

Université Mohamed Khider - Biskra
Faculté des Sciences et de la technologie
Département: Génie Electrique
Réf :



جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم : الهندسة الكهربائية
المراجع :

Mémoire présenté en vue de l'obtention
Du Diplôme de

Magister en Automatique

Option : Commande et Identification des Systèmes Dynamiques

Commande tolérante aux défauts en utilisant les Réseaux de Neurones Artificiels et les Systèmes d'Inférence Floue

Présentée par :

ACHBI Mohammed Said

Soutenue publiquement le :26/01/2012

Devant le jury composé

ZOUZOU Salah Eddine	Professeur	Université de Biskra	Président
BOUMEHRAZ Mohamed	Maître de conférences A	Université de Biskra	Rapporteur
DRID Said	Maître de conférences A	Université de Batna	Examineur
MENACER Arezki	Maître de conférences A	Université de Biskra	Examineur

Remerciements

En premier lieu, je remercie Dieu, le tout puissant, pour m'avoir donné, le courage, la patience, la volonté et la force nécessaires, pour affronter toutes les difficultés et les obstacles, qui se sont hissés au travers de mon chemin, durant toutes mes années d'études.

Je tiens à remercier infiniment mon professeur **Dr. BOUMEHRAZ Mohamed**, Maître de conférences à l'université de Biskra, d'avoir voulu diriger ce travail ainsi que pour leur aide, leur disponibilité et leur encouragement qu'il n'a cessé de me prodiguer, tout au long de la réalisation de ce mémoire.

Je remercie Monsieur **ZOUZOU Salah Eddine**, Maître de conférences à l'université de Biskra, pour avoir accepté de me faire l'honneur de présider le jury de ma soutenance, ainsi que : Monsieur **DRID Said**, Maître de conférences à l'Université de Batna, Monsieur **MENACER Arezki**, Maître de conférences à l'Université de Biskra, d'avoir accepté d'être membres du jury.

Finalement, je remercie toutes personnes de près ou de loin qui m'ont aidé à finir ce travail et je voudrais dire qu'avec la finalisation de ma mémoire commence une nouvelle étape de ma vie pour laquelle chacune des personnes qui ont été proches de moi m'ont apporté, merci encore à tous.

Introduction générale

Introduction générale

Suite au développement important et rapide qu'a connu le monde industriel, les procédés de fabrication sont de plus en plus complexes et sophistiqués et la tâche de diagnostic est devenue indispensable pour assurer la sûreté de fonctionnement et la disponibilité de ces systèmes. Par conséquent, l'augmentation de la fiabilité, la disponibilité et la sûreté de fonctionnement, constitue à l'heure actuelle, l'une des préoccupations majeures des industriels. La recherche dans le domaine de supervision et l'utilisation des outils déjà existants en automatique, ont permis de développer des méthodes de sûreté de fonctionnement telles que le diagnostic des défauts affectant les systèmes.

Dans certains systèmes complexes, (aéronautique, centrales nucléaires, systèmes économiques...), la phase de détection et de diagnostic des défauts est nécessaire mais n'est pas suffisante pour garantir la sûreté de fonctionnement car il est indispensable de modifier la loi de commande en temps réel afin de maintenir la stabilité du système et de garantir ainsi un fonctionnement acceptable en mode dégradé. Ainsi, il est nécessaire d'associer au diagnostic une loi de commande tolérante aux défauts. Donc, cette étude est très intéressante de passer par une phase de modélisation visant à obtenir une représentation permettant de décrire son fonctionnement. En effet, l'hypothèse de linéarité des relations entrées-sorties d'un système permet d'élaborer simplement un modèle approximant son comportement. Cependant, de tels modèles ne permettent la représentation du comportement d'un système qu'autour d'un point de fonctionnement donné, l'hypothèse de linéarité n'étant vérifiée que dans une zone restreinte de l'espace de fonctionnement. Sachant que les systèmes réels sont de nature non linéaire, les systèmes de commande et de diagnostic développés sur la base de modèles linéaires fournissent des performances dégradées dès qu'on s'éloigne du point de fonctionnement. Donc, il est impératif de prendre en considération les non-linéarités dans la phase de modélisation.

Plusieurs techniques pour la surveillance et le diagnostic des défauts dans les systèmes dynamiques ont été développées ces dernières années. Cependant la plupart de ces techniques sont basées sur la surveillance du procédé seul sans la prise en compte de son interaction avec le système de commande. L'utilisation des techniques d'intelligence permet d'avoir une représentation exacte d'un modèle non linéaire exprimé sous une forme générale et offre aussi la possibilité de représenter une classe plus large de systèmes non linéaires.

L'objectif de ce mémoire est de proposer une stratégie de commande tolérante aux défauts basée sur l'utilisation des techniques d'intelligence artificielle (réseaux de neurones, les systèmes d'inférence floue) permettant de s'accommoder automatiquement l'effet des défauts qui peuvent apparaître sur un système tout en étant capable de maintenir le fonctionnement nominal.

Ce mémoire est organisé en cinq chapitres qui peuvent être résumés comme suit :

Dans le premier chapitre, nous présenterons la terminologie du domaine et l'analyse du terme diagnostic et nous donnons aussi les différents types de défauts qui peuvent se produire dans un procédé. Après, nous présenterons quelques généralités sur le diagnostic des systèmes dynamiques et décrivons les différentes voies suivies dans le domaine du diagnostic des défauts.

Le deuxième chapitre présentera les principales commandes tolérantes aux défauts pour amener et justifier les choix et les orientations de ce mémoire. Nous avons donné une idée détaillée sur les différents axes de recherche ainsi que les travaux effectués dans ce domaine. Il permet notamment de classer diverses méthodes en deux grandes catégories : les lois de commandes tolérantes aux défauts passives d'une part, et actives d'autre part.

Nous proposons dans le troisième chapitre une étude bibliographique sur la modélisation des systèmes non linéaires. Nous représenterons les méthodes de modélisation par les réseaux de neurones, les systèmes d'inférence floue et les systèmes neuro-flous. Nous citons aussi les méthodes de construction de ces modèles.

Le quatrième chapitre a pour objectif de représenter avec détails les différentes étapes de diagnostic et de commande tolérante aux défauts où ces étapes sont réalisées avec les techniques de réseaux de neurones, les systèmes d'inférence floue et les systèmes neuro-flous.

Le cinquième et dernier chapitre, présentera une étude détaillée de commande tolérante aux défauts présentées au quatrième chapitre sur un système hydraulique/thermique de trois cuves. La modélisation de ce système est présentée, et les résultats de l'application des méthodes de diagnostic et de commande tolérante aux défauts sont exposés et analysés.

