprentissage des réseaux de neurones dans les systèmes d'inférence de la logique floue.

Dans ce travail, nous avons utilisé la seconde méthode qui est les systèmes d'inférence neuro-flous car nous voulons préserver les caractéristiques du régulateur flou. Le système d'inférence utilisé ici est plus exactement connu sous le nom de **ANFIS** (Adaptive Network-Based Fuzzy Inference System).

L'ANFIS est une technique d'optimisation des systèmes d'inférence flous de type Sugéno grâce à un réseau de neurones.

La méthode de Sugéno est l'une des approches de la modélisation des régulateurs flous. Contrairement à la méthode de Mamdani, la variable de sortie n'est pas floue mais plutôt sous la forme d'équations mathématiques linéaires contenant des paramètres optimisables. La forme mathématique de la variable de sortie offre plus d'opportunités d'optimisation à travers le réseau de neurones intervenant dans le système neuro-flou ce qui fait de la méthode de Sugéno une méthode plus intéressante dans l'application de l'ANFIS.

## **2. Théorie derrière la conception d'un ANFIS :**

### **2. 1. Modèle de l'ANFIS :**

La modèle ANFIS traduit les étapes de la construction du modèle flou de Sugéno en introduisant chaque étape sous la forme d'une couche de neurones.

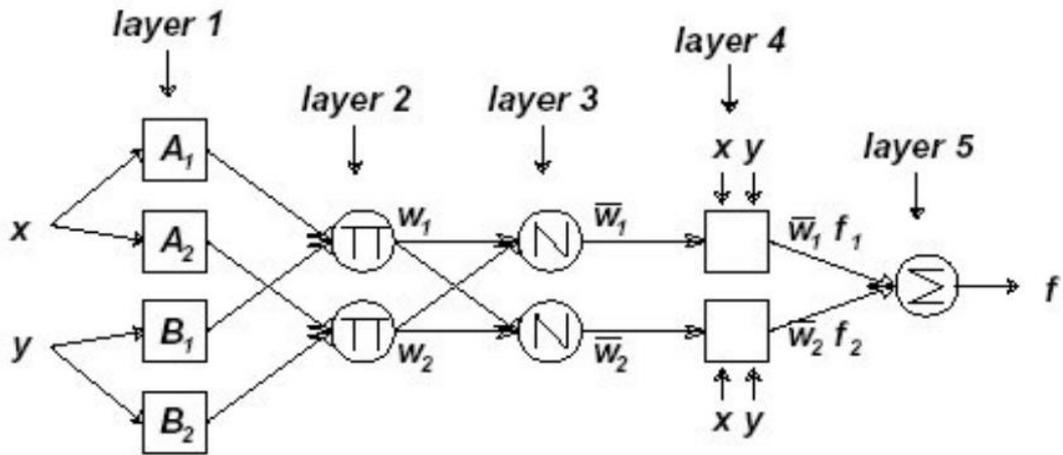
Dans le modèle de Sugéno, nous calculons une fonction conséquence  $f_i$  qui dépend linéairement des deux variables d'entrées  $x$  et  $y$  à partir de chaque règle, suivant les équations [9] :

$$f_i = p_i * x + q_i * y + r_i \quad (3.1)$$

Puis nous calculons leur moyenne pondérée pour obtenir la sortie finale  $f$  avec :

$$f = \frac{w_1}{w_1+w_2} * f_1 + \frac{w_2}{w_1+w_2} * f_2 = \bar{w}_1 * f_1 + \bar{w}_2 * f_2 \quad (3.2)$$

Il y'a cinq étapes pour construire le modèle de Sugéno traduites en cinq couches d'un réseau de neurones qui représentera le modèle ANFIS. Chaque case ronde ou carrée dans la **Figure 4.1** représente un neurone. Les couches à neurones carrés possèdent des paramètres optimisables suite à l'entraînement de l'ANFIS sur plusieurs itérations contrairement aux couches à neurones ronds où il ne se produit que des opérations arithmétiques.



**Figure 3.1 :** Représentation du réseau de neurone de l'ANFIS [3]

Considérons le cas de deux entrées floues  $x$  et  $y$  [8] :

- **Première couche :**

D'abord, les deux entrées  $x$  et  $y$  sont traduites en valeurs floues ( $A_1, A_2, B_1$  et  $B_2$  dans la couche 1 de la **Figure 3.1**) Dans cette première couche, les paramètres prémisses des fonctions d'appartenance choisies sont calculés et optimisés suite à l'entraînement du ANFIS.

Les paramètres prémisses sont les constantes qui donnent leur forme aux fonctions d'appartenance.

Exemple : pour une fonction gaussienne, la fonction d'appartenance s'écrit de la forme :

$$U(x) = \exp\left(-\left(\frac{x-c_i}{a_i}\right)^2\right) \quad (3.3)$$

Avec :  $a_i$  et  $c_i$  étant les paramètres prémisses.

Les fonctions peuvent être gaussiennes, sous forme de cloche, trapézoïdales ou même triangulaires.

- **Deuxième couche :**

Dans la deuxième couche, il y a autant de neurones que de règles. La sortie de chaque nœud donne un poids synaptique  $W_i$  de la règle du minimum entre l'une des fonctions d'appartenance de  $x$  avec l'une des fonctions d'appartenance de  $y$ . Comme le montre la **Figure 3.2**, la règle du minimum consiste à prendre la valeur minimale entre les degrés d'appartenance de  $x$  et  $y$  à leurs fonctions respectives.

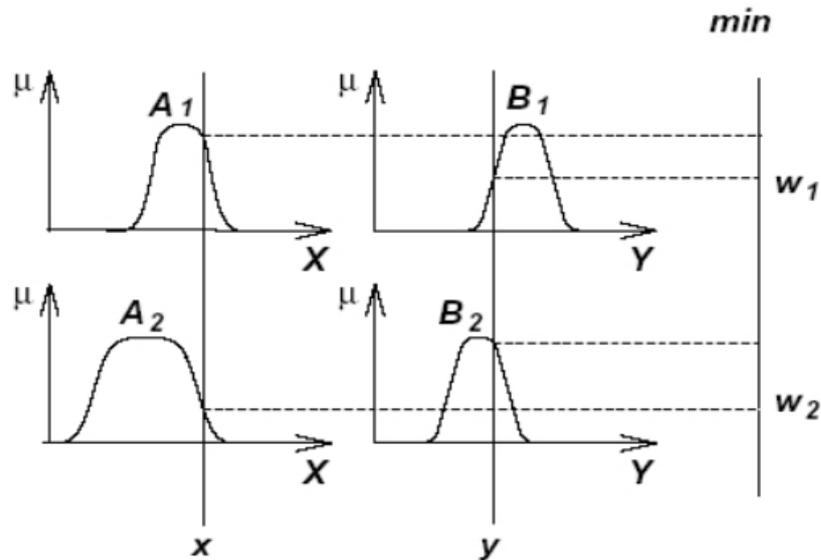


Figure 3.2 : Raisonnement flou avec méthode du minimum

- **Troisième couche :**

Chaque poids obtenu précédemment est divisé par la somme de tous les poids nous donnant ainsi des poids normalisés  $\bar{W}_i$  qui interviendront dans le calcul de la moyenne pondérée.

$$\bar{W}_i = \frac{W_i}{W_i + W_j}$$

- **Quatrième couche :**

Pour chaque neurone, la fonction conséquente est écrite sous sa forme linéaire :

$$f_i = p_i * x + q_i * y + r_i$$

$p_i, q_i$  et  $r_i$  sont les paramètres conséquents qui seront modifiés et optimisés au cours des itérations grâce à l'entraînement du modèle ANFIS. Cette fonction conséquente est ensuite multipliée par son poids normalisé  $\bar{W}_i$ .

$$\bar{W}_i * f_i$$

- **Cinquième couche :**

Finalement, la cinquième couche ne comprend qu'un seul neurone qui est chargé de calculer la moyenne pondérée de toutes les fonctions conséquentes calculées dans la quatrième couche.

$$f = \bar{W}_1 * f_1 + \bar{W}_2 * f_2$$

## 2. 2. Entraînement du modèle ANFIS :

Dans ce travail, l'ANFIS a été entraîné grâce à une méthode hybride alliant la méthode des moindres carrés et la descente du gradient. L'entraînement se déroule en deux étapes pour chaque itération [9].

Premièrement, la propagation avant part du nœud 1 au nœud 5. Lors de cette propagation, les paramètres des prémisses ( $a_i$  et  $c_i$  pour l'équation (4.3)) sont fixés. Les paramètres des conséquents ( $p_i$ ,  $q_i$  et  $r_i$ ) sont optimisés grâce à l'estimation par moindres carrés dans la couche 4. Cela implique de résoudre un système d'équations linéaires pour minimiser l'erreur entre les valeurs de sortie estimées et les valeurs de sortie souhaitées. Les valeurs de sortie souhaitées étant les données d'entraînement.

La deuxième étape est la rétropropagation qui part du nœud 5 au nœud 1. Ici ce sont les paramètres des conséquents qui sont fixés et les paramètres des prémisses qui sont optimisés grâce à la descente de gradient dans la couche 1.

La différence entre les valeurs des sorties estimées et les valeurs des sorties souhaitées est calculée pour déterminer l'erreur, ensuite Les gradients des paramètres des fonctions d'appartenance d'entrée sont calculés en fonction de l'erreur et des dérivées des fonctions d'appartenance. Les paramètres sont finalement mis à jour en utilisant l'algorithme de descente de gradient.

## 3. Conception d'un régulateur ANFIS :

Pour la conception d'un contrôleur ANFIS sur MATLAB, l'un des meilleurs moyens est d'utiliser l'outil « Neuro-Fuzzy Designer » [10].

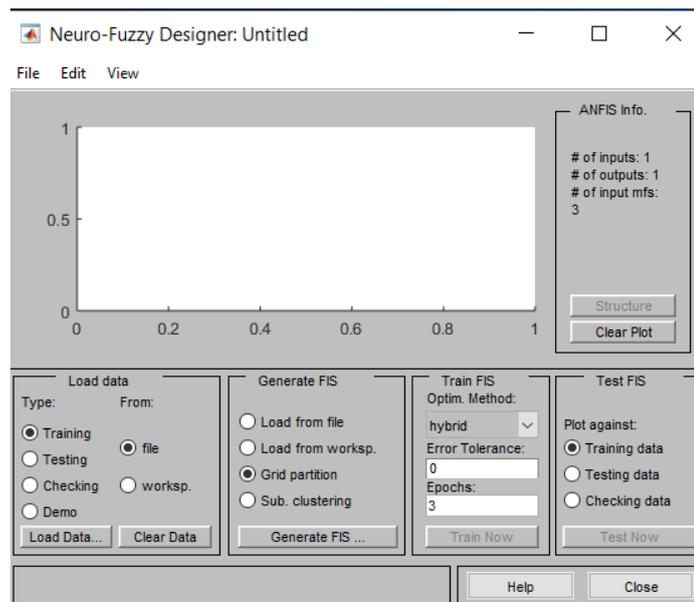
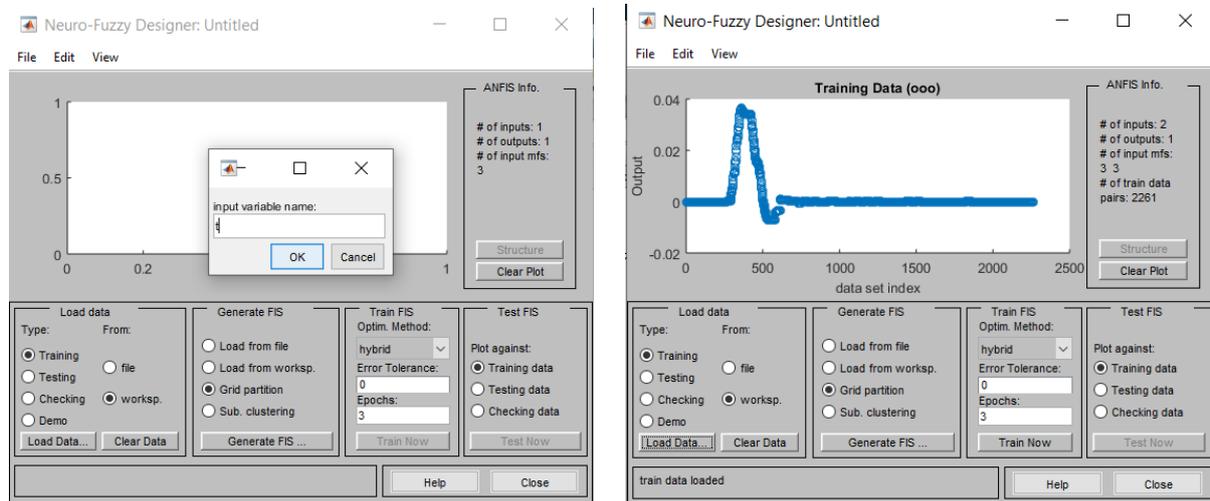


Figure 3.3 : Fenêtre de l'outil « Neuro-Fuzzy Designer »

La conception du contrôleur se déroule en quatre étapes illustrées par les quatre rubriques contenues dans l'application.

- **Load data :**

La première étape est de charger les données d'entraînement dans l'application.



**Figure 3.4 :** Chargement des données d'entraînement

Les données d'entraînement utilisées sont sous la forme d'une matrice. Les premières colonnes représentent les entrées et la dernière colonne représente la sortie.

	1	2	3
1	0	0	-2.5587e-19
2	-6.3981e-75	-6.3981e-75	-2.5587e-19
3	-4.7225e-32	-4.7225e-32	-2.5587e-19
4	9.6673e-32	9.6673e-32	-2.5587e-19
5	1.7734e-33	1.7734e-33	-2.5587e-19
6	-4.1361e-33	-4.1361e-33	-2.5587e-19
7	-3.2348e-32	-3.2348e-32	-2.5587e-19
8	4.2405e-32	4.2405e-32	-2.5587e-19
9	6.6227e-20	6.6227e-20	-2.3998e-19
10	3.1858e-19	3.1858e-19	-1.7941e-19

**Figure 3.5 :** Dix premiers points d'une matrice de donnée

Dans ce travail nous utilisons deux entrées qui sont l'erreur de fréquence et sa dérivée dans le cas « une zone » et l'ACE et sa dérivée dans le cas « 2 zones ou plus ». La sortie, quant à elle, est la puissance de référence.

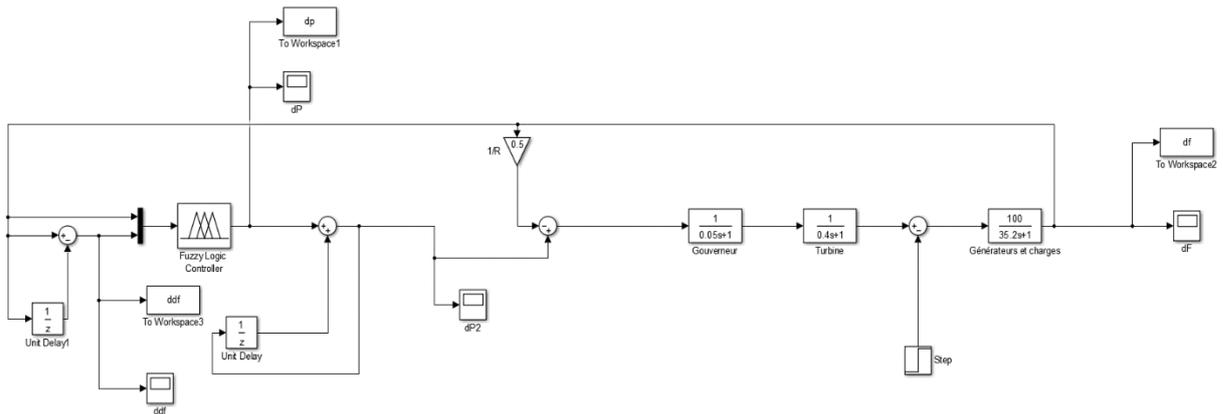


Figure 3.6 : Schéma d'entraînement et de simulation

**Remarque :** les signaux d'entrées doivent être directement recueillis à l'entrée du régulateur et le signal de sortie directement à la sortie du régulateur pour ne pas fausser les résultats.

- **Generate FIS :**

Après avoir chargé les données, on génère le FIS (Fuzzy Inference System) afin de générer l'itération 0 du modèle ANFIS [11].

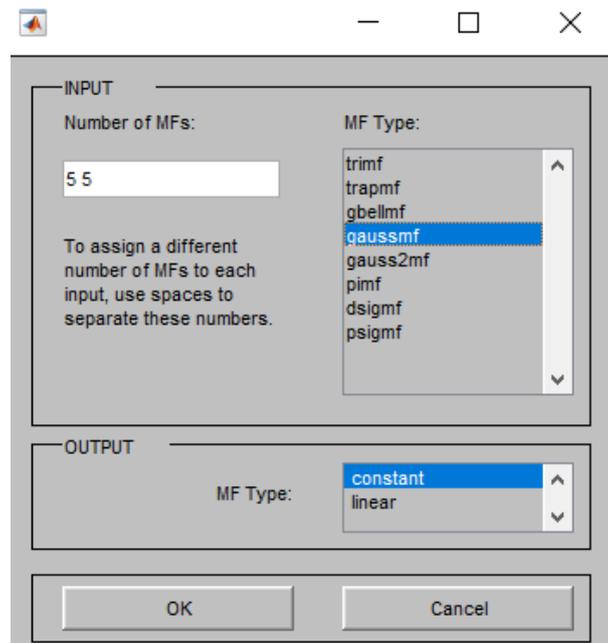


Figure 3.7 : Choix relatifs aux fonctions d'appartenance

Comme montré dans la **Figure 3.7**, nous pouvons choisir le nombre de fonctions d'appartenance pour chaque variable d'entrée ainsi que leur type. Dans le cas de l'AGC, les types de fonctions d'appartenance avec les meilleurs résultats, selon nos tests, sont les fonctions gaussiennes « gaussmf » et les fonctions cloches « gbellmf ».