Chapitre I

Introduction au diagnostic des défauts

- 1.1. Introduction*
- 1.2. Terminologie.*
- 1.3. Classification des défauts*
- 1.4. Modélisation des défauts*
- 1.5. Le diagnostic des défauts*
- 1.6. Différentes étapes de diagnostic*
- 1.7. Classification des méthodes de diagnostic*
- 1.8. Procédure de diagnostic a base de modèle*
- 1.9. Critères de performance d'un système de diagnostic*
- 1.10. Conclusion*

I.1. INTRODUCTION

Un système tolérant aux défauts possède la capacité de maintenir les objectifs nominaux en dépit de l'occurrence d'un défaut et à s'en accommoder de manière automatique. Il permet notamment de garantir la stabilité du système et/ou des performances dégradées acceptables en présence de défauts.

Il convient de noter que la réalisation d'une commande tolérante aux défauts n'est pas un problème classique et que les conditions induites par la présence de défauts obligent à développer des stratégies particulières. La synthèse d'une loi de commande tolérante aux défauts impose de mettre en œuvre une démarche comprenant en général deux étapes :

- étape de diagnostic et évaluation des performances ;
- étape de synthèse d'un mécanisme d'accommodation, ou de reconfiguration du système.

Le diagnostic des défauts signifie la détermination des composants défailants au sein d'un système physique. Le diagnostic des défauts dans les systèmes physiques, a été un sujet important de recherche dès les débuts de l'automatique moderne. On distingue généralement, dans la procédure de diagnostic, deux étapes principales : la détection et la localisation, plus l'identification pour certaines méthodes. Les définitions précises de ces notions sont rappelées dans la suite de cette section.

L'objectif de ce chapitre introductif est de rappeler les techniques les plus courantes dans le domaine de diagnostic des défauts et les différents types de défauts, ainsi que les définitions des termes souvent utilisés en diagnostic. Dans la littérature associée à ce domaine, on peut trouver plusieurs définitions quelquefois divergentes, c'est pourquoi nous nous mettons dans un premier temps les définitions des mots clés qui sont utiles pour la compréhension de ce travail.

I.2. TERMINOLOGIE

La diversité des définitions trouvées dans différents travaux fait que nous avons jugé important d'établir un lexique des termes qui seront utilisés dans ce travail. Nous commençons, tout d'abord, par préciser les définitions de quelques notions propres au diagnostic. On peut trouver en littérature des définitions qui sont complètement différentes de celles données dans ce mémoire, mais il est utile de préciser la signification de ces termes du point de vue automatique. Les définitions ont été extraites à partir des références suivantes [4 ,22 ,56 , 77,81].

- **Défaut :**

Le défaut possède plusieurs significations dans la littérature. Un défaut se définit comme une anomalie comportementale au sein d'un système physique localisée au niveau d'un composant. Aussi, on considère comme un défaut tout écart entre la caractéristique observée sur le dispositif et la caractéristique de référence, lorsque celui-ci est en dehors des spécifications.

- **Défaillance :**

Une défaillance se définit comme une anomalie fonctionnelle au sein d'un système physique, c'est-à-dire la cessation de la capacité d'un système physique ou d'un dispositif à accomplir ses fonctions.

- **Panne :**

C'est l'inaptitude d'un dispositif à accomplir une fonction vitale. Dès l'apparition d'une défaillance, caractérisée par la cessation du dispositif à accomplir sa fonction, le dispositif sera déclaré en panne. Par conséquent, une panne résulte toujours d'une défaillance.

- **Dégradation:**

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

- **Symptôme :**

Un symptôme est défini comme un caractère distinctif d'un état fonctionnel anormal.

- **Signe :**

Un signe est défini comme un caractère distinctif d'un état comportemental anormal.

- **Perturbation :**

On peut définir une perturbation comme une entrée inconnue du système et qui n'est pas une commande, c'est une entrée non contrôlée.

- **Résidu :**

Un résidu est souvent appelé indicateur de défaut ou variable d'écart, parce qu'il résulte d'une comparaison entre un comportement réel et un comportement de référence.

- **Détection :**

La détection permet de caractériser le fonctionnement du système de normal ou d'anormal, c'est-à-dire, la détermination de la présence de défauts.

- **Localisation :**

La localisation permet de déterminer le sous-ensemble fonctionnel défaillant.

- **Isolation :**

L'isolation c'est la détermination des caractéristiques des défauts qui se produisent en précisant le type, l'emplacement et l'instant de défaut.

- **Reconfiguration :**

Une fonction consistant à changer la commande envoyée au système ou la disposition matérielle du système pour éviter (ou faire face à) une panne.

I.3. CLASSIFICATION DES DEFAUTS

Lorsque l'on conçoit un système de diagnostic, la première question que l'on se pose, est de savoir ce que l'on veut détecter, c'est-à-dire de définir le type de dysfonctionnement que l'on veut diagnostiquer. Les défauts affectant un système peuvent être de différentes natures et sont généralement classés en défauts d'actionneurs, défauts de capteurs et défauts de système. Ainsi on pourra être amené à détecter des biais, des dérives et/ou des valeurs aberrantes. Selon l'influence sur les systèmes, les défauts peuvent être classés d'additifs et multiplicatifs.

I.3.1. Selon leurs types

Généralement dans la littérature, les défauts sont classés en fonction de leur localisation (emplacement). Les défauts sont des événements qui apparaissent à différents endroits du système. Donc, il est possible de distinguer trois types de défauts, à savoir :

- 1) Les défauts actionneurs :**

Un défaut actionneur est une défaillance représentant une perte totale ou partielle d'un actionneur agissant au niveau de la partie opérative et détériorant le signal d'entrée du système.

Par exemple, un actionneur collé, ne répond plus au signal de commande, représente le cas d'une perte totale de l'actionneur. Par contre, une fuite dans un vérin ou une perte de puissance d'un moteur représente un défaut actionneur partiel qui réagit de manière similaire au régime nominal mais en partie seulement, c'est-à-dire avec une certaine dégradation dans son action sur le système.

2) Les défauts capteurs :

Un défaut capteur peut aussi être partiel ou total. Ce défaut est la cause d'une mauvaise image de l'état physique du système. Un capteur totalement défectueux donne une information différente de la valeur réelle comme par exemple le cas d'une coupure d'un fil électrique. Un défaut capteur partiel apparaît sous forme d'un biais, d'une dérive, d'une baisse d'efficacité, d'un défaut de calibrage...etc. Ce défaut produit un signal avec plus ou moins d'adéquation avec la valeur vraie de la variable à mesurer. Ceci peut se traduire par une réduction de la valeur affichée par rapport à la valeur vraie, ou de la présence d'un biais ou de bruit accru empêchant une bonne lecture. A fin de détecter des défauts capteurs, l'utilisation d'une redondance matérielle est possible pour augmenter la tolérance aux défauts. Cette technique est très fiable, mais ses inconvénients majeurs sont l'encombrement et le coût élevé.

3) Les défauts systèmes :

Les défauts systèmes sont des défauts qui apparaissent dans les composants du système lui-même, c'est-à-dire les défauts qui ne peuvent pas être classifiés ni parmi les défauts capteurs, ni les défauts actionneurs. Ces défauts représentent des changements dans les paramètres du système. A cause de la diversité des situations de défaillances, ce type de défaut est difficile à diagnostiquer.

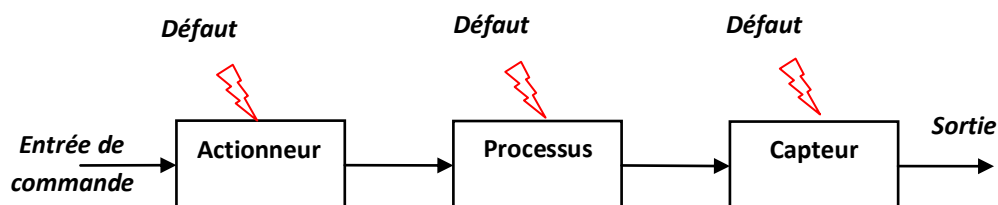


Figure I.1 : Les défauts actionneurs, capteurs et composants.

I.3.2. Selon leurs caractéristiques temporelles

Une classification des défauts à partir de leurs évolutions temporelles peut être divisée en trois catégories distinctes :

1) Les biais (*Abrupts*):

Un biais se définit comme un saut brutal du signal. Habituellement, il est très grave car il affecte les performances et/ou la stabilité du système commandé. La caractéristique principale de ce type de défaut est la discontinuité dans l'évolution temporelle de la variable.

2) Les valeurs aberrantes (*Intermittentes*):

Les valeurs aberrantes sont des défauts dits fugitifs, qui représentent un cas particulier de défauts abrupts avec la propriété d'un signal qui revient de façon aléatoire aux valeurs normales. Ces défauts affectent le système de manière instantanée et se manifestent par un écart important par rapport à la valeur nominale du signal. Leur cause est souvent due à un parasite.

3) Les dérives (*Graduels*):

Une dérive correspond à une croissance lente et continue du signal défaut et donc un éloignement progressif de sa valeur nominale. Ce type de défaut est plus difficile à détecter à son origine du fait de sa faible amplitude et son évolution lente [53].

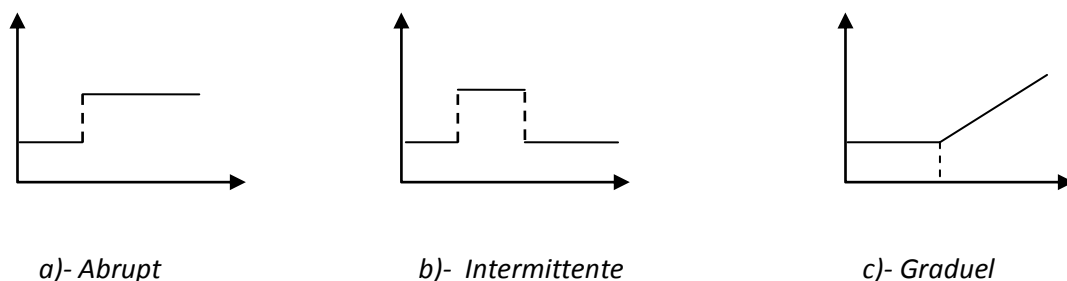


Figure I.2 : Différents types de défauts : défaut abrupt, intermittent, graduel.

I.3.3. Selon leur influence

Les défauts peuvent être différenciés selon leurs influences sur les systèmes. Donc ils peuvent être qualifiés d'additifs et multiplicatifs. Selon la figure (I.3), des défauts sont additifs s'ils affectent le comportement du procédé indépendamment des entrées connues (tels que les défauts actionneurs et défauts capteurs). S'ils affectent le procédé d'une façon dépendante des entrées connues, ils sont considérés comme étant multiplicatifs (défauts système).

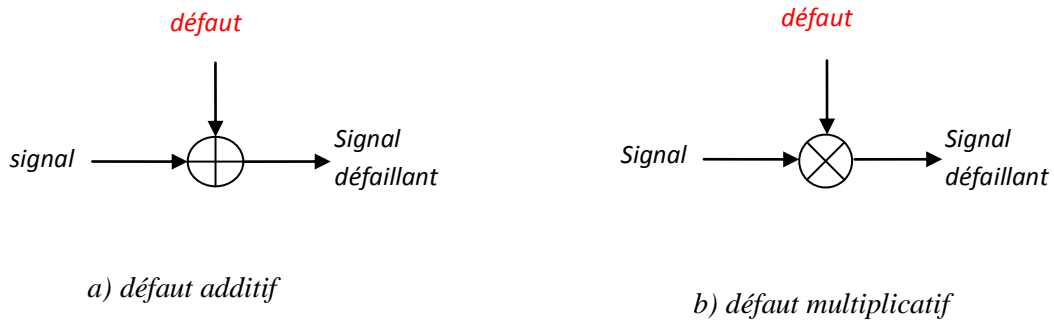


Figure I.3 : Représentation additive et multiplicative des défauts.

I.4. MODELISATION DES DEFAUTS

Comme déjà mentionné dans la section précédente, les défauts peuvent être représentés soit additivement soit multiplicativement. Dans cette section nous intéressons à la représentation mathématique des défauts.

Si l'on dispose d'un modèle linéaire d'un procédé dans des conditions nominales sous la forme d'une représentation d'état, telle que :

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u \\ y = C \cdot x + D \cdot u \end{cases} \quad (I.1)$$

Où x, u et y , représentent les vecteurs d'état, de commande et de sortie du système respectivement, tel que : $x \in \mathbb{R}^n, y \in \mathbb{R}^p$ et $u \in \mathbb{R}^m$ et A, B, C, D sont des matrices de dimensions respectives $n \times n, n \times m, p \times n$ et $p \times m$.

I.4.1. Défauts multiplicatifs

Le modèle multiplicatif est souvent utilisé pour représenter un dysfonctionnement dans les actionneurs ou les capteurs du système.

Un dysfonctionnement dans un actionneur donné est modélisé par un changement brusque dans la commande nominale qui prend la forme suivante :

$$u_f = u + (I - \Sigma_A).(\bar{u} - u) \quad (\text{I.2})$$

Avec $\bar{u} \in \mathfrak{R}^m$ est un vecteur (pas nécessairement constant) qu'on ne peut pas manipuler, et où :

$$\Sigma_A = \text{diag}\{\sigma_1^a, \sigma_2^a, \dots, \sigma_m^a\}, \sigma_i^a \in \mathfrak{R}. \quad (\text{I.3})$$

Ainsi, $\sigma_i^a = 0$ représente un défaut total du $i^{\text{ème}}$ actionneur du système de manière à ce que le signal de commande provenant de cet actionneur devienne égal au $i^{\text{ème}}$ élément du vecteur incontrôlable \bar{u} , c'est-à-dire $u_f(i) = \bar{u}(i)$.