Ceci est dû à :

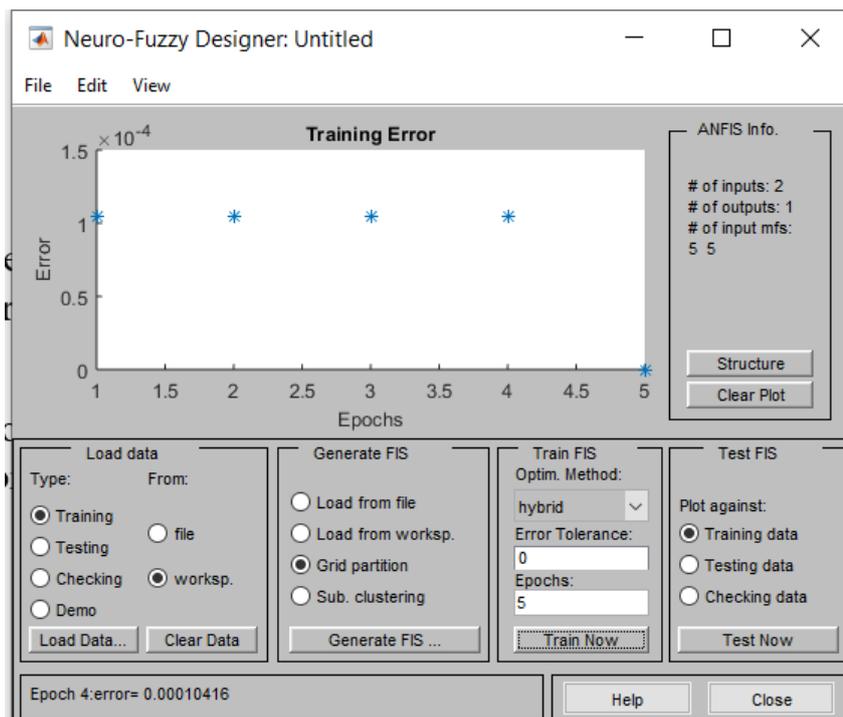
- Leur souplesse : Courbes lisses et continues.
- Leur flexibilité : Ajustement de la forme très malléable grâce aux paramètres prémisses.
- Leur localisation : Les 2 fonctions ont des pics et des centres localisés contrairement aux trapézoïdes par exemple ce qui leur permet de se concentrer sur des plages d'entrées spécifiques.

Comment choisir le nombre de fonctions et leur type :

- Pour le type : il faut essayer les différentes fonctions et voir laquelle donne les meilleurs résultats.
- Pour le nombre : il faut prendre le plus petit nombre de fonctions pour lequel l'erreur atteint sa limite.

- **Entraînement :**

Nous pouvons préciser les conditions d'arrêt de l'entraînement grâce à une tolérance d'erreur minimale ou à un nombre d'itérations maximal.

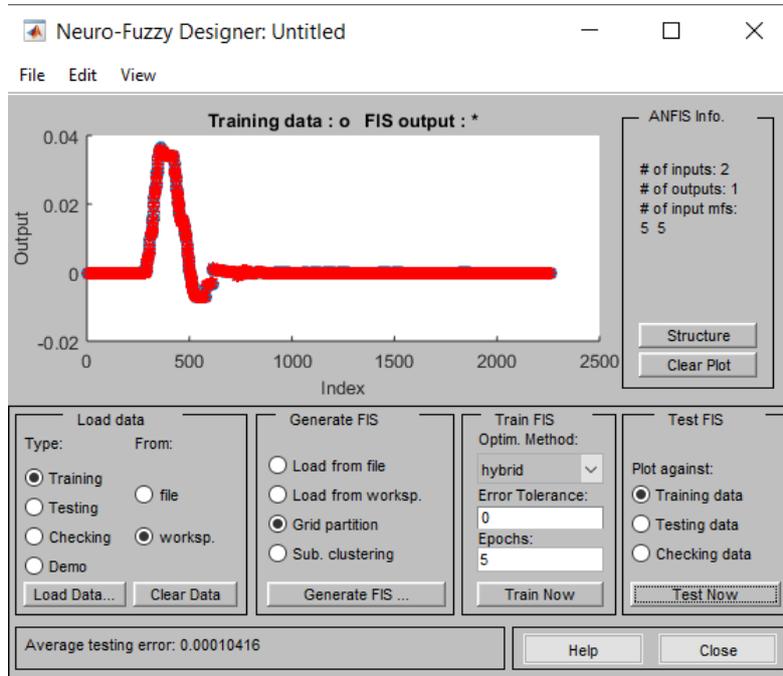


**Figure 3.8 :** Phase d'entraînement du modèle ANFIS

A la fin de l'entraînement, la version finale du ANFIS est enfin générée.

### - Test FIS :

Une fois l'entraînement terminé, on teste le modèle ANFIS généré avec un autre nombre de données connues, le but étant que les résultats du ANFIS (en rouge) se superposent au maximum sur les données de test (en bleu).



**Figure 3.9** : Phase de test du modèle ANFIS

Nous avons dorénavant un régulateur ANFIS fonctionnel, il suffit juste de l'exporter sur Workspace pour pouvoir l'utiliser directement sur Simulink.

## **4. Application du ANFIS sur 1 zone :**

Le schéma Simulink utilisé est le même que celui dans le cas de la régulation par logique floue. La fonction Simulink du régulateur reste la même, il suffit juste d'y introduire l'ANFIS à partir du Workspace (voir **Figure 2.10**).

La question qui se pose maintenant est la suivante : Comment obtenir les meilleures données d'entraînement ?

Pour cela, il faut comparer les résultats des différents régulateurs flous. En considérant les deux régulateurs « flou1 » et « flou2 » présentés dans le chapitre 3, le régulateur « flou1 » donne des résultats non convaincants pour tous les points de fonctionnement tandis que le régulateur « flou2 » offre de très bonnes performances dans le cas où  $dP=1.5\%$ .

Il est possible d'appliquer ces résultats sur les autres valeurs de déviations de puissance grâce au ANFIS.

Toutes les opérations du schéma Simulink sont linéaires, il est donc possible d'obtenir des données d'entraînement d'un point quelconque à partir du point  $dP=1.5\%$  en résolvant une règle de trois :

$T_1$ (Données du point  $dP=0.5\%$ )  $\rightarrow 0.5$

$T_2$ (Données du point  $dP=1.5\%$ )  $\rightarrow 1.5$

ce qui donne :

$$T_1 = \frac{T_2 \cdot 0.5}{1.5}$$

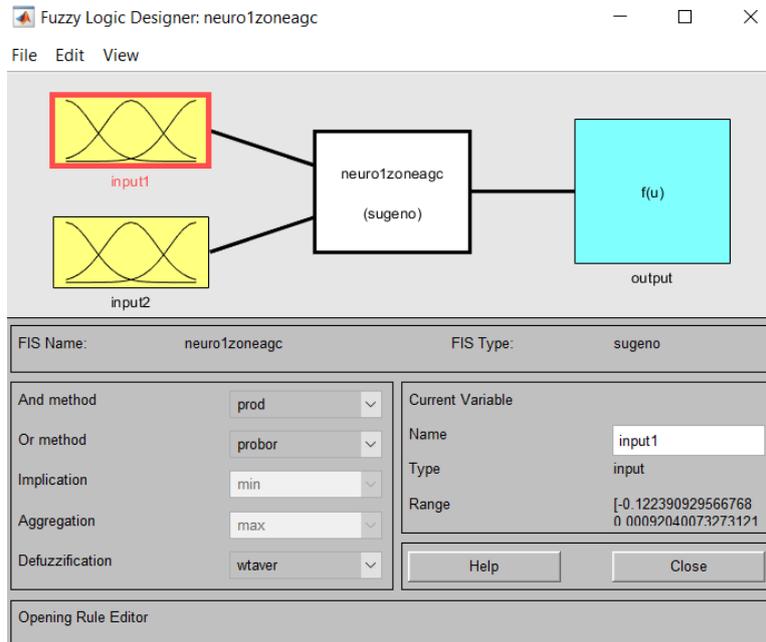
La nouvelle matrice de données  $T_1$  nous donnera un ANFIS qui reproduira le comportement du cas  $dP=1.5\%$  pour le cas  $dP=0.5\%$ .

En appliquant cette méthode sur plusieurs points équidistants (ici de 0.5% à 10.05% avec un pas de 0.5%) on peut entrainer un seul régulateur ANFIS pour se comporter de manière optimale sur toute la plage de données.

Le code MATLAB pour obtenir une matrice d'entrainement corrigée pour le cas de  $dP=1.5\%$  s'écrit de la manière suivante :

```
n=10*1.5;
t=[df ddf dp];
T=[t*5/n;t*10/n;t*15/n;t*20/n;t*25/n;t*30/n;t*35/n;t*40/n;t*45/n;t*50/n;t*55/n;t*60/n;t*65/n;t*70/n;t*75/n;t*80/n;t*85/n;t*90/n;t*95/n;t*100/n;t*105/n];
```

L'entrainement nous donnera un régulateur flou de type Sugéno avec des fonctions d'appartenance, des règles et des fonctions conséquentes générées et optimisées par réseaux de neurones.



**Figure 3.10 :** Fenêtre principale du régulateur ANFIS

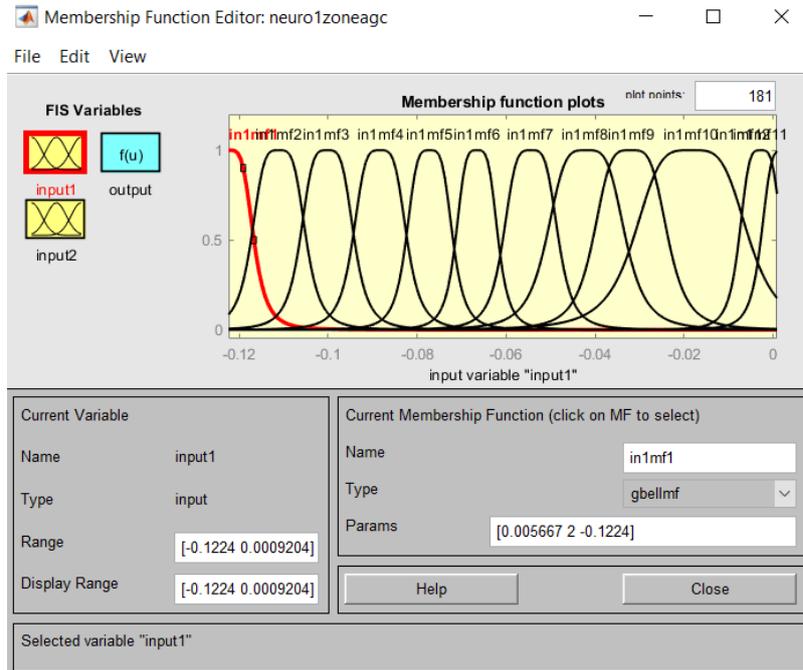


Figure 3.11 : Fonctions d'appartenance de  $\Delta f$  du régulateur ANFIS

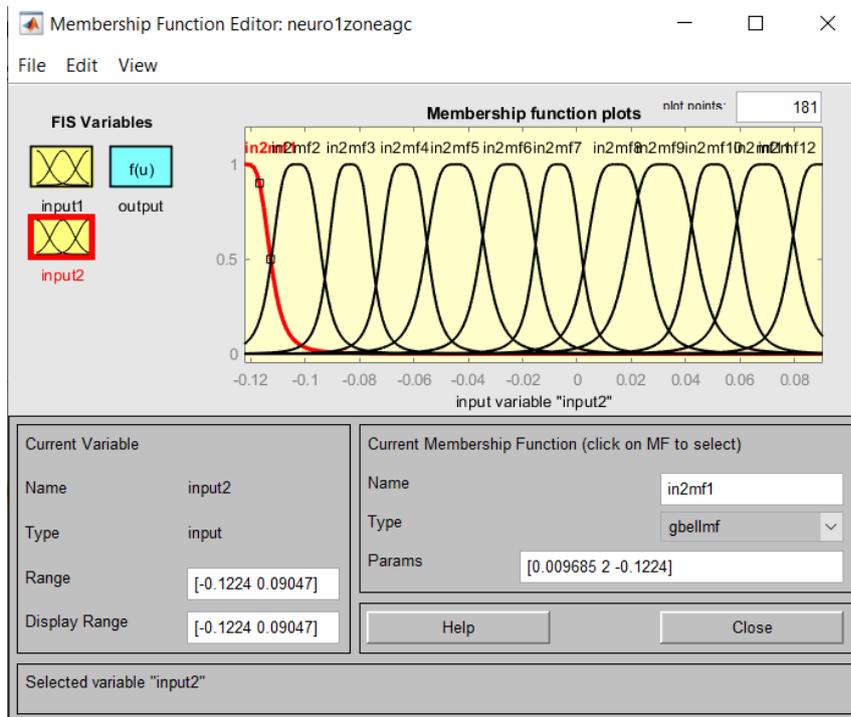


Figure 3.12 : Fonctions d'appartenance de  $\Delta \dot{f}$  du régulateur ANFIS

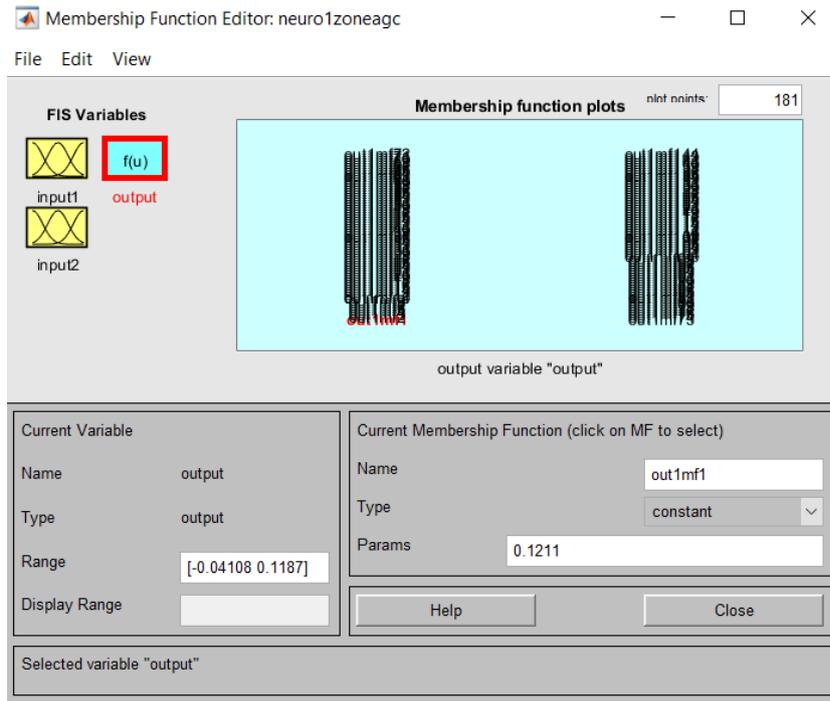


Figure 3.13 : Fonctions conséquentes de la variable de sortie  $\Delta P_{ref}$  du régulateur ANFIS

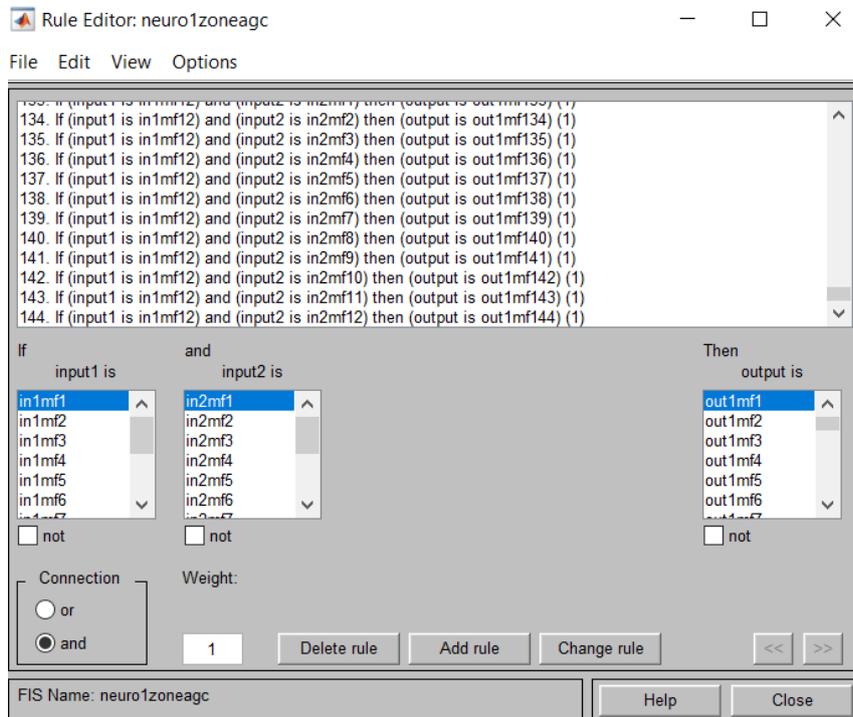
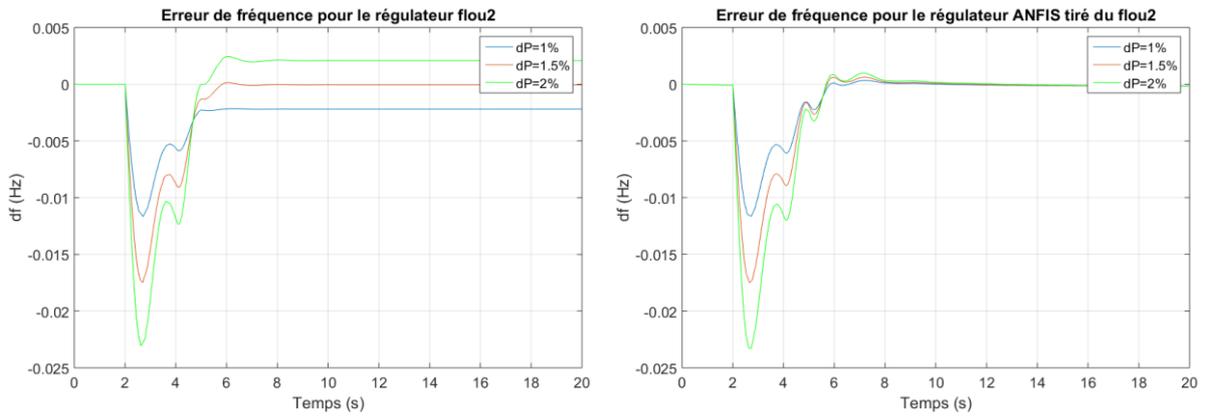