D'autre part, $\sigma_i^a = 1$ implique que le $i^{\text{ème}}$ actionneur fonction normalement, c'est-à-dire $u_f(i) = u(i)$.

Le remplacement de la commande nominale u par sa forme en cas de défaut u_f nous donne le modèle d'état (à défauts multiplicatifs) suivant :

$$\begin{cases} \dot{x} = A.x + B.\Sigma_A.u + B.(I - \Sigma_A).\bar{u} \\ y = C.x + D.\Sigma_A.u + D.(I - \Sigma_A).\bar{u} \end{cases} \quad (\text{I.4})$$

Les modèle de la forme (I.4) sont dits à défauts multiplicatifs et ont été largement utilisés dans le domaine de commande tolérante aux défauts. Il est à noter que bien que les défauts multiplicatifs n'affectent pas directement la dynamique du système contrôlé lui-même, ils peuvent affecter d'une manière très significative la dynamique du système en boucle fermée et peuvent affecter même la commandabilité du système [64].

D'une manière pareille, les défauts capteurs représentent des mesures différentes des valeurs réelles des variables de sortie. Ils peuvent être représentés comme suit :

$$y_f = y + (I - \Sigma_s).(\bar{y} - y) \quad (\text{I.5})$$

Ainsi, $\Sigma_s = \text{diag}\{\sigma_1^s, \sigma_2^s, \dots, \sigma_m^s\}, \sigma_i^s \in \mathfrak{R}. \quad (\text{I.6})$

Si $\sigma_i^s = 0$, il s'agit d'un défaut total du $j^{\text{ème}}$ capteur et si $\sigma_j^s = 1$, le $j^{\text{ème}}$ capteur fonctionne correctement. En cas de défaut de capteur, le système (I, 1) devient :

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u \\ y = \Sigma_s \cdot C \cdot x + \Sigma_s \cdot D \cdot u + (I - \Sigma_s) \cdot \bar{y} \end{cases} \quad (\text{I.7})$$

La combinaison des deux types de défauts multiplicatifs (actionneur et capteur) change le modèle (I.1) en :

$$\begin{cases} \dot{x} = A \cdot x + B \cdot \Sigma_A \cdot u + B \cdot (I - \Sigma_A) \cdot \bar{u} \\ y = \Sigma_s \cdot C \cdot x + \Sigma_s \cdot D \cdot \Sigma_A \cdot u + \Sigma_s \cdot D \cdot (I - \Sigma_A) \cdot \bar{u} + (I - \Sigma_s) \cdot \bar{y} \end{cases} \quad (\text{I.8})$$

Le modèle multiplicatif est utilisé pour représenter la majorité des défauts de capteurs et d'actionneurs.

I.4.2. Défauts additifs

La représentation additive des défauts est plus utilisée que la représentation multiplicative, le système d'état prend la forme :

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u + F \cdot f_f \\ y = C \cdot x + D \cdot u + E \cdot f_f \end{cases} \quad (\text{I.9})$$

Où $f_f \in \mathfrak{R}^{n_f}$ est un signal décrivant les défauts. Cette représentation peut en principe être utilisée pour représenter une large classe de défauts y compris les capteurs, actionneurs et composants système. Cependant, l'utilisation du modèle (I.9) induit un signal f_f dépendant de x, y et u . Par exemple, si on veut représenter additivement un défaut total dans tous les actionneurs, (mettre $\Sigma_A = 0$ et $\bar{u} = 0$ dans (I.2)) alors, pour avoir (I.9) équivalent à (I.4) le signal f_f doit vérifier :

$$\begin{bmatrix} F \\ E \end{bmatrix} \cdot f_f = \begin{bmatrix} B \\ D \end{bmatrix} \cdot \Sigma_A \cdot u \quad (\text{I.10})$$

Ce qui rend f_f dépendant de u et complique par conséquent la synthèse du régulateur.

Un autre inconvénient de la représentation additive des défauts des capteurs et des actionneurs est qu'en terme de relation entrée-sortie ces deux défauts sont très difficiles à distinguer. En effet, supposons que le modèle :

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u + f_f^a \\ y = C \cdot x + D \cdot u + f_f^s \end{cases} \quad (\text{I.11})$$

Soit utilisé pour représenter les défauts dans les capteurs et les actionneurs. La fonction de transfert correspondante s'écrit :

$$y(t) = (C(sI - A)^{-1}.B + D).u + C(sI - A)^{-1}.f_f^a + f_f^s \quad (I.12)$$

A partir de cette fonction on remarque que l'effet d'un défaut d'actionneur sur la sortie peut être représenté par f_f^a mais aussi par f_f^s .

Un avantage des modèles additifs est que les défauts sont représentés par des signaux et non par des changements dans les matrices d'état du système, comme c'est le cas pour la représentation multiplicative. C'est pour cela que la majorité des méthodes FDD (détection et diagnostic de défauts) se basent sur la représentation additive [35].

- **Défauts de composants systèmes :**

Les défauts des composants du système sont les plus rencontrés car ils incluent pratiquement tout changement dans quelque élément du système. Ces défauts provoquent des changements dans toutes les matrices de la représentation d'état du système car celles-ci dépendent pratiquement des mêmes paramètres physiques subissant un changement.

Les défauts de composants système sont en général modélisés sous la forme d'un système à paramètres variables.

$$\begin{cases} \dot{x} = A(f).x + B(f).u \\ y = C(f).x + D(f).u \end{cases} \quad (I.13)$$

Où $f \in \mathcal{R}^{n_f}$ est un vecteur représentant les défauts.

I.5. LE DIAGNOSTIC DES DEFAUTS

De façon générale, les défauts dans les processus automatisés causent des réactions indésirables et même l'arrêt du système commandé, et les conséquences pourraient causer des dégâts aux parties techniques du système, au personnel ou à l'environnement. Pour cela, il faut diagnostiquer le système c'est-à-dire, trouver les composants défectueux.

Le mot diagnostic possède de nombreuses significations dans la littérature. En effet, étymologiquement, ce mot vient du grecque et signifie (Dia : par, Gnosis : connaissance).

Une autre définition considère le diagnostic comme un problème de reconnaissance de formes. L'ensemble des états est homologué à un ensemble de classes et le vecteur forme est le vecteur composé par les paramètres observés du système. Cette définition s'applique à de nombreuses applications de diagnostic dans lesquelles le diagnostic est réalisé par des méthodes de reconnaissance de formes [20].

Nous intéressons à la définition technique, c'est-à-dire le diagnostic industriel. Suivant la norme AFNOR, le diagnostic est l'identification de la cause probable de la (ou des) défaillance(s) à l'aide d'un raisonnement logique fondé sur un ensemble d'informations provenant d'une inspection, d'un contrôle ou d'un test [4].

Cette définition dernière résume deux tâches principales : observer les symptômes de la défaillance et identifier la cause à l'aide d'un raisonnement logique fondé sur des observations [81].

L'objectif de la fonction diagnostic est de rechercher les causes et de localiser les organes qui ont entraîné une observation particulière. Dans d'autre terme, le diagnostic est une procédure qui consiste à détecter et localiser un composant ou un élément défectueux selon le principe de comparaison d'informations caractéristiques de l'état courant du système à celles établies en l'absence de défauts. Dans ce que suit, nous allons expliquer en détails les différentes étapes de cette procédure.

I.6. DIFFERENTES ETAPES DE DIAGNOSTIC

La procédure de détection et d'isolation des défauts passe par trois étapes essentielles : La détection, la localisation et l'identification. Ces étapes sont résumées sur la figure suivante comme suit :

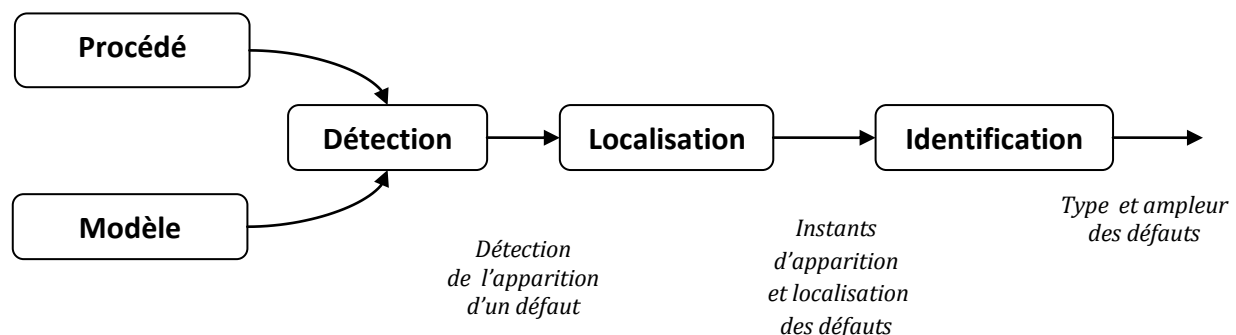


Figure I.4 : Procédure de détection et d'isolation des défauts.

- **Détection**

L'objectif de la procédure de détection est de déterminer l'apparition et l'instant d'occurrence d'un défaut. Le principe est de comparer le comportement du modèle nominal du système avec celui du système réel, c'est-à-dire déterminer si l'état de fonctionnement du système est normal ou anormal, ce qui permet de générer des résidus. La qualité de la détection dépend généralement de deux paramètres essentiels qui sont la fausse alarme et la non détection.

- **Localisation :**

Après avoir détecté la présence d'un défaut, une procédure de localisation est utilisée pour déterminer le composant ou l'élément défectueux. Cette opération porte le nom de localisation ou d'isolation de défauts. On parle de localisation quand on est, de plus, capable de préciser la nature du/des défaut(s) occurrent(s).

- **Identification :**

L'objectif de la procédure est d'identifier la valeur réelle de paramètre en défaut et d'estimer l'instant de l'apparition et l'instant d'occurrence d'un défaut. De plus, l'identification peut comprendre une procédure visant à déterminer la cause du défaut, c'est-à-dire son origine.

I.7. CLASSIFICATION DES METHODES DE DIAGNOSTIC

De façon générale, il y a plusieurs méthodes de diagnostic connues et de nouvelles techniques sont proposées constamment dans la littérature. Ces méthodes sont divisées en deux catégories : les méthodes de diagnostic avec modèle et les méthodes de diagnostic sans modèle.

L'existence d'un modèle formel ou mathématique de l'équipement détermine la méthode de diagnostic utilisée. La première catégorie se base sur le modèle analytique du système à surveiller et utilise généralement des techniques de l'Automatique [18].

En revanche, les techniques de diagnostic sans modèle sont divisées en deux parties. La première partie regroupe les outils de traitement de bas niveau et la deuxième partie est celle des techniques dites de haut niveau.

La première sous catégorie est orientée vers les outils statistiques de traitement du signal qui sont en contact direct avec le signal capteur, et ne servent généralement que pour la génération d’alarmes brutes, sans aucune information concernant leur signification.

Également, la deuxième sous catégorie est orientée vers la communication avec l’opérateur, cette sous catégorie utilise l’Intelligence Artificielle, et serve comme outil de base pour l’aide à la décision, les méthodes de cette catégorie sont plus efficaces, car elles sont capables de détecter, d’interpréter et de diagnostiquer les défauts [17].