Figure 3.14 : Règles d'inférence du régulateur ANFIS

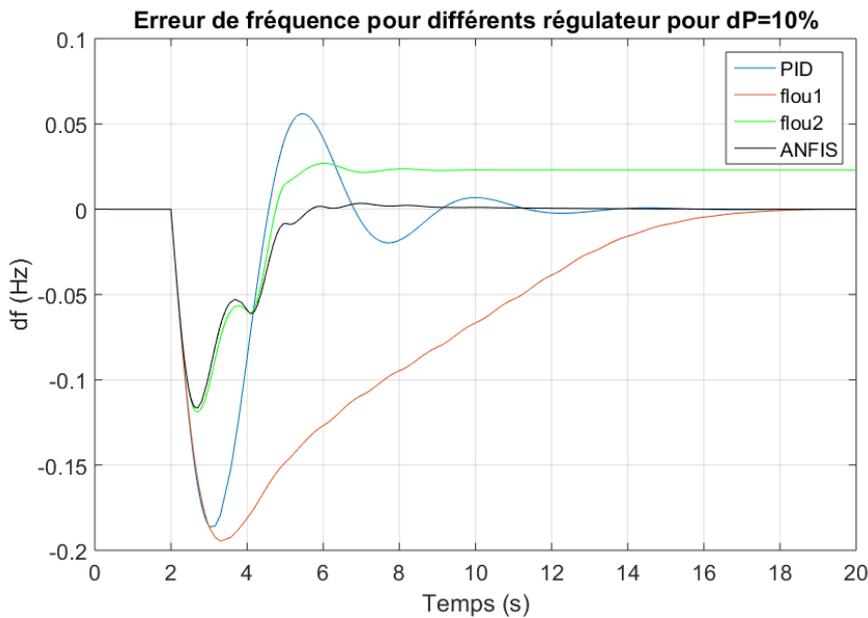
Les résultats du régulateur ANFIS sont donnés dans la **Figure 4.15** :



**Figure 3.15** : Comparaison des réponses du système pour un régulateur flou et un régulateur ANFIS

On voit bien que l'erreur de fréquence et la vitesse de montée sont similaires pour les deux méthodes, mais la convergence devient assurée dans le cas de l'ANFIS.

La figure ci-dessous représente l'erreur de fréquence pour un  $dP=10\%$  pour les différents régulateurs vus jusqu'ici :



**Figure 3.16** : Réponse du système pour les différents régulateurs

Contrairement à d'autres régulateurs comme le flou1, le temps de convergence reste fixe même en augmentant la déviation de puissance ( $dP$ ) dans le cas de l'ANFIS. Nous remarquons une convergence à 6s après une perturbation à 2s soit un temps de réglage de 4s seulement contrairement aux 8s du PID et aux 16s du flou2. Le régulateur flou1 converge quant à lui vers une valeur non nulle comme vu dans les exemples précédents.

L'ANFIS nous permet donc de profiter d'un régulateur flou défectueux ou non optimal avec ne serait-ce qu'un point de fonctionnement utile en généralisant ce cas aux autres points de fonctionnement.

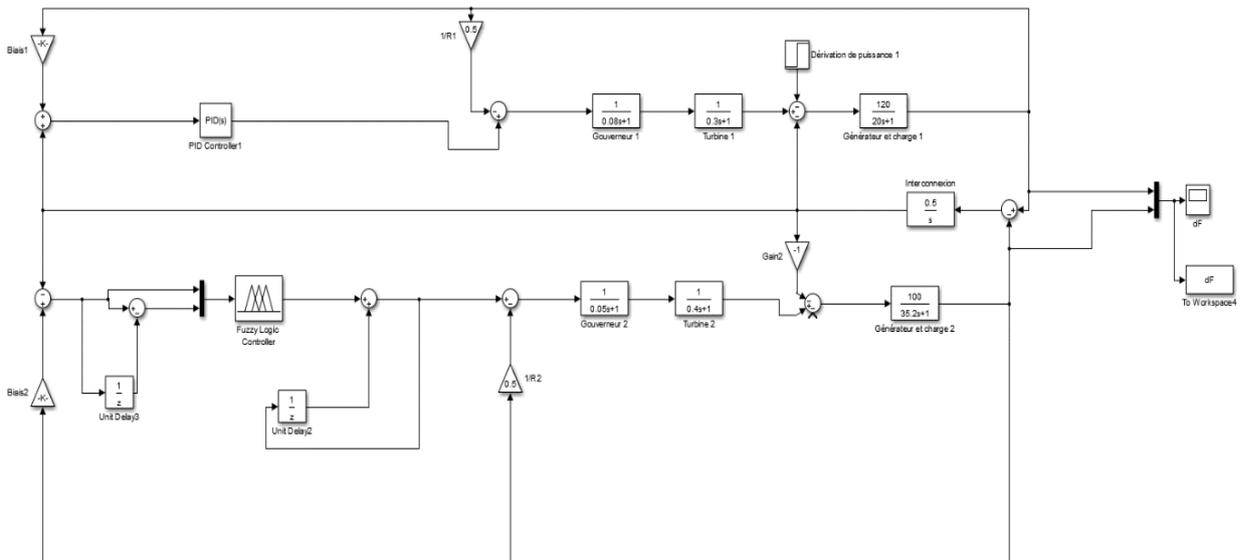
**Remarque :** Il est aussi important de mentionner que le temps d'exécution des schémas Simulink est plus rapide dans le cas de l'utilisation des régulateurs ANFIS comparé aux régulateurs flous.

Pour l'exemple précédent : la moyenne du temps d'exécution du ANFIS pour 5 simulations est de 2.726 secondes contre 3.840 secondes pour le régulateur flou dans les mêmes conditions. L'ANFIS offre une exécution du programme plus rapide d'une seconde, ce qui est important pour ce type de calcul qui doit se faire de manière instantanée. Cette différence de temps est d'autant plus importante pour des systèmes plus complexes (2 zones, 3 zones...).

## 5. Application du ANFIS sur 2 zones :

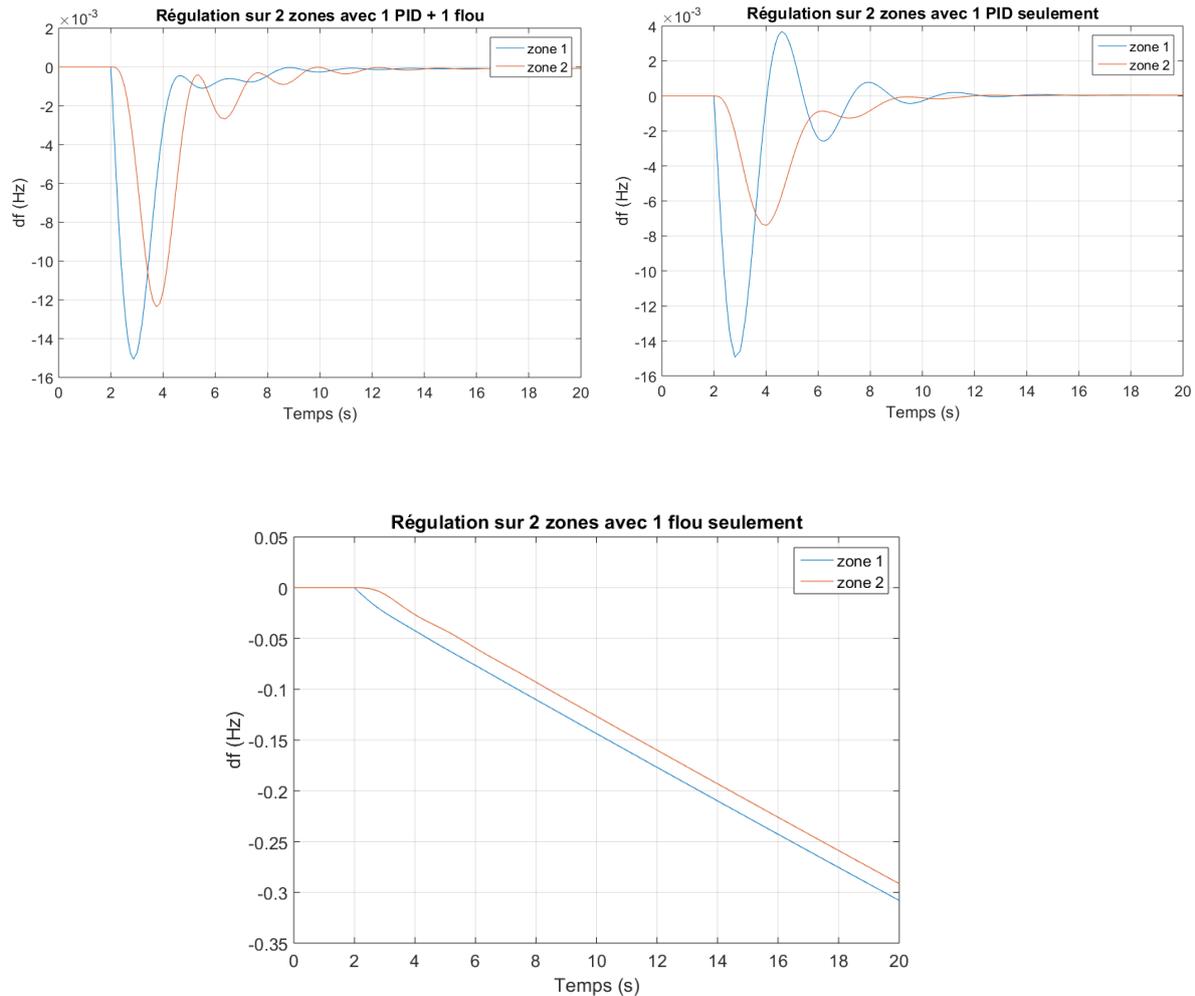
### 5.1. Un régulateur PID (zone 1) et un régulateur ANFIS (zone 2) :

Dans plusieurs travaux, le régulateur intelligent est posé au niveau d'une zone non perturbée et le PID au niveau de la zone perturbée.



**Figure 3.17 :** Schéma à deux zones avec régulation ANFIS dans la zone non perturbée

Ce cas de figure n'est pas optimal car le régulateur situé dans la zone perturbée sera celui qui aura le plus gros impact sur la régulation du réseau.



**Figure 3.18 :** Réponses du système pour une perturbation sur la zone 1

Nous pouvons voir à travers la **Figure 3.18** que le régulateur intelligent ne peut pas réguler la fréquence s'il est mis seul dans la zone non perturbée, et que son impact est beaucoup moins important que celui du PID se trouvant dans la zone perturbée.

Une autre problématique serait l'inefficacité du ANFIS s'il venait à être entraîné sur la base d'une zone non perturbée. Le régulateur ne pourra pas fonctionner lors de l'apparition d'une perturbation car l'ANFIS n'est pas orienté vers l'extrapolation des données, il ne sera performant que sur la plage d'entraînement sur laquelle il aura été bâti.

Le meilleur moyen de construire un régulateur ANFIS performant est donc d'appliquer des perturbations de puissance sur les 2 zones du réseau et de traiter les 3 cas possibles qui sont : perturbation dans la zone 1, perturbation dans la zone 2 et perturbation dans les 2 zones en même temps.

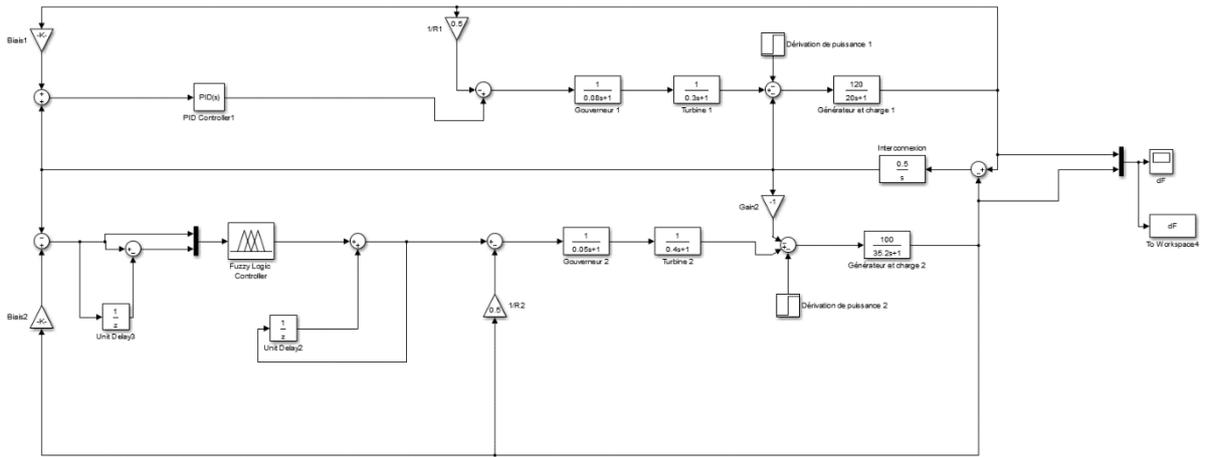


Figure 3.19 : Schéma à deux zones avec les deux zones perturbées

Comme pour le cas mono-zone, nous pouvons tester des régulateurs flous pour voir leur impact. Ici, la régulation va se faire grâce à un PID et à un ANFIS.

En fixant les paramètres du PID pour l'une des zones et en testant les régulateurs flous flou1 et flou2 nous remarquons des résultats similaires au cas mono-zone avec le régulateur flou1 nous donnant une convergence garantie mais très lente et le régulateur flou2 nous donnant une convergence très rapide vers une valeur non nulle pour quasi tous les points de fonctionnement, comme montré sur la Figure 3.20 en prenant des variations de puissance aléatoires ( $dP_1 = 5\%$  et  $dP_2 = 8\%$ ).

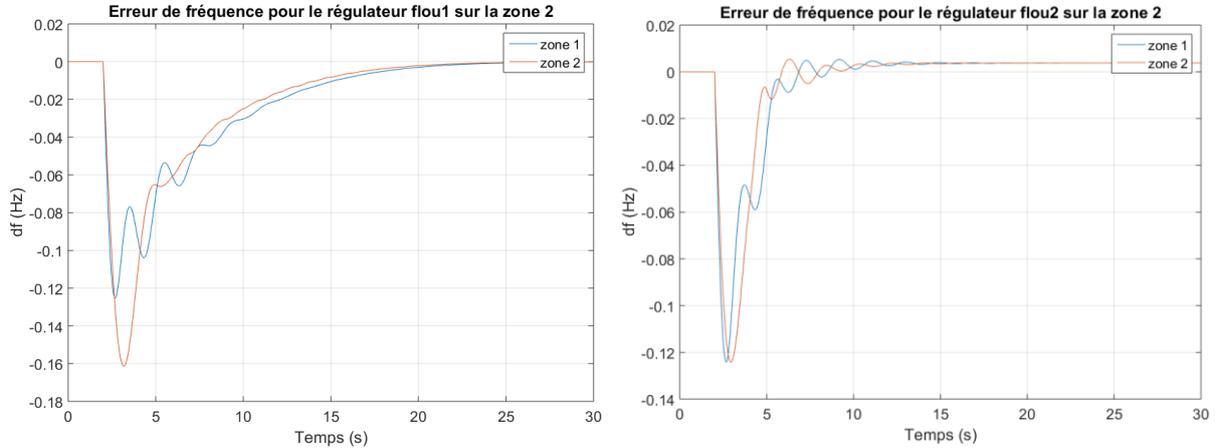


Figure 3.20 : Réponse du système pour les régulateurs flous sur la zone 2

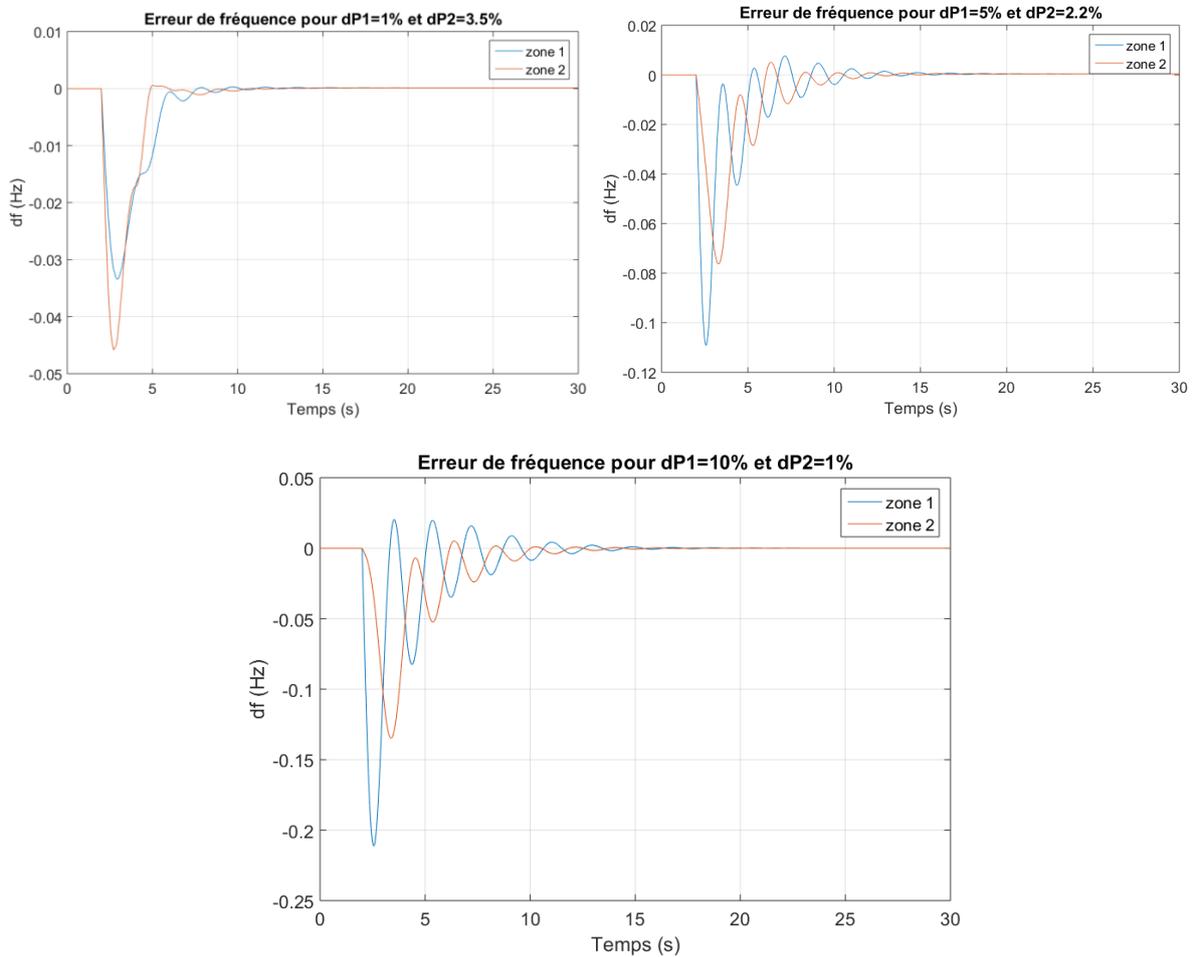
Il revient comme pour le cas mono-zone à chercher le ou les meilleurs points de fonctionnement du régulateur flou2 pour entrainer l'ANFIS.