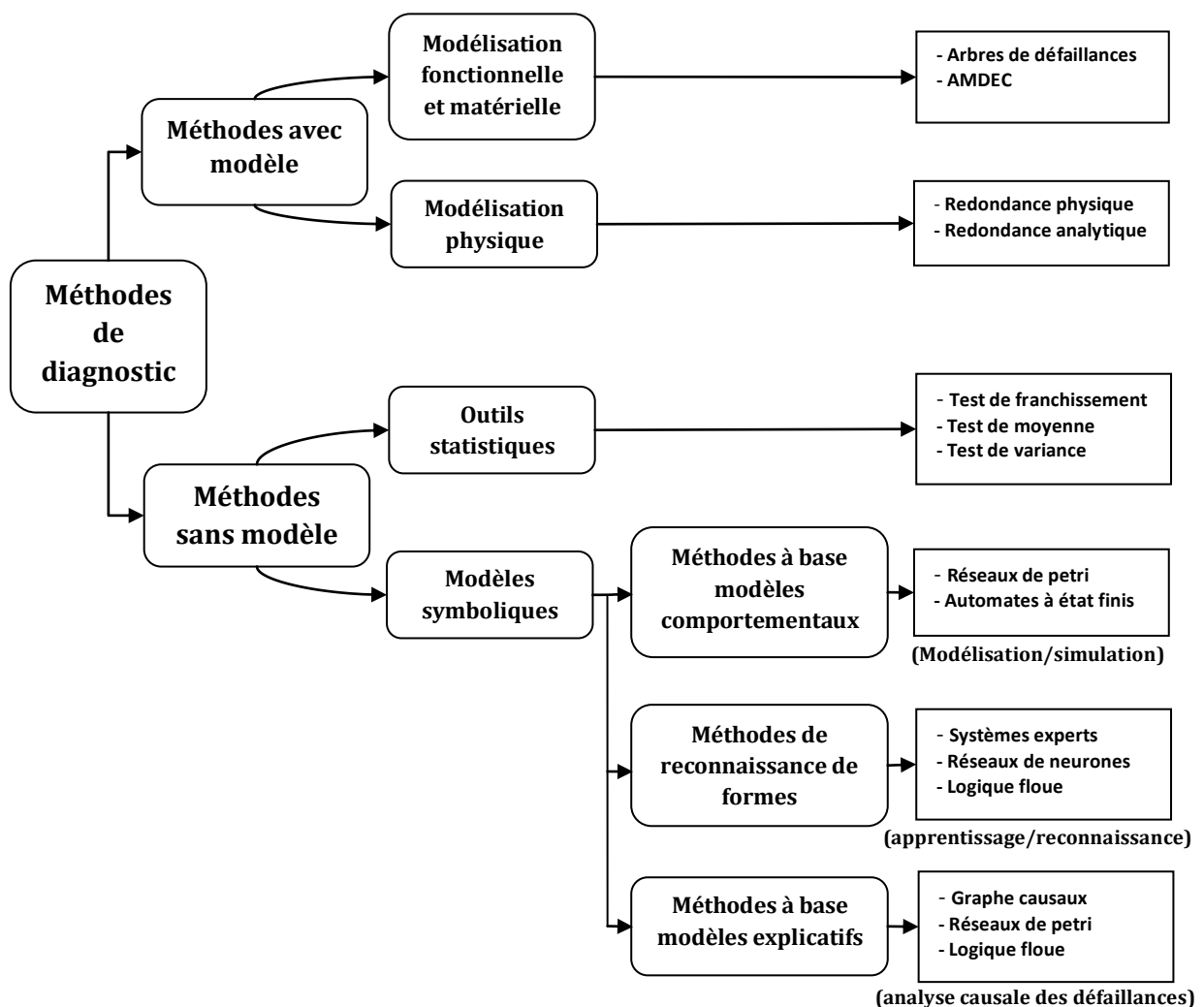


Figure I.5 : Classification des méthodes de diagnostic.

I.7.1. Méthodes de diagnostic sans modèle (à base de données historiques)

Dans le domaine pratique, la plupart des systèmes sont non linéaires et la modélisation est une tâche difficile voir impossible. Alors pour les applications industrielles, les méthodes qui sont applicables pour ce genre de systèmes sont les méthodes sans modèle, appelées aussi méthodes à base de données historiques. Le diagnostic sans modèle se compose essentiellement de deux types : diagnostic avec des tests statistiques du traitement du signal et diagnostic par intelligence artificielle.

I.7.1.1. Diagnostic par outils statistiques

L'idée de base de cette technique de diagnostic par outils statistiques consiste à supposer que les signaux provenant de la chaîne de mesure possèdent certaines propriétés statistiques. On effectue alors quelques tests qui permettent de vérifier si ces propriétés sont présentées dans un échantillon des signaux mesurés. Nous citons les tests les plus importants [77].

1) Test de franchissement de seuils :

Comme le nom indique, ce test consiste à comparer les signaux avec des seuils préétablis. Le franchissement de ce seuil par un des signaux capteurs génère une alarme. Ce type de méthode est très simple à mettre en œuvre mais ne permet pas d'établir un diagnostic de défaillance, cette méthode est aussi très sensible aux fausses alarmes.

2) Test de moyenne :

On note aussi la méthode de test de moyenne, ce test de comparaison est effectué sur la moyenne du signal contenu dans une fenêtre d'un ensemble de valeurs et non sur une valeur ponctuelle.

3) Test de variance :

Un autre outil statistique se base sur le calcul de la variance d'un signal. Tant que cette variance se situe dans une bande située autour de sa valeur nominale, l'évolution du système est supposée normale.

I.7.1.2. Diagnostic par outils symboliques

Les méthodes de diagnostic par outils symboliques se basent sur les techniques de l'intelligence artificielle et font appel à des connaissances symboliques. On distingue donc parmi les modèles symboliques, les méthodes à base de modèles comportementaux, les méthodes de reconnaissance et les méthodes à base de modèles explicatifs [44].

1) Méthodes à base de modèles comportementaux :

Ces méthodes se caractérisent par les termes modélisation et simulation. Elles regroupent notamment des outils tels que :

- les réseaux de petri,
- les automates d'états finis.

Habituellement, ces méthodes sont basées sur la simulation de comportement du fonctionnement du système, à partir d'une modélisation faite sur son comportement, le plus souvent une modélisation de bon fonctionnement qui, contrairement aux modèles numériques, n'est pas basée sur la physique du système mais sur une considération en terme de mode de fonctionnement.

2) Méthodes de reconnaissance de formes :

Ces approches de diagnostic par reconnaissance de formes sont des approches intermédiaires entre les techniques fondées sur l'utilisation d'un système expert et les techniques analytiques. Le principe de fonctionnement des ces méthodes est de représenter tout comportement du système par une classe. Une procédure d'apprentissage permet de déterminer certain nombre de classes. Chacune de ces classes représente un mode de fonctionnement.

Notons que dans ces méthodes, on retrouve des outils tels que :

- les méthodes statistiques de reconnaissance de formes,
 - les réseaux de neurones,
 - la logique floue,
 - les réseaux neuro-flous,
 - les systèmes experts,
 - le raisonnement à partir de cas.
-

3) Méthodes à base de modèles explicatifs :

Ce type regroupe les méthodes qui permettent de fournir une représentation de l'analyse causale des liens entre les défaillances, leurs causes et leurs effets observables. Dans ces méthodes, on considèrera particulièrement des applications basées sur des outils tels que :

- les graphes de fluence,
- les graphes causaux,
- les graphes contextuels,
- les réseaux de petri,
- la logique floue.

I.7.2. Méthodes de diagnostic avec modèle

Les méthodes à base de modèle ont été développées dès le début des années 70. Ce type de méthodes à base de modèle, fait appel à des techniques d'automatique. Ces méthodes ont pour but de comparer les valeurs des variables réelles avec les valeurs calculées en utilisant un modèle qui met en rapport les variables du processus. Elles se composent de deux sous catégories de méthodes : méthodes de diagnostic par modélisation fonctionnelle et matérielle et méthodes de diagnostic par modélisation physique.

I.7.2.1. Méthodes de diagnostic par modélisation fonctionnelle et matérielle

Les méthodes les plus couramment rencontrées sont :

- l'Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticité (AMDEC).
- les Arbres De Défaillances (ADD).

Le principe de ces méthodes consiste à établir a priori et de la manière la plus complète possible, les liens entre les causes initiales des défaillances et leurs effets mesurables.

I.7.2.2. Méthodes de diagnostic par modélisation physique

Une autre catégorie de méthodes analytiques est représentée par les méthodes basées sur une modélisation physique des processus surveillés. Ces méthodes sont les méthodes les plus familières aux automaticiens. Le principe de ces méthodes est de comparer les mesures effectuées sur le système avec les informations fournies par le modèle analytique [25]. Tout écart

représente alors une défaillance, ce qui explique la nécessité des outils de la théorie de la décision utilisés pour déterminer si cet écart est dû à des aléas normaux, comme par exemple le bruit de mesure ou s'il traduit une défaillance du système. Généralement, ces méthodes peuvent être séparées en deux catégories :

- techniques de redondance physique et analytique,
- les techniques d'estimation paramétrique.