Dans cet exemple, le point de fonctionnement optimal du régulateur flou2 (la valeur de la variation de puissance de ( $dP_2$ ) pour laquelle le réglage converge vers zéro) varie avec la variation de puissance de la zone 1 ( $dP_1$ ).

Exemple : Pour  $dP_1=1\%$  le point de fonctionnement de flou2 est à  $dP_2=3.5\%$

Pour  $dP_1=5\%$  le point de fonctionnement de flou2 est à  $dP_2=2.2\%$

Pour  $dP_1=10\%$  le point de fonctionnement de flou2 est à  $dP_2=1\%$



**Figure 3.21 :** Réponse du système pour le régulateur flou2 dans la zone 2

**Remarque :** la variation de la fréquence a un comportement beaucoup plus oscillatoire lorsque le régulateur PID de la zone 1 est plus sollicité que le régulateur intelligent. C'est le cas pour une variation de puissance  $dP_1$  plus importante que la variation de puissance  $dP_2$ .

Les données d'entraînement du ANFIS sont recoltées à partir des résultats du régulateur flou2 corrigés (usage de la règle de trois comme montré dans le cas mono-zone) sur une plage suffisante de variation de puissance  $dP_2$  (Ex : De  $dP_2=0.5\%$  à  $dP_2=10.05\%$ ) pour différents points de variation de puissance  $dP_1$  (Prendre 2 ou 3 points suffisent [ $dP_1 = 1\%$ ,  $5\%$  et  $10\%$ ] car la variation des points de  $dP_2$  de fonctionnement est assez faible).

Pour faire simple, il revient à exécuter les mêmes démarches que pour le cas mono-zone trois fois, une fois pour chaque point  $dP_1$  et à mettre toutes ces données dans une seule et même matrice d'entraînement.

La figure suivante représente les 3 plages des valeurs d'entrainement :

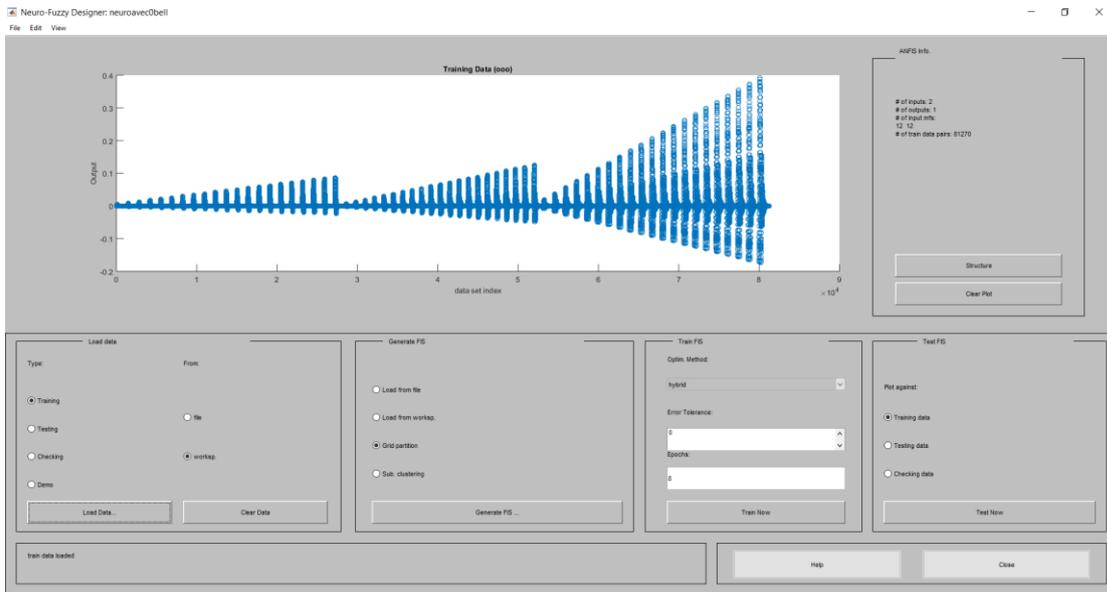


Figure 3.22 : Données d'entrainement du régulateur ANFIS

Après usage du « Neuro-fuzzy designer », nous obtenons un ANFIS à 12 fonctions en cloche pour chacune des 2 entrées qui pourra répliquer les résultats des données d'entrainements corrigées.

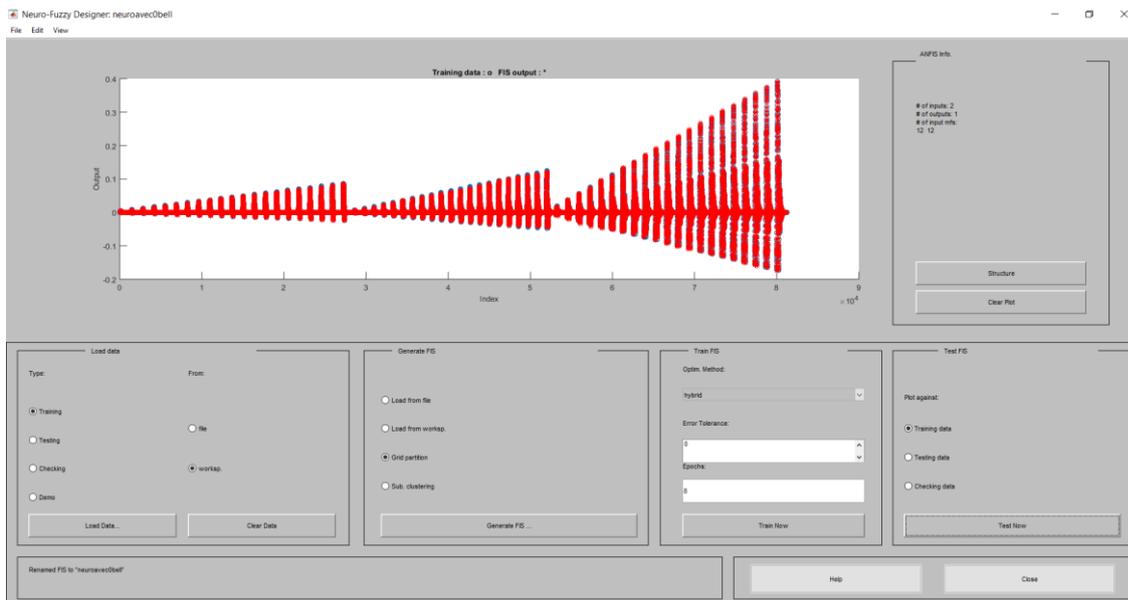


Figure 3.23 : Résultats des tests du régulateur ANFIS

Exemples d'application du régulateur ANFIS obtenu :

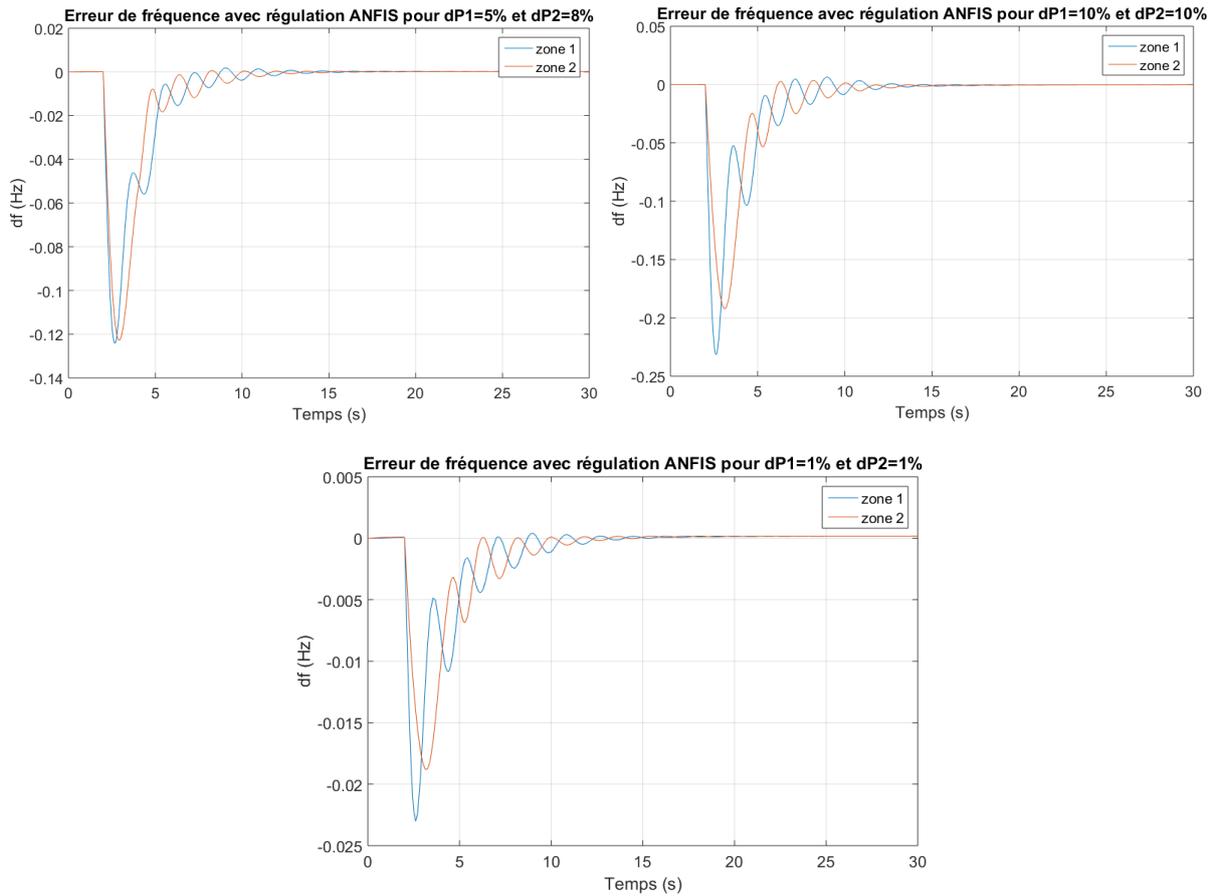


Figure 3.24 : Réponse du système pour le régulateur ANFIS dans la zone 2

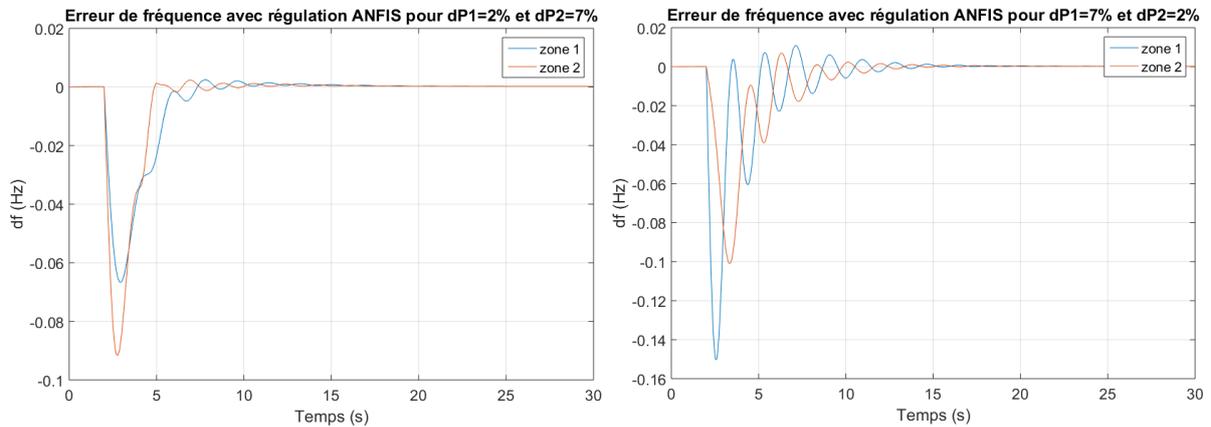


Figure 3.25 : Réponse du système pour le régulateur ANFIS dans la zone 2 en échangeant  $dP_1$  avec  $dP_2$

De la **Figure 3.24** nous remarquons que le régulateur ANFIS est fonctionnel dans différents cas de figures, avec des variations de puissance allant de 1% à 10%. Avec néanmoins d'assez fortes oscillations.

Les oscillations sont dues majoritairement à l'effet du régulateur PID. C'est d'autant plus flagrant en étudiant les résultats de la **Figure 3.25**.

Dans cette figure, nous avons d'abord appliqué une forte déviation de puissance au niveau de la zone 2 avec une faible déviation pour la zone 1. Nous avons ensuite inversé les valeurs des déviations de puissance (Soit  $dp_1=2\%$  et  $dp_2=7\%$  ensuite  $dp_1=7\%$  et  $dp_2=2\%$ ).

Nous remarquons que la régulation est bien meilleure lorsque le régulateur ANFIS régule la zone à forte déviation de puissance avec des oscillations quasi nulles, prouvant ainsi que le PID est responsable des oscillations.

### 5.2. Deux régulateurs ANFIS sur la zone 1 et la zone 2 :

Nous allons remplacer le régulateur PID par un régulateur intelligent pour comparer les 2 méthodes de régulation.

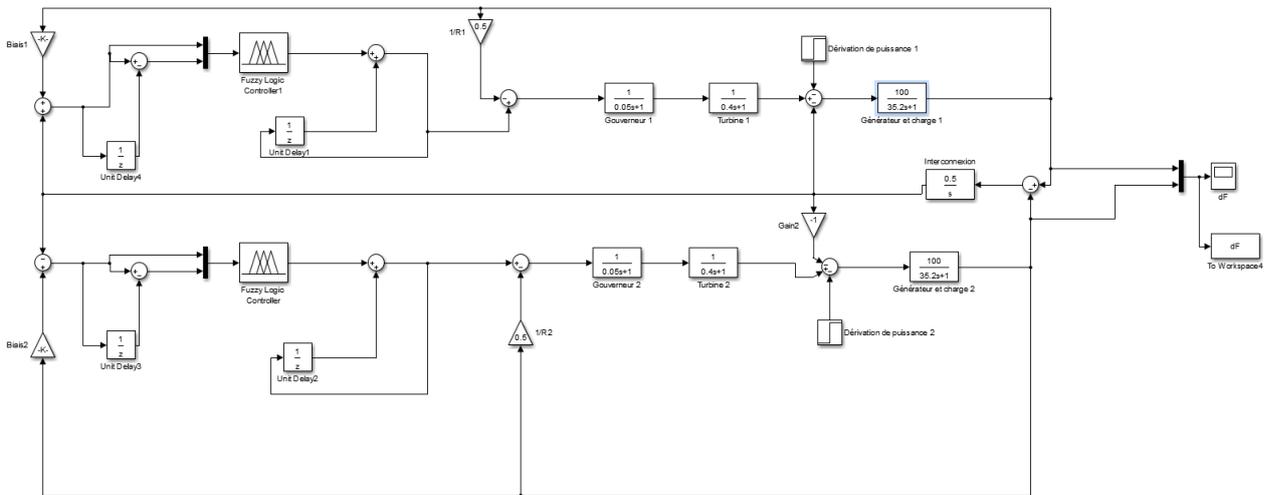


Figure 3.26 : Schéma à deux zones avec deux régulateurs ANFIS

En utilisant le régulateur ANFIS entraîné à partir du flou2 sur les 2 zones, nous obtenons les résultats suivants ( $dp_1=6\%$  et  $dp_2=8\%$ ) :

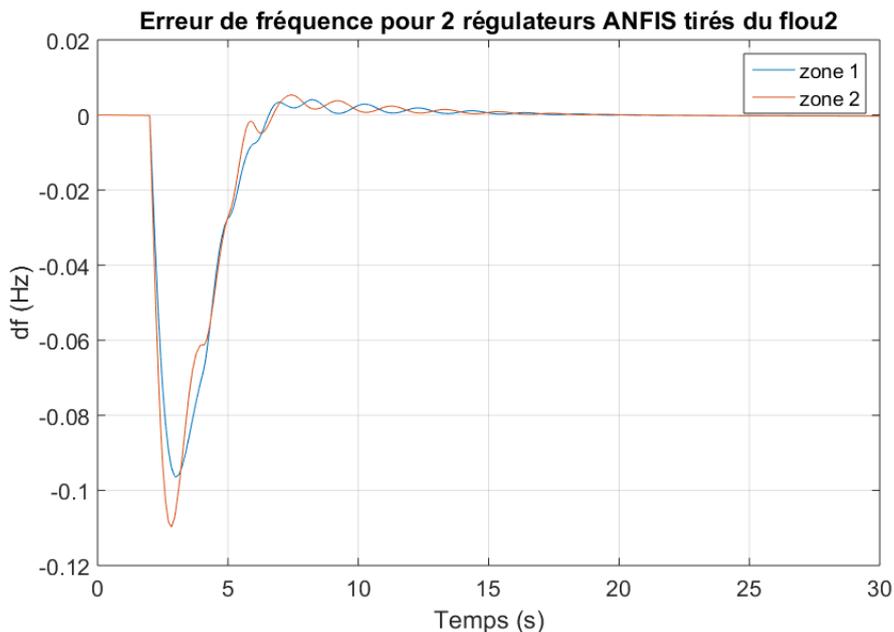
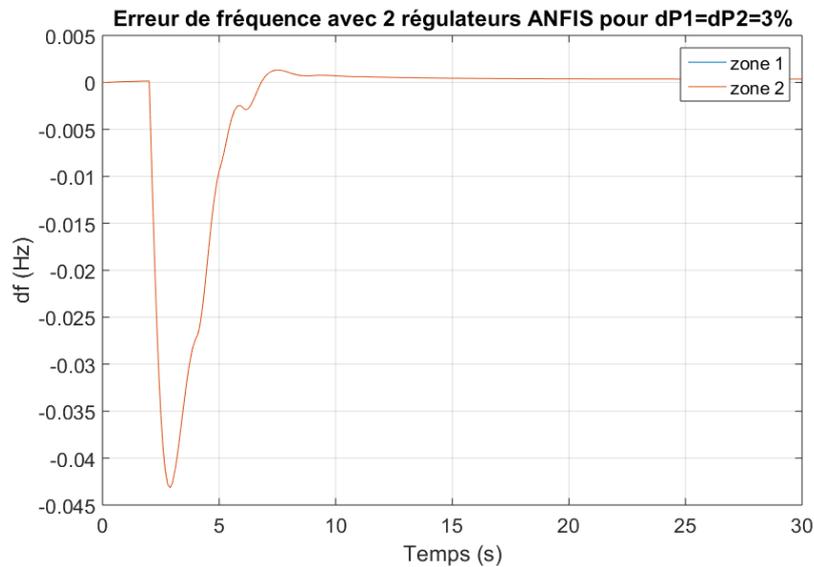


Figure 3.27 : Réponse du système pour 2 régulateurs ANFIS identiques

La régulation est belle et bien présente avec une chute de fréquence basse mais le temps de convergence n'est pas assez bon pour une méthode utilisant 2 régulateurs neuro-flous. Contrairement au cas mono-zone où le régulateur ANFIS donne une convergence à partir de  $t=6s$ , ici la régulation est retardée par de petites oscillations qui persistent sur plusieurs secondes.

Ceci est dû à un manque de coordination entre les 2 régulateurs, résultant ainsi en des actions de contrôle parfois contradictoires. Chaque régulateur répond différemment à sa variation de puissance causant un comportement oscillatoire lorsque les 2 signaux de contrôle interagissent.

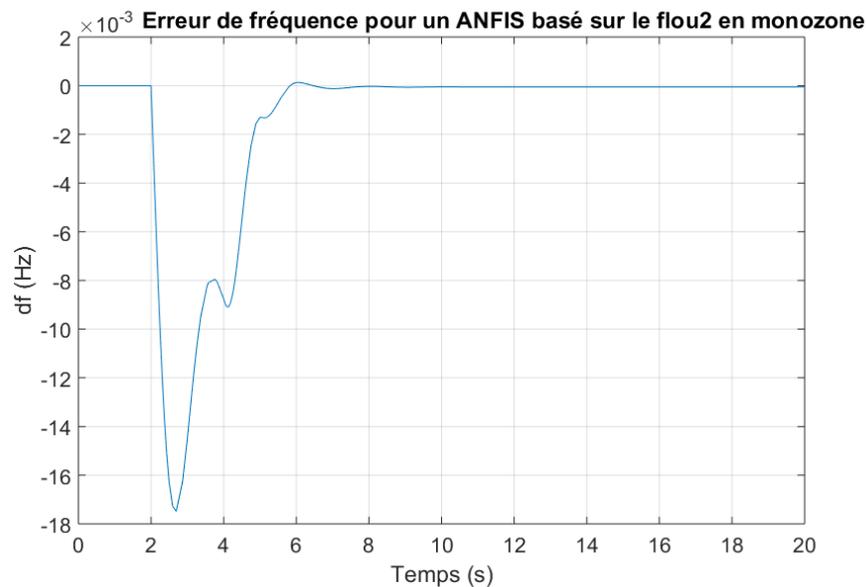
Nous pouvons mieux comprendre l'effet du manque de coordination précédent, lorsque nous prenons des variations de puissance égales dans les 2 zones.



**Figure 3.28** : Réponse du système pour 2 régulateurs ANFIS identiques avec  $dP_1 = dP_2$

Dans ce cas où les perturbations de charge sont identiques dans les 2 zones, les 2 figures sont superposées car les 2 régulateurs ANFIS/flou identiques auront exactement les mêmes actions de contrôle dû au fait de réguler les mêmes cas de variation de puissance ( $dP_1 = dP_2$ ). Nous pouvons supposer que le fait d'avoir 2 variations de puissance différentes, ce qui plus probable en pratique, qui nous donne des actions de contrôle différentes et contradictoires entre les 2 régulateurs, entraînant ainsi des oscillations.

Ceci est d'autant plus vrai lorsque les régulateurs n'ont pas une évolution complètement lisse en temps normal. L'hésitation et les petites erreurs du régulateur se verront amplifiées lors de leur couplement comme montré dans la **Figure 3.29** :

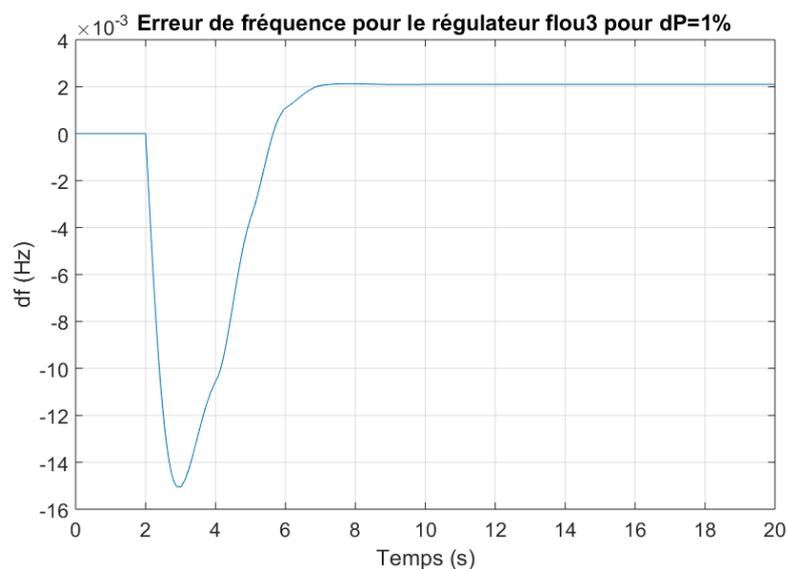


**Figure 3.29 :** Réponse du système pour 1 régulateur ANFIS en mono-zone

De plus, contrairement au cas où il y a un régulateur PID, nous ne pouvons pas fixer le comportement de l'un des régulateurs intelligents pour essayer d'optimiser le second car leur comportement dynamique est différent pour chaque point de fonctionnement.

Un moyen de réduire ces oscillations serait d'implémenter un ANFIS basé sur un autre régulateur flou avec un comportement plus lisse en plus du ANFIS basé sur le régulateur flou2. Pour ceci, un troisième régulateur flou (flou3) sera utilisé.

Le régulateur flou3 a une chute de fréquence acceptable et un temps de convergence assez rapide mais vers des valeurs non nulles pour de faibles variations de puissance  $dP$ .



**Figure 3.30 :** Réponse du système pour le régulateur flou3 en mono-zone

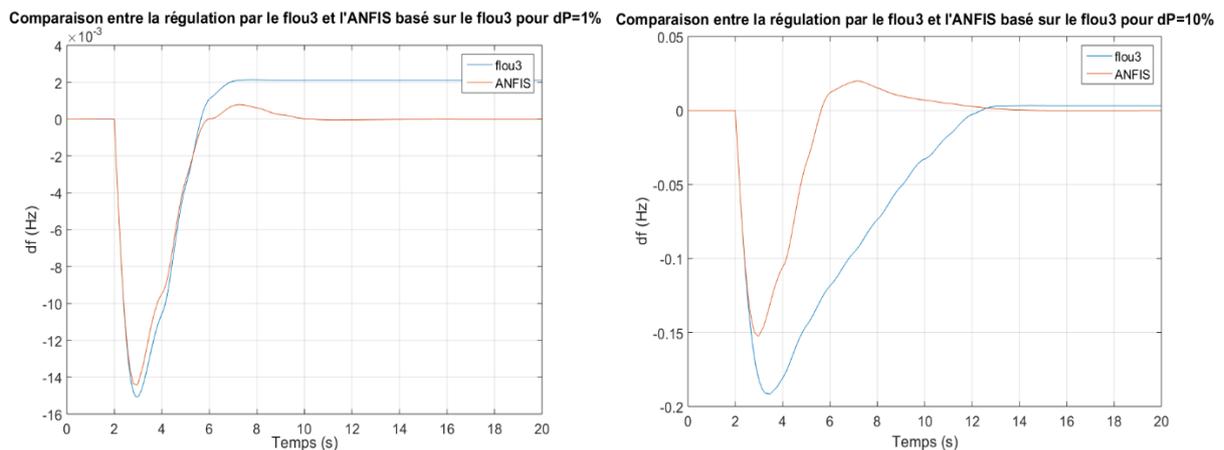
Nous pouvons utiliser ce régulateur pour entrainer un ANFIS après avoir traité ses données.

Premièrement, on remplace les valeurs de convergence non nulles qui sont de l'ordre de 0.002 par 0 dans la matrice des données pour forcer l'ANFIS à créer un régulateur qui convergera vers 0.

Ensuite, on utilise le point de fonctionnement de  $dp=1\%$  pour l'entraînement sur toute la plage de données grâce à la règle de trois (comme montré précédemment) pour préserver la vitesse de convergence qui se perd lors de l'augmentation de la variation de puissance.

**Remarque :** forcer les données d'entraînement convergentes, non nulles vers 0 et entrainer des points de fonctionnement à partir d'autres points grâce à la règle de trois sont des avantages que nous procure l'ANFIS, mais ça ne veut pas dire que toute opération algébrique sur les données d'entraînement est possible.

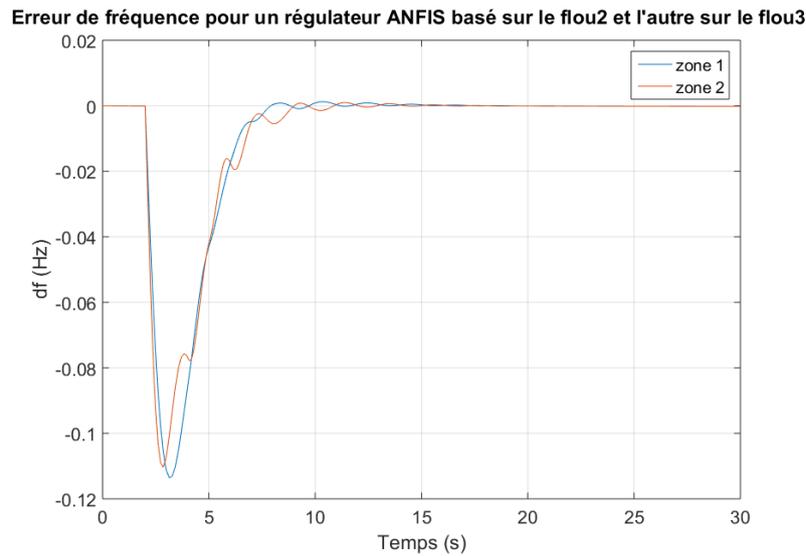
Nous obtenons le résultat illustré dans la figure ci-dessous :



**Figure 3.31 :** Réponse du système pour le régulateur ANFIS tiré du flou3 en mono-zone

L'ANFIS procure un temps de convergence réduit et une réponse complètement nulle, surtout pour les grandes valeurs de déviation de puissance.

En implémentant le nouvel ANFIS dans la zone 1, nous obtenons la réponse suivante :



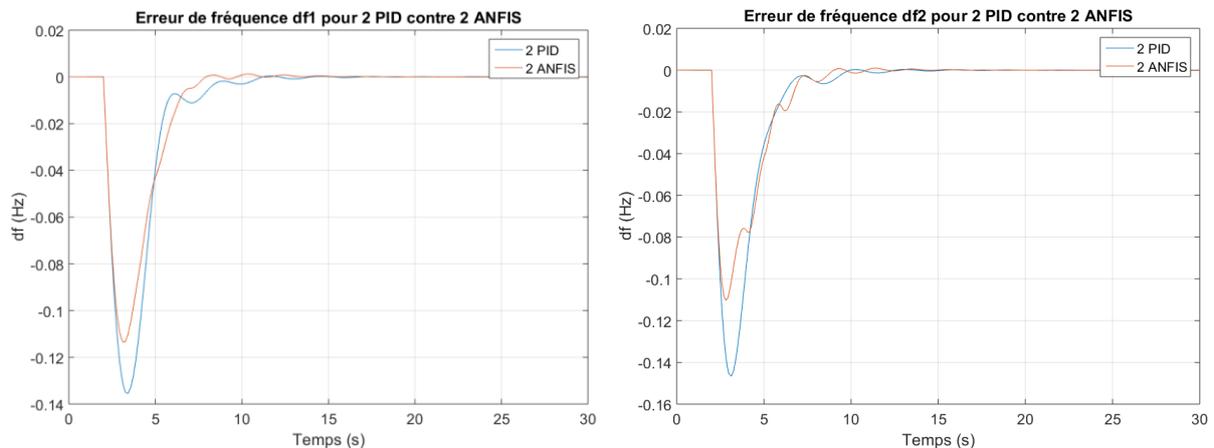
**Figure 3.32 :** Réponse du système pour 2 régulateurs ANFIS différents

Par rapport au cas utilisant 2 ANFIS similaires, nous remarquons une baisse des oscillations résultant en un temps de convergence plus rapide.

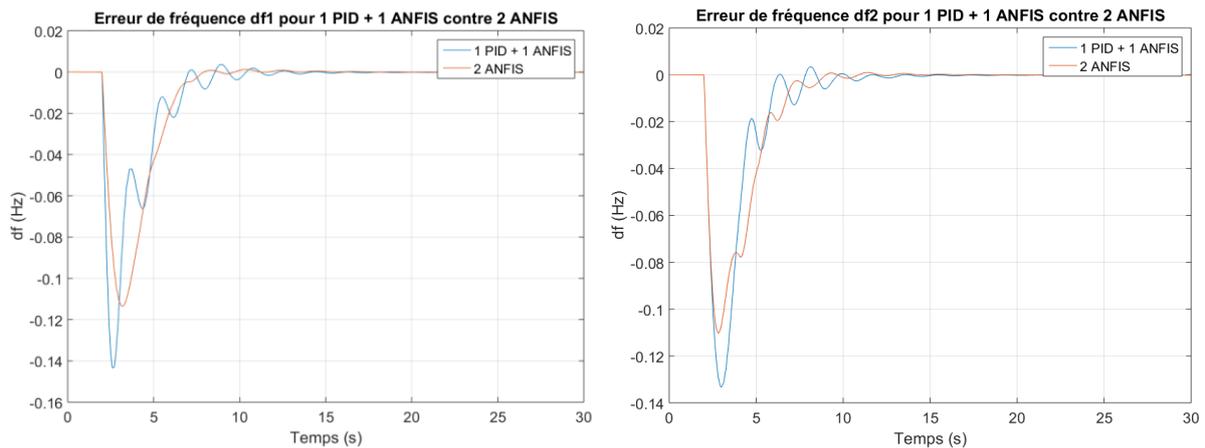
Les performances des 2 régulateurs couplés restent néanmoins inférieures à leurs compétences individuelles à cause de leur manque de coordination. Il faut donc bien choisir les régulateurs neuro-flous pour les coupler efficacement.

### **5. 3. Comparaison des méthodes utilisées sur 2 zones :**

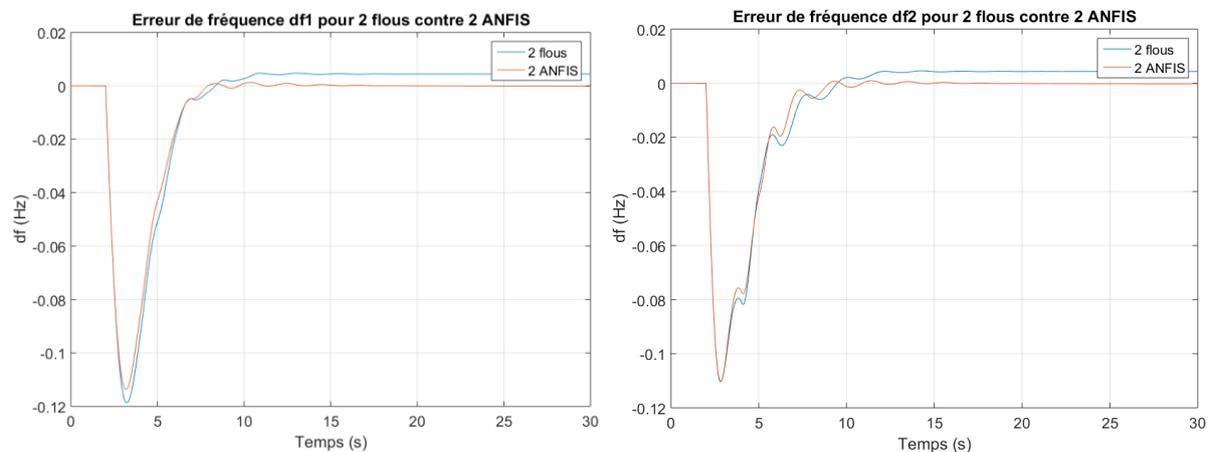
Comparaison de l'utilisation des 2 ANFIS différents avec les autres méthodes vues auparavant pour  $dP_1 = 6\%$  et  $dP_2 = 8\%$ .