1) Redondance physique

Pour s'assurer de la validité d'une mesure et distinguer les défaillances capteurs des défaillances système, une méthode simple consiste à mesurer la même grandeur physique, à l'aide de différents capteurs, d'où le nom "Redondance physique" ou "Redondance matérielle". Ensuite, on considère qu'une défaillance capteur s'est produite dans au moins une des composantes, si les composantes identiques placées dans le même environnement émettent des signaux distincts, et dans le cas contraire, on considère que ces composants sont dans un état de fonctionnement nominal [78]. Ce type de redondance a été très couramment employé dans les domaines à haut niveau de risque, notamment dans le cas de l'aéronautique ou le nucléaire. Cette méthode a l'avantage d'être relativement efficace et conceptuellement simple mais coûteuse à être mise en œuvre et ne s'applique généralement que sur des capteurs. Un autre inconvénient est que les composantes identiques placées dans le même environnement peuvent rencontrer des défauts en même temps et de la même façon.

2) Redondance Analytique

Par opposition à la redondance physique, la redondance analytique est basée sur l'utilisation des modèles analytiques qui sont une représentation mathématique des lois d'évolution des variables physiques du système, ce qui permet de comparer le comportement observé et le comportement du système. Cette comparaison fournit alors une quantité appelée "résidu" qui va assister à déterminer si le système est dans un état défaillant ou non. Parmi les différentes méthodes de détection utilisant des modèles mathématiques, nous trouvons principalement : l'espace de parité et les observateurs.

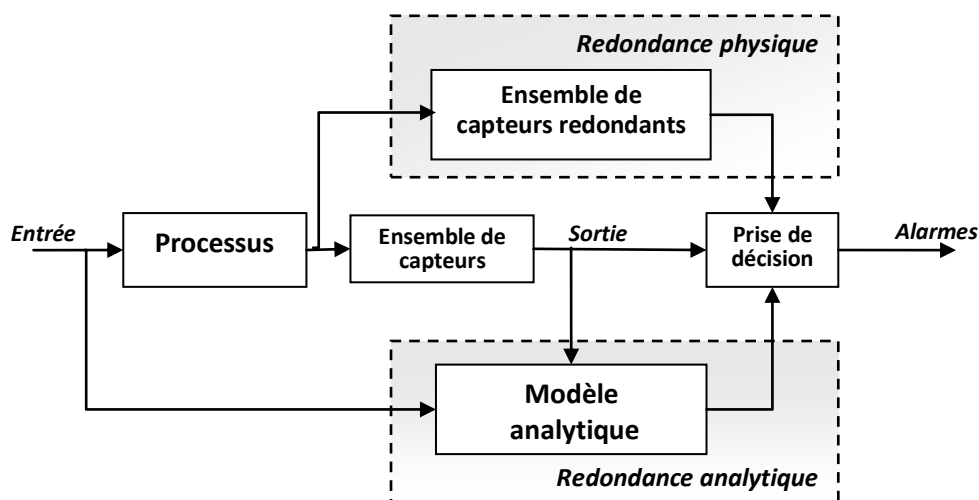


Figure I.6 : Architecture de redondance physique et analytique.

3) Méthodes d'estimation paramétrique

L'idée de base de la méthode d'estimation paramétrique est de supposer qu'il ya un modèle paramétrique qui peut décrire le comportement du système et que les valeurs de ces paramètres en fonctionnement nominal soient connues. Pour détecter l'apparition de défaillances dans le système, le principe consiste alors à estimer en continu les paramètres du système à partir de mesures des entrées et des sorties et en l'évaluation de la distance qui les sépare des valeurs de référence de l'état normal du procédé. La théorie de la décision est pareille comme pour les méthodes de redondance analytique.

La différence entre les méthodes de redondance analytique et les méthodes d'estimation paramétrique est qu'on effectue, pour les premières, la comparaison entre l'état estimé et l'état théorique du système, alors que pour les secondes, on compare les paramètres estimés avec les paramètres théoriques du système.

I.8. PROCEDURE DE DIAGNOSTIC A BASE DE MODELE

Le diagnostic à base de modèle est largement présent dans la littérature pour des applications industrielles [31]. Ces méthodes s'appuient sur la comparaison du comportement observé et du comportement attendu (prédit par un modèle) du système.

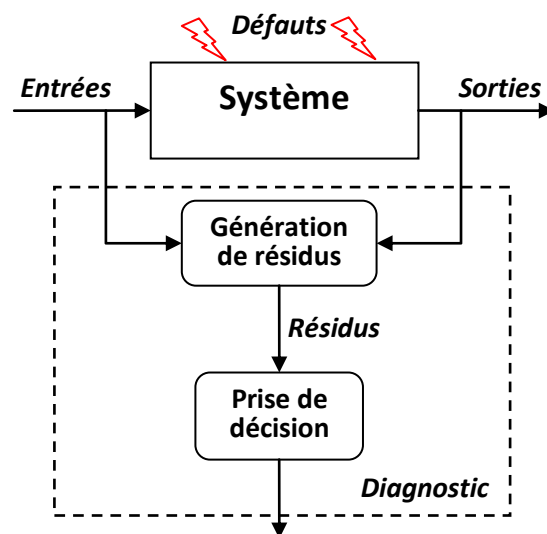


Figure I.7 : Diagnostic à base de modèle.

Nous nous intéresserons aux méthodes de diagnostic à base de modèle (*Model-based-FDI : Fault Detection and Isolation*) qui utilisent un modèle et reposent sur la génération et l'étude d'un signal particulier appelé "indicateur de défaut" ou "résidu". Ces méthodes ont un rôle très important dans la reconfiguration de la commande.

Comme le montre la figure (I.7), ce type de méthodes est généralement réalisé en deux étapes : génération des résidus puis analyse et évaluation des résidus grâce à un système de décision. Le rôle de ce système de décision est de déterminer si le résidu est significatif pour décider de l'existence d'un défaut.

I.8.1. Génération de résidus

Cette première phase consiste à produire des indicateurs de défauts (résidus) qui sont des signaux résiduels reflétant l'écart entre les mesures des signaux du procédé, actionneurs ou capteurs, et la valeur théorique fournie par le modèle dans des conditions de fonctionnement normal. Les résidus contiennent alors des informations sur les anomalies ou dysfonctionnements du système à surveiller. La qualité de la génération de résidus est un élément essentiel pour garantir les performances d'un système de diagnostic.

Plusieurs méthodes existent pour la génération de résidus. Ces méthodes peuvent être subdivisées en deux approches principales :

- ✓ **Approche quantitative** : Observateur, filtre de Kalman, espace de parité, estimation paramétrique...
- ✓ **Approche qualitative** : Logique floue, réseaux de neurones, reconnaissance de forme, système expert...