**Figure 3.33 :** Comparaison entre 2 PID et 2 ANFIS



**Figure 3.34 :** Comparaison entre 1 PID + 1 ANFIS et 2 ANFIS



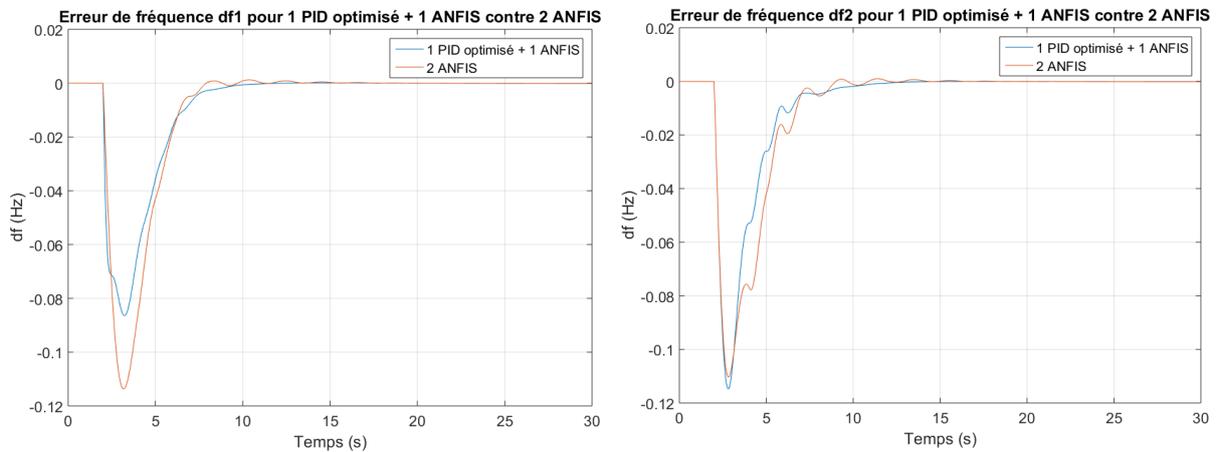
**Figure 3.35 :** Comparaison entre 2 régulateurs flous et 2 ANFIS

- **Comparaison entre 2 PID et 2 ANFIS :** selon la **Figure 3.33**, pour le cas des 2 ANFIS, la chute de fréquence est réduite de 0.02 à 0.03 Hz sur les 2 zones et la convergence est plus rapide (de 3 secondes sur la zone 1 et 1 seconde sur la zone 2).
- **Comparaison entre 1 PID + 1 ANFIS et 2 ANFIS :** selon la **Figure 3.34**, pour le cas des 2 ANFIS, la chute de fréquence est également réduite de 0.02 à 0.03 Hz sur les 2 zones et la convergence est beaucoup plus rapide (environ 4 secondes sur les 2 zones) grâce à l'atténuation des oscillations.
- **Comparaison entre 2 régulateurs flous et 2 ANFIS :** selon la **Figure 3.35**, la chute de fréquence des 2 types de régulateurs est similaire si ce n'est pour une chute légèrement plus faible pour l'ANFIS, ainsi qu'une annulation de l'erreur de fréquence pour l'ANFIS contrairement aux régulateurs flous.

L'utilisation des deux régulateurs ANFIS différents que nous avons conçus sont donc supérieurs à l'utilisation du PID classique et des régulateurs flous en termes de chute de fréquence et en termes de vitesse de convergence.

Il y a cependant une meilleure approche qui consiste à prendre un régulateur ANFIS avec un régulateur PID optimisé (grâce à des méthodes d'IA par exemple) pour profiter de la convergence rapide du ANFIS couplé à sa bonne compatibilité avec le PID.

Le PID implémenté dans la centrale électrique consiste en une action intégrale seulement avec  $K_i = -0.425$ . En introduisant des paramètres proportionnels et dérivatifs, nous pouvons diminuer les oscillations et la chute de fréquence de manière plus efficace.



**Figure 3.36** : Comparaison entre 1 PID optimisé + 1 ANFIS et 2 ANFIS

Il est difficile d'obtenir des temps de convergence plus bas en utilisant les régulateurs PID mais nous pouvons diminuer la chute de fréquence en utilisant des PID optimisés par tâtonnement ou avec des méthodes plus développées comme l'intelligence artificielle.

#### **5. 4. Entraînement d'un modèle ANFIS à partir des données d'un régulateur PID :**

Jusqu'à maintenant, nous n'avons entraîné les modèles ANFIS qu'à partir des données fournies par différents régulateurs flous. L'ANFIS étant lui-même un régulateur flou dans sa conception primaire, il lui est facile d'analyser la logique derrière le mode de fonctionnement d'un régulateur flou et de le reproduire de manière optimisée.

Etant donné les résultats satisfaisants de la combinaison d'un régulateur ANFIS et d'un régulateur PID dont on a varié les paramètres montrés dans la **Figure 3.36**, nous nous demandons s'il est possible d'entraîner un ANFIS à partir de ce PID et d'obtenir des résultats satisfaisants.

La réponse fournie par le PID est beaucoup plus lisse que celle des régulateurs. Ceci est dû au fait que le modèle du régulateur flou utilisé dans cette étude contienne des blocs discrets (delay) qui aboutissent à des réponses moins régulières et donc moins lisses.

Il faut donc utiliser des blocs fournissant des réponses continuent tel la fonction dérivative  $du/dt$  de Simulink.

Il faut également augmenter le nombre des variables d'entrées du régulateur ANFIS pour augmenter le nombre de données nécessaires à son entraînement et ainsi lui donner une meilleure image du modèle PID à copier [12]. Dans cet exemple, les variables d'entrée sont : l'erreur ACE, la dérivée de l'ACE et la puissance de transit  $\Delta P_{12}$ .

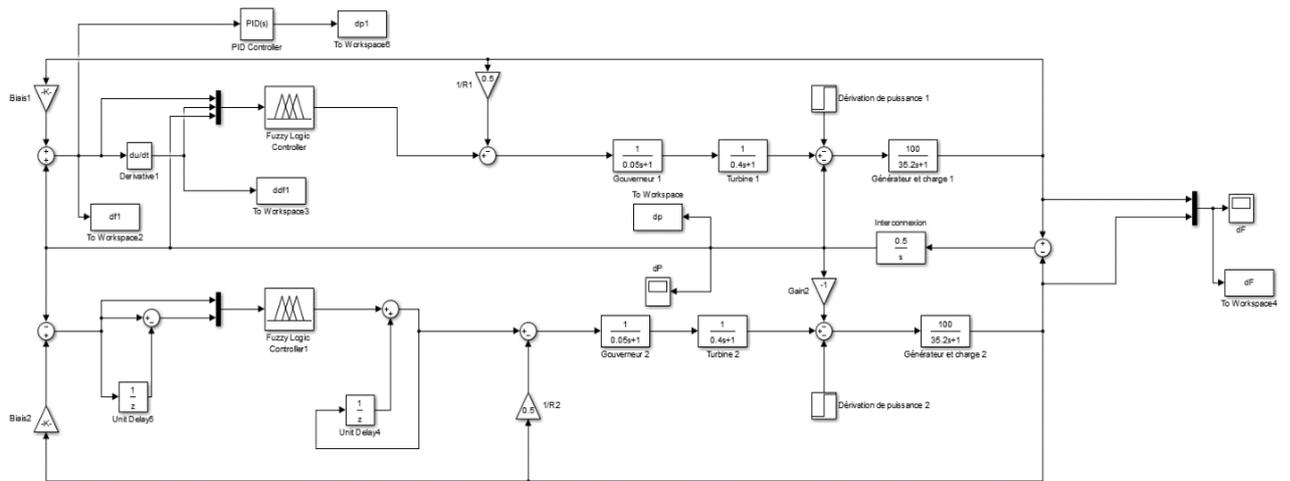


Figure 3.37 : Schéma d'apprentissage et de simulation d'un ANFIS entraîné à partir d'un PID

Grâce au modèle Simulink de la Figure 3.37 nous obtenons les résultats illustrés ci-dessous pour des déviations de puissance de :  $dp_1 = 0\%$  et  $dp_2 = 4\%$  :

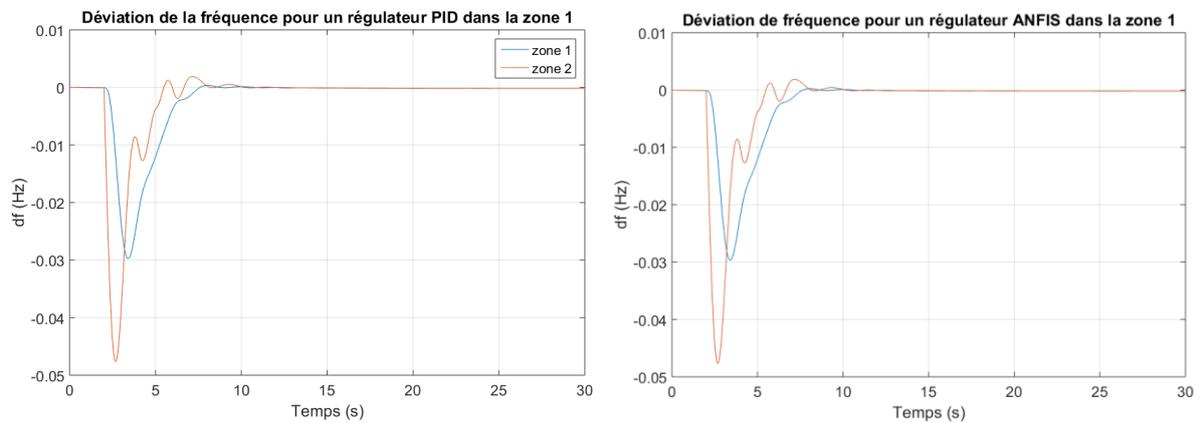
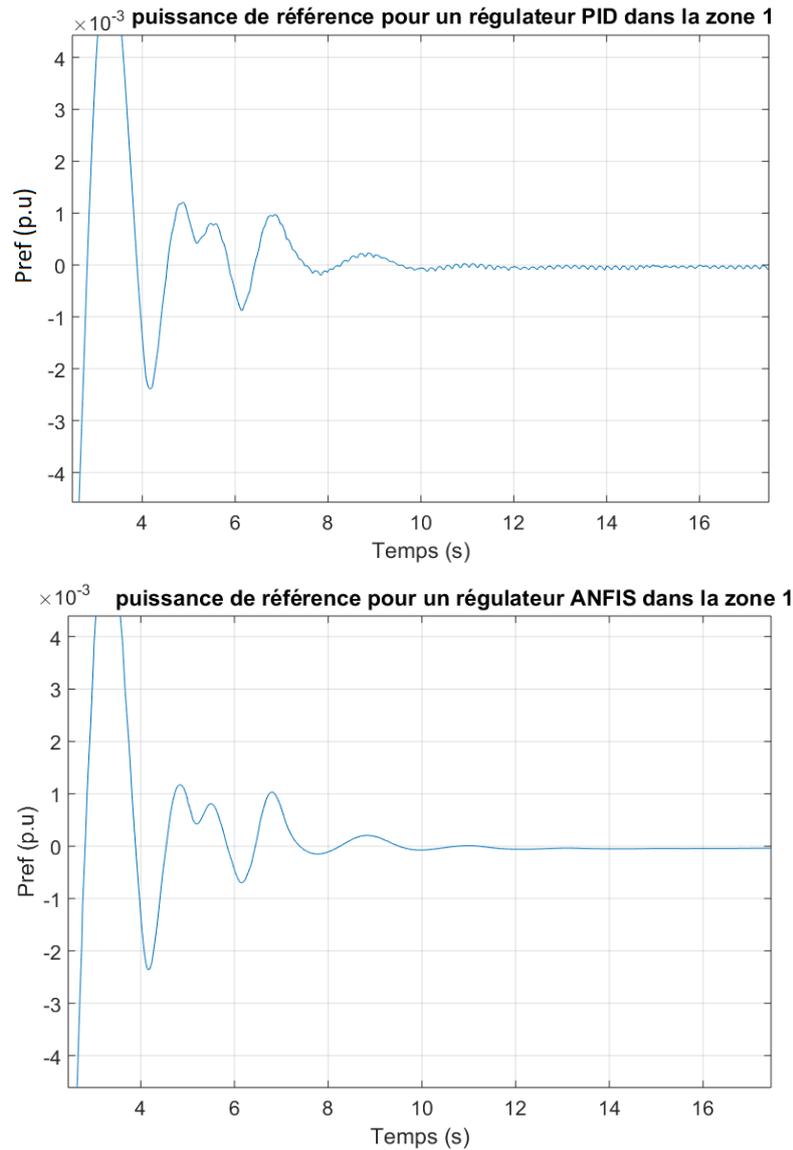


Figure 3.38 : Comparaison entre le PID dans la zone 1 et l'ANFIS qu'il a entraîné

La déviation est exactement la même dans le cas du PID et de l'ANFIS, prouvant ainsi qu'il est possible d'entraîner un modèle ANFIS à partir des données obtenues d'un PID.



**Figure 3.39 :** Puissance de référence dans le cas du PID et de l'ANFIS

L'avantage de cette méthode est l'élimination des perturbations au niveau de la puissance de référence  $P_{ref}$  lors de l'utilisation du ANFIS au lieu du PID.

Plus les paramètres d'un régulateur PID sont augmentés en valeur absolue, plus le correcteur sera performant au niveau de la régulation de la variable désirée et plus la sortie directe du correcteur sera perturbée en état de stabilité.

Les perturbations au niveau de la puissance de référence ne sont pas très importantes. Elles peuvent néanmoins causer de sérieux problèmes de stabilité du système s'il apparaît une nouvelle perturbation de puissance de charge pendant l'instabilité de la puissance de référence.

Dans ce travail, nous n'avons pu entraîner l'ANFIS que dans le cas où le PID n'est soumis à aucune déviation de charge dans sa zone ( $dP_1 = 0\%$  et  $dP_2 = [0 ; 10] \%$ ).

Lorsqu'il y a une déviation de charge au niveau du correcteur PID, sa réponse est très dynamique et abrupte à l'instant de la perturbation. Ce niveau de variation pose problème au

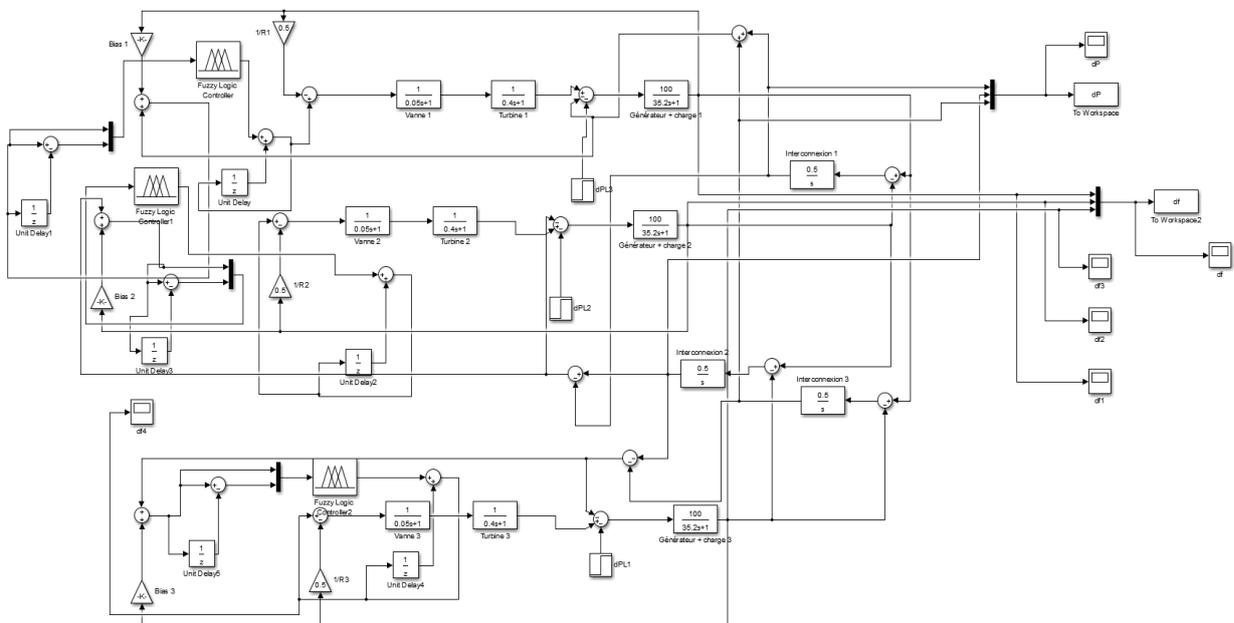
niveau de l'entraînement du ANFIS qui ne peut pas créer de modèle satisfaisant avec la méthode utilisée dans ce travail.

Trouver une méthode pour entrainer un modèle ANFIS à partir d'un régulateur PID bien optimisé, pour toutes les possibilités de déviation de puissance pourrait s'avérer intéressant.

## 6. Application du ANFIS sur 3 zones :

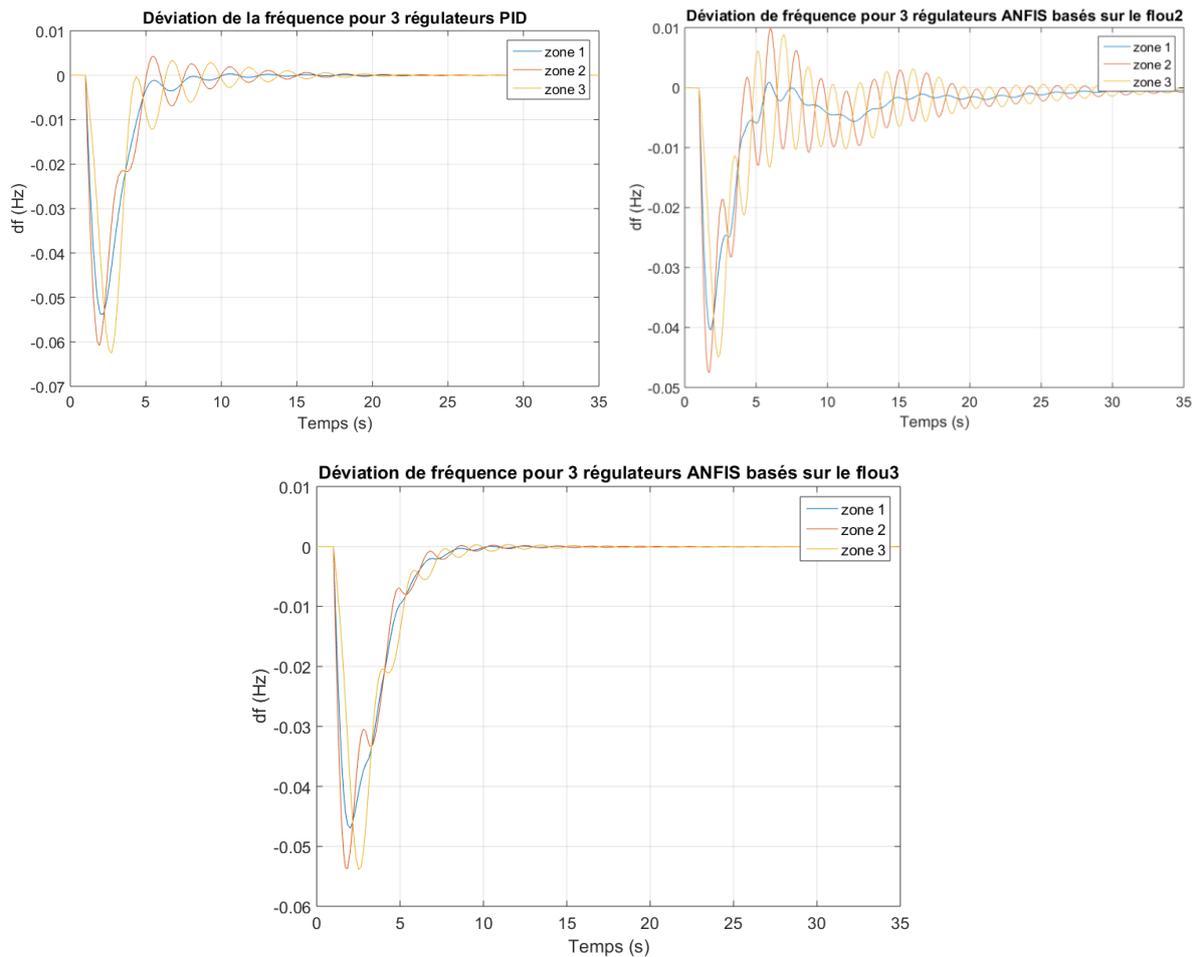
Tout comme le cas à deux zones, les mêmes raisonnements s'appliquent lorsqu'on étudie un cas à trois zones.

En utilisant le schéma Simulink de la **Figure 3.40** nous pouvons voir la différence entre une régulation avec trois régulateurs PID et une régulation avec trois régulateurs ANFIS (les régulateurs ANFIS utilisés étant ceux obtenus à la fin de l'étude du cas deux zones).



**Figure 3.40** : Schéma à 3 zones avec régulation ANFIS

En prenant  $dP_1=3\%$ ,  $dP_2=4\%$  et  $dP_3=1\%$ , nous obtenons les résultats suivants :



**Figure 3.41 :** Réponse du système à 3 zones pour différents régulateurs

Les régulateurs ANFIS offrent des chutes de fréquence plus faibles par rapport au PID avec une diminution d'environ 0.015 Hz pour les ANFIS basés sur le flou2 et de 0.01 Hz pour les ANFIS basés sur le flou3.

Les régulateurs ANFIS basés sur le flou2 provoquent néanmoins énormément d'oscillations de fortes amplitudes qui durent longtemps. Comme vu dans le cas 2 zones, cela est dû aux petites oscillations créées par chaque ANFIS individuellement qui s'amplifient exponentiellement pour 3 zones.

Les régulateurs ANFIS basés sur le flou3 ont quant à eux un très bon temps de convergence avec de faibles oscillations amorties par rapport aux deux autres méthodes. Ces régulateurs représentent donc la meilleure alternative parmi les trois proposées.

## **7. Conclusion :**

Dans ce chapitre, nous avons vu l'application des régulateurs neuro-flous « ANFIS » dans plusieurs cas différents, allant du cas mono-zone au cas de trois zones.

Les résultats obtenus sont beaucoup plus intéressants que ceux obtenus à partir des régulateurs flous, les régulateurs ANFIS représentent donc bien une alternative plus performante en termes de régulateurs intelligents.

La construction d'un modèle ANFIS, c'est-à-dire le choix des intervalles des fonctions d'appartenance, les règles flous, les paramètres des fonctions conséquentes..., est optimisé par un réseau de neurone de cinq couches, calculant ainsi les paramètres optimaux du régulateur. Néanmoins, l'impact de l'homme se fait toujours ressentir à travers le choix des données d'entraînement. Le concept même du ANFIS est de créer un modèle dont les résultats tendent vers ceux fournies dans les données d'entraînement. De mauvaises données d'entraînement entraîneront donc un modèle ANFIS avec de mauvaises performances.

Dans ce travail, nous avons identifier deux types de mauvaises données d'entraînement :

- Données qui font diverger l'entraînement du ANFIS : Toutes les données ne sont pas facilement reproductibles par la logique floue. Des données non quantifiables en domaines et/ou règles flous ne pourront pas créer un modèle ANFIS fonctionnel, entraînant donc la divergence de l'entraînement.
- Données tirées à partir de modèles peu performants : Des données tirés de régulateurs flous non performants pourront aisément entraîner un modèle ANFIS, mais ce dernier sera fortement limité par leurs performances.

Il faut avant tout, obtenir des données performantes, et cela ne serait-ce que pour un seul point de fonctionnement du système, afin de créer un modèle ANFIS qui pourra généraliser ces résultats pour le restant des points.

Dans ce chapitre, nous avons identifié deux façons dont l'ANFIS peut améliorer les données tirées d'un régulateur flou :

- Généraliser les données d'un point de fonctionnement avec de bons résultats sur tous les autres points de fonctionnement à travers la règle de trois, leur donnant ainsi la même forme de réponse.
- Forcer un signal à tendre vers zéro en remplaçant ses valeurs de convergence non nulles par des valeurs nulles lors de l'entraînement de l'ANFIS.

L'ANFIS bénéficiera ainsi de la souplesse, de l'adaptabilité et de la performance fournie par la logique floue avec l'optimisation et la correction apportée par les réseaux de neurones.

Nous avons aussi remarqué des perturbations sous la forme de petites oscillations dans les cas multizone lors de l'utilisation de plusieurs régulateurs ANFIS, retardant ainsi le temps de convergence de la fréquence. Cette perturbation est le fruit du manque de coordination entre les différents régulateurs qui agissent indépendamment. C'est aussi dû à la présence d'oscillations générées par chaque régulateur isolé. Ces oscillations s'amplifient en se rencontrant dans le cas où il y a plusieurs régulateurs ANFIS.

# **Chapitre 4 :**

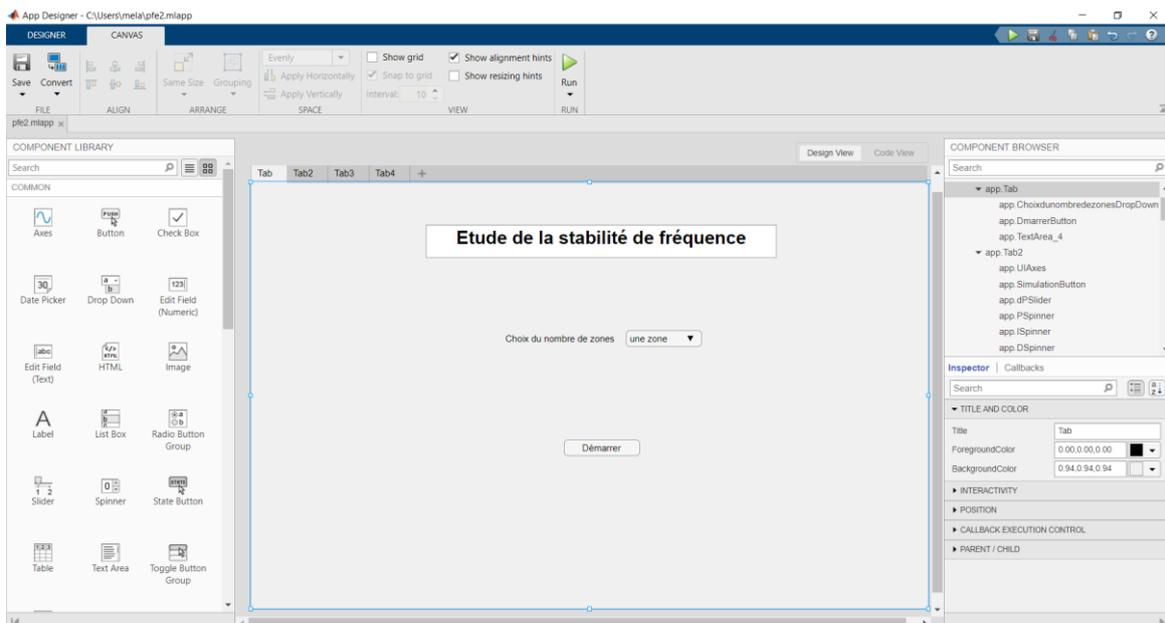
Création d'une interface graphique

## 1. Appdesigner :

Pour regrouper les résultats obtenus au cours de ce travail en une seule application, nous avons utilisé l'outil « appdesigner » de MATLAB.

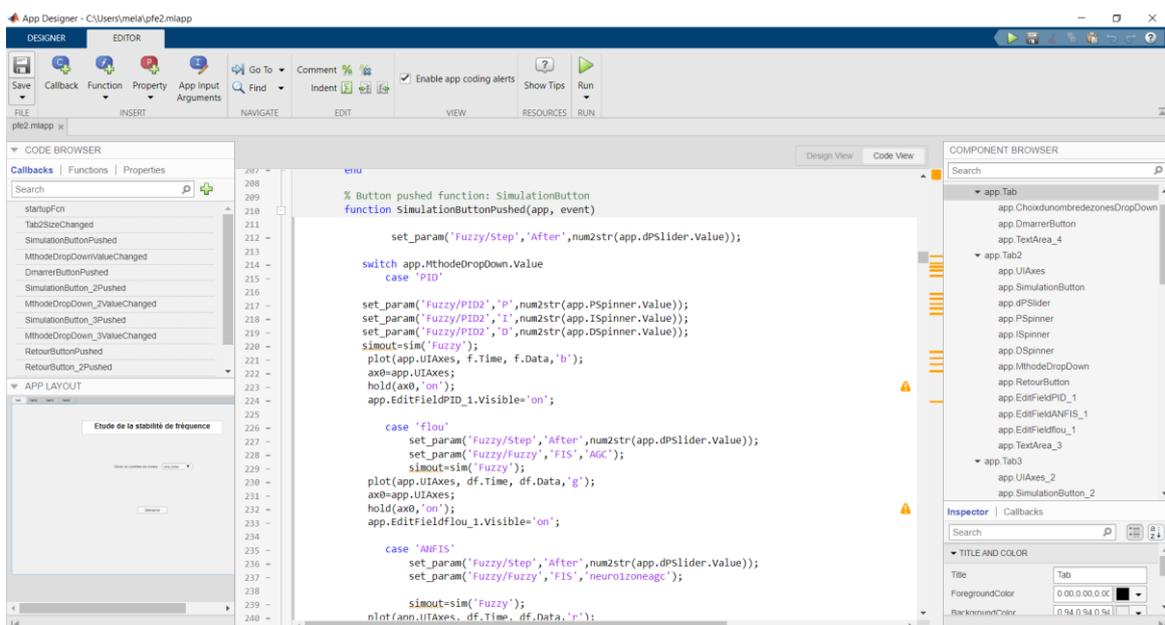
L'appdesigner permet de créer des applications en suivant deux tâches principales :

- Le design de l'interface graphique utilisateur (GUI) en utilisant la bibliothèque des composantes visuelles fournies (**Figure 4.1**) :



**Figure 4.1** : Fenêtre de MATLAB GUI

- La programmation du comportement de l'application (**Figure 4.2**) :



**Figure 4.2** : Fenêtre du codage du MATLAB GUI

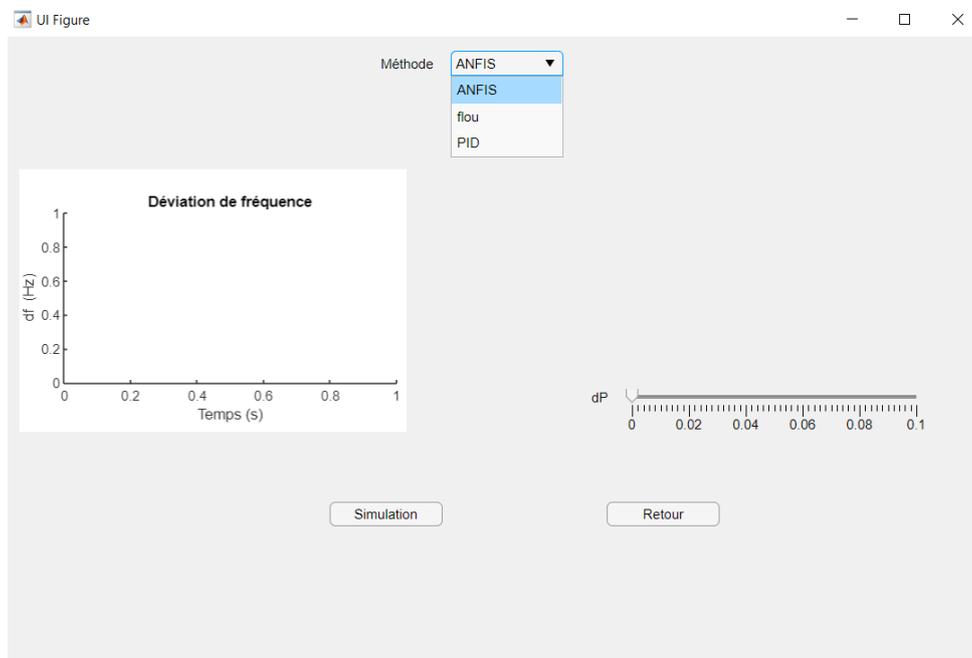
## 2. Présentation de l'application :

En synchronisant l'application créée avec les schémas Simulink utilisés précédemment dans ce travail, nous obtenons l'interface suivante :



**Figure 4.3 :** Fenêtre d'accueil de l'application

A partir de la première fenêtre, nous pouvons choisir le nombre de zones que l'on veut étudier.



**Figure 4.4 :** Fenêtre du cas mono-zone

La fenêtre du cas mono-zone offre le choix du correcteur et de la puissance de perturbation à appliquer.

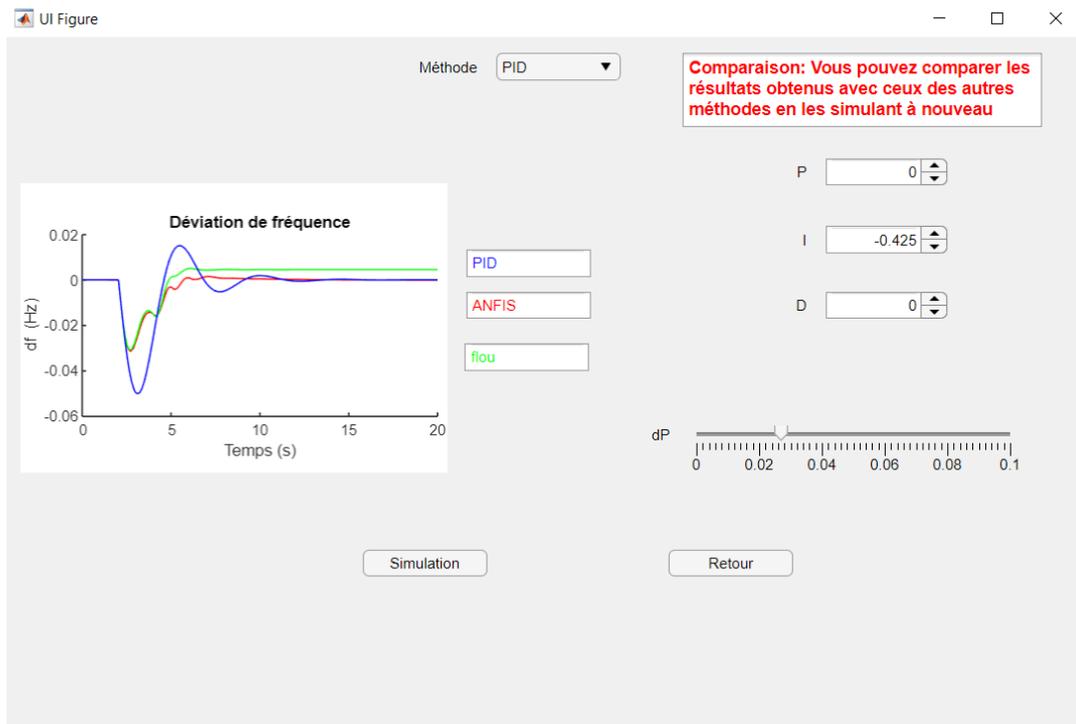
Le bouton « Simulation » lance la simulation du système tandis que le bouton « retour » renvoie l'utilisateur vers la première fenêtre de l'application.



Figure 4.5 : Réponse du système mono-zone au régulateur ANFIS

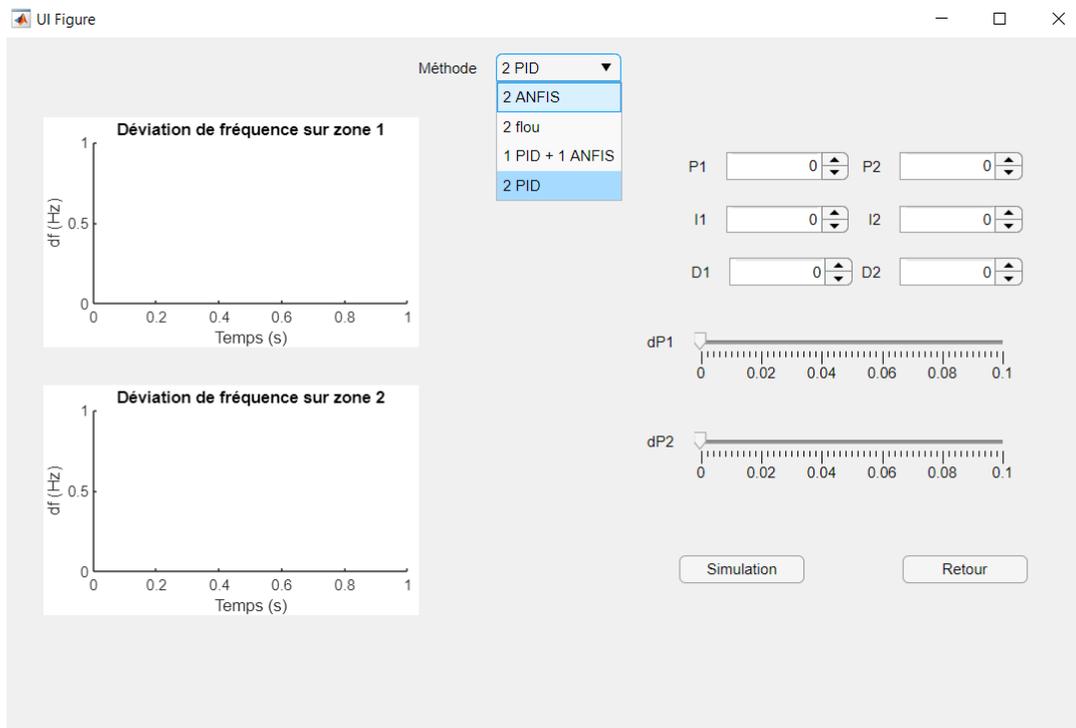


Figure 4.6 : Comparaison avec le régulateur flou pour le cas mono-zone



**Figure 4.7 :** Comparaison entre les différents régulateurs pour le cas mono-zone

- Une légende s'affiche en même temps que chaque réponse.
- L'utilisateur peut choisir les paramètres du PID.
- Il est possible d'afficher les réponses de plusieurs correcteurs en même temps afin de les comparer.



**Figure 4.8 :** Fenêtre du cas 2 zones

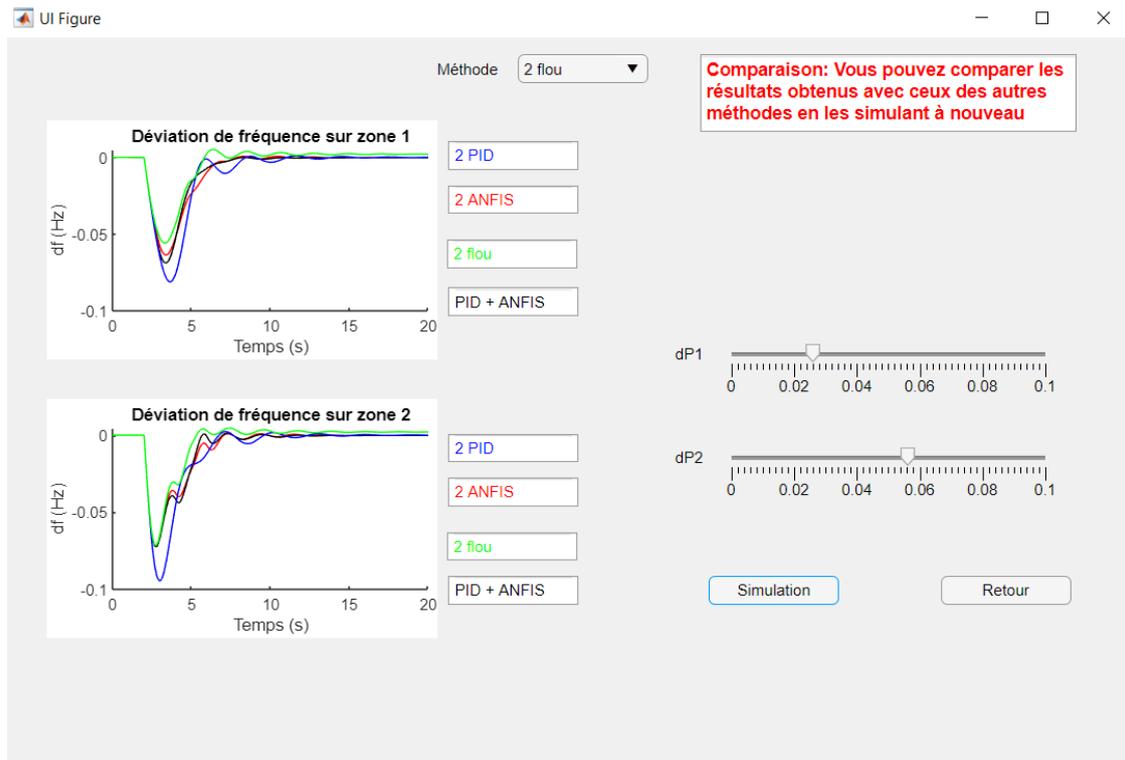


Figure 4.9 : Comparaison entre les différents régulateurs pour le cas 2 zones

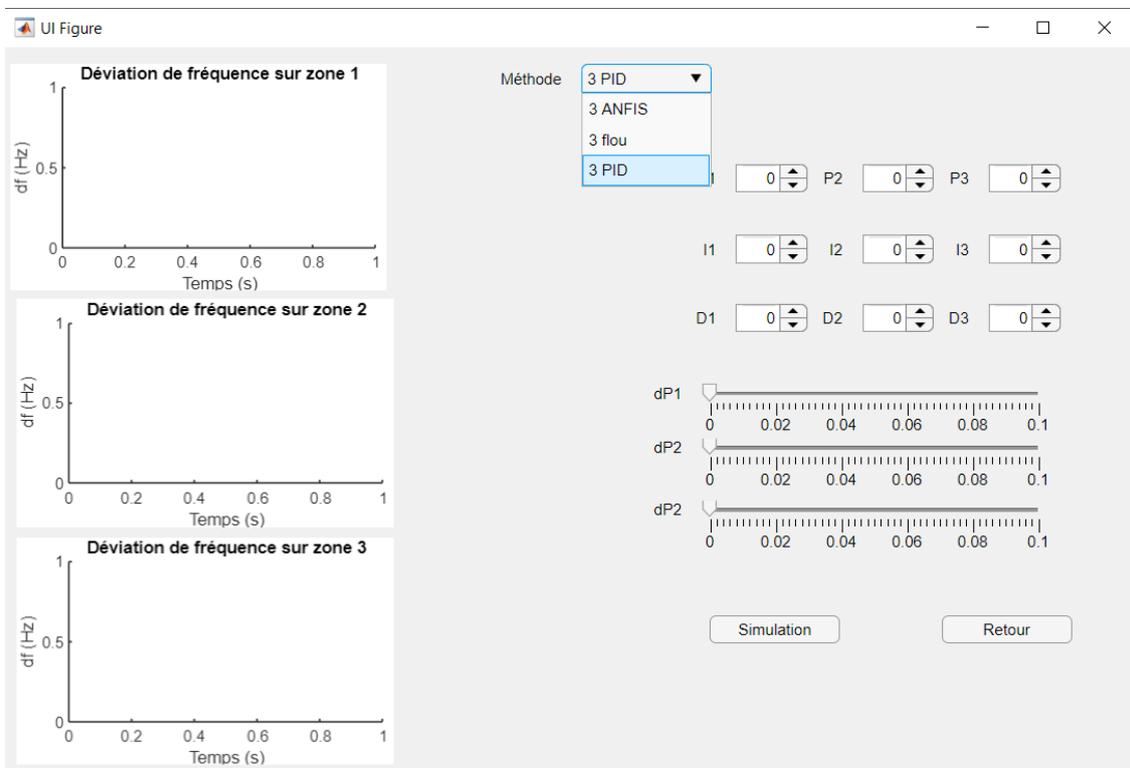
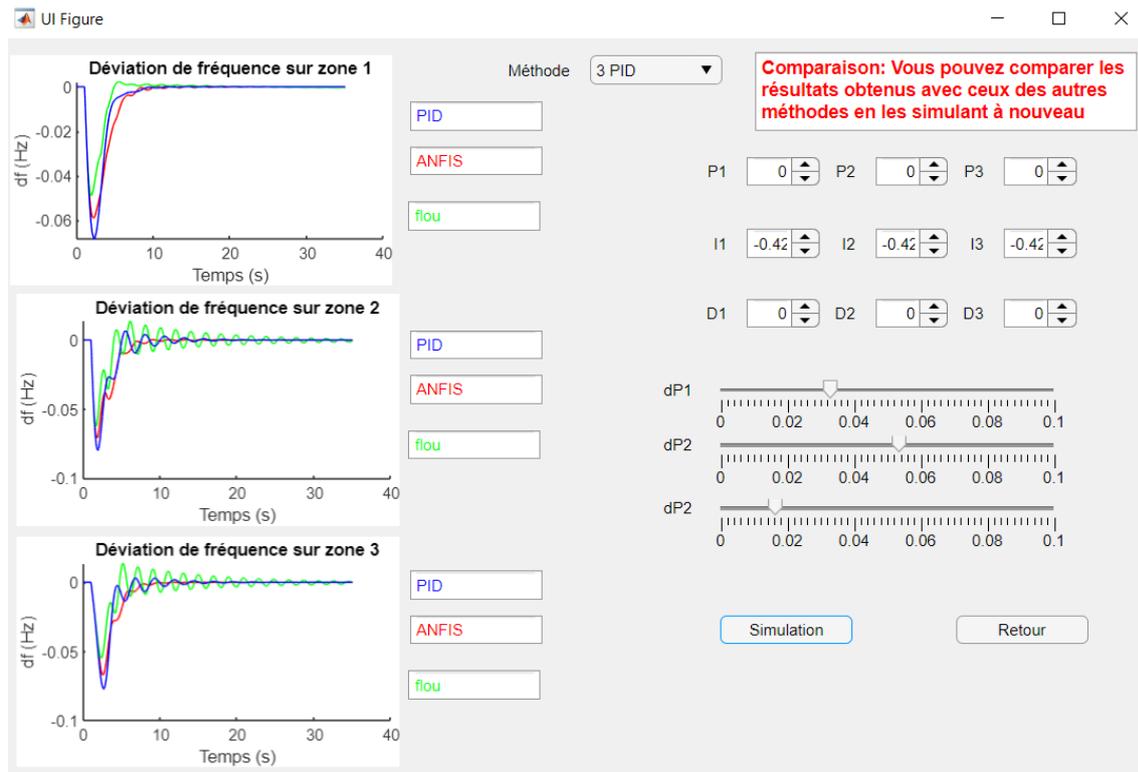


Figure 4.10 : Fenêtre du cas 3 zones



**Figure 4.11 :** Comparaison entre les différents régulateurs pour le cas 3 zones

Les fenêtres des cas 2 zones et 3 zones sont similaires à la fenêtre du cas mono-zone. Ces fenêtres proposent en plus leurs propres régulateurs, la possibilité de choisir la puissance de perturbation de chaque zone et la possibilité de choisir les paramètres de chaque PID.

### **3. Conclusion :**

L'application permet de visualiser et de comparer les réponses des erreurs de fréquence pour différentes zones, pour différentes valeurs de perturbation et pour différents correcteurs tout en offrant une interface interactive pour l'utilisateur.

# **Conclusion générale**

Dans ce mémoire, nous avons étudié l'utilisation des régulateurs hybrides neuro-flous dans l'AGC des réseaux électriques dans plusieurs cas différents. Et nous avons comparé les résultats obtenus avec ceux des régulateurs PID classiques et des régulateurs flous.

Nous avons d'abord modélisé le réseau électrique sur Simulink en transformant chaque composante principale du réseau en modèle linéaire.

Suite à cela, des régulateurs PID classiques ont été testés sur des cas mono-zone ou multizones où nous avons obtenu des chutes de fréquence et des temps de convergence assez importants.

Nous avons introduit le régulateur flou comme une alternative intelligente au PID. Les résultats obtenus par les régulateurs flous variaient énormément avec la variation des paramètres de base, nous donnant pour certains modèles des réponses lentes mais lisses et pour d'autres des réponses plus rapides avec des convergences non nulles et oscillatoires.

En observant l'importance du choix des paramètres du régulateur flou, nous avons introduit le concept des correcteurs neuro-flous dans l'AGC. Grâce aux capacités d'apprentissage, d'entraînement et de correction des réseaux de neurones, nous avons pu concevoir des correcteurs flous, plus connus sous le nom d'ANFIS, qui corrigent les défauts des correcteurs flous précédents, et ce en corrigeant les données obtenues à partir des régulateurs flous et en les utilisant pour entraîner l'ANFIS.

Les régulateurs ANFIS ont montré des performances supérieures aux deux autres méthodes avec des chutes de fréquence et des temps de convergence plus faibles.

Nous avons néanmoins remarqué que l'utilisation de plusieurs régulateurs ANFIS dans un réseau multizone peut créer des petites perturbations qui retardent le temps de convergence de la fréquence. Ceci est dû au manque de coordination entre les régulateurs qui agissent indépendamment de leur contrepartie. C'est également dû à la présence d'oscillations générées par ces régulateurs qui s'amplifient dans le cas où il y a plusieurs régulateurs ANFIS.

Une autre limitation du ANFIS est le choix des données d'entraînement qui limitent les performances du régulateur si elles ne sont pas choisies de manière optimale, choix qui dans ce travail dépend de l'utilisateur.

Les régulateurs ANFIS sont donc de très bons substituts aux régulateurs PID et flous classiques. On peut néanmoins apporter quelques améliorations à cette méthode.

Pour terminer, nous avons créé une application sur l'Appdesigner de MATLAB afin d'offrir un moyen de visualisation et de comparaison des données interactif aux utilisateurs.

Voici quelques perspectives pour essayer d'optimiser d'avantage la régulation de fréquence :

- Obtenir des données d'entraînement à partir de régulateurs flous plus performants.
- Identifier d'autres façons d'amélioration des données tout en fournissant des données utilisables par l'ANFIS.
- Essayer d'entraîner l'ANFIS à partir des régulateurs PID optimisés par méta-heuristiques pour supprimer complètement le besoin des régulateurs PID et alléger les calculs.
- Créer des ANFIS qui peuvent coopérer de manière optimale dans les cas multizones.

## Références bibliographiques :

- [1] Abdelhafid Hellal, "Commande automatique de production (AGC)", Cours à l'Ecole Nationale Polytechnique d'Alger, 2020-2021.
- [2] Abdelhadi Mohamed Nial et Zanoun Meriem, "Etude des boucles de LFC intelligentes dans l'AGC des réseaux électriques", Mémoire de Projet de Fin d'Etude, Ecole Nationale Polytechnique d'Alger, Juillet 2021.
- [3] Djaber Benghouini, "Optimisation des paramètres du PID dans l'AGC avec les méta-heuristiques" Mémoire de Magister, Université Amar Thelidji, Laghouat, 2011.
- [4] Aspexit, "La logique floue ou l'extension de la logique classique ", Publié sur site, Avril 2019.
- [5] Abdelhafid Hellal, "Polycopié du cours d'intelligence artificielle", Cours à l'Ecole Nationale Polytechnique d'Alger, 2020-2021.
- [6] Surya Prakash et S. K. Sinha, "Application of artificial intelligence in load frequency control of interconnected power system", International Journal of Engineering, Science and Technology, Vol. 3, No. 4, pp. 264-275, 2011.
- [7] H. Buhler, "Réglage par logique floue", Edition Presses polytechniques et universitaires romandes, Lausanne, 1994.
- [8] J.R. Jang, "ANFIS: Adaptive-network-Based Fuzzy Inference System", IEEE Trans. On Systems, Man and Cybernetics, Vol. 23, No.3, May. 1993.
- [9] R. J. Jyh-Shing, Chuen-Tsai Sun et Eiji Mizutani, "ANFIS: Adaptive Neuro-Fuzzy Inference Systems", Neuro-fuzzy and soft computing: a computational approach to learning and machine learning, 1997.
- [10] "Anfis", Mathworks.com.
- [11] Swasti R. Khuntia et Sidhartha Panda, "A Novel Approach for Automatic Generation Control of a Multi-Area Power System", 24<sup>th</sup> Canadian conference on electrical and computer engineering, 2011.
- [12] Meisam Babanezhad, Armin Masoumian, Ali Taghvaie Nakhjiri, Azam Marjani and Saeed Shirazian , "Influence of number of membership functions on prediction of membrane systems using adaptive network based fuzzy inference system (ANFIS)", Nature.com/open acces, September 2020.