I.8.2. Prise de décision

Cette deuxième phase consiste à analyser les résidus pour détecter et localiser la présence ou non d'un ou de plusieurs défauts sur le procédé et éventuellement isoler l'élément défaillant. Donc cette étape permet de classer l'infinité des comportements possibles des résidus en plusieurs catégories représentant les différentes situations de défaut. La qualité de la détection dépend généralement de deux paramètres essentiels qui sont la fausse alarme et la non détection.

On parle de localisation quand on est, de plus, capable de préciser la nature du ou des défauts occurrents. Comme pour la génération de résidus, il existe plusieurs méthodes permettant la satisfaction de cette tâche.

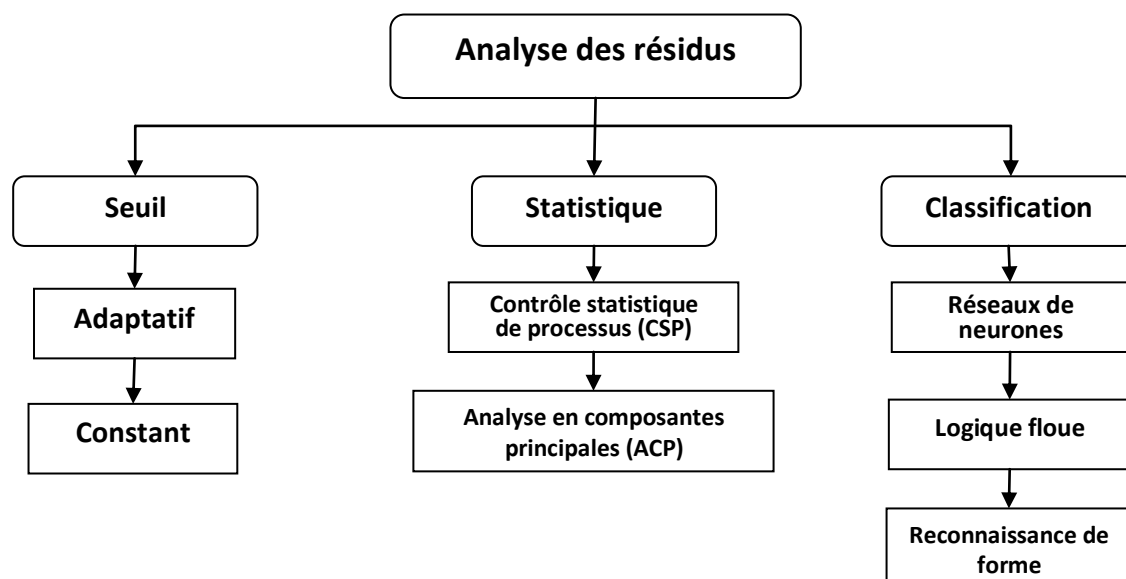


Figure I.8 : Méthodes d'analyse des résidus.

I.9. CRITERES DE PERFORMANCE D'UN SYSTEME DE DIAGNOSTIC

Pour pouvoir assurer qu'un système de diagnostic est performant, il s'agit de présenter les principaux critères permettant d'évaluer les performances d'un système de diagnostic. D'une manière générale, nous pouvons regrouper les différents critères de performance du système de détection de la manière suivante [37,24]. :

- **Déteçtabilité**

La déteçtabilité est l'aptitude du système de diagnostic à pouvoir déceler la présence d'une défaillance sur le procédé. Elle est fortement liée à la notion d'indicateurs de défauts (résidus). Elle résulte d'un compromis entre le taux de fausse alarme et celui du non détection.

- **Isolabilité**

L'isolabilité est la capacité du système de diagnostic à remonter directement à l'origine du défaut. Il faut noter qu'une alarme engendre bien souvent de nouvelles alarmes, ce qui rend une telle fonction complexe et il devient dès lors difficile de retrouver l'organe défaillant. La propriété d'isolabilité est liée à la structure des résidus et à la procédure de détection elle-même.

- **Sensibilité**

La sensibilité caractérise l'aptitude du système à détecter des défauts de certaine amplitude. Elle dépend non seulement de la structure des résidus mais aussi du rapport de l'amplitude du bruit de mesure avec celle du défaut.

- **Robustesse**

La robustesse détermine la capacité à détecter des défauts indépendamment des erreurs de modélisation, (Sensibilité du résidu aux défauts et insensibilité vis-à-vis des perturbations). Généralement, la robustesse est définie par rapport à toutes les entrées inconnues.

- **Le coût économique et le temps de développement**

En pratique, d'autres critères sont à prendre en considération. En phase d'industrialisation, les contraintes économiques sont essentielles. La rapidité de détection peut être un facteur déterminant. De même, les coûts économiques vont conditionner la stratégie de diagnostic.

I.10. CONCLUSION

Dans ce chapitre, nous avons présenté les différentes méthodes utilisées dans le domaine de diagnostic de défauts. Ces méthodes sont généralement divisées en deux groupes : méthodes de diagnostic avec modèle et sans modèle. Les premières se basent sur l'existence d'un modèle formel de l'équipement et utilisent généralement les techniques de l'automatique. La deuxième catégorie est plus intéressante dès lors qu'un modèle de l'équipement est inexistant ou difficile à obtenir. Dans ce cas, on utilise les outils de la statistique et de l'intelligence artificielle. Le choix d'une de ces méthodes dépend essentiellement des connaissances disponibles sur le procédé, sans oublier les considérations techniques et économiques.

On peut conclure aussi que le problème du diagnostic consiste à générer des résidus qui doivent être nuls en fonctionnement normal, et assez sensibles à tout défaut affectant le système à surveiller d'une part, et d'autre part à analyser ces résidus pour détecter la présence d'un défaut, et localiser l'élément défaillant.

Dans le chapitre suivant, nous présenterons la problématique de la tolérance aux défauts. Nous présenterons aussi les différentes techniques utilisées dans le domaine de commande tolérante aux défauts.

Chapitre II

Commande tolérante des défauts

- II.1. Introduction*
 - II.2. Positionnement du problème*
 - II.3. Diagnostic et la tolérance aux défauts*
 - II.4. Fondements et motivations*
 - II.5. Classification des approches FTC*
 - II.6. Différentes méthodes FTC*
 - II.7. Conclusion*
-

II.1. INTRODUCTION

Ce chapitre a pour but de présenter les principaux concepts de la commande tolérante aux défauts. La classification des méthodes résultantes, permet d'effectuer un état de l'art de l'ensemble des méthodes de commande tolérante aux défauts afin de synthétiser des systèmes tolérants aux défauts.

Le problème de la commande tolérante aux défauts a été largement abordé ces deux dernières décennies et a fait l'objet d'un nombre important de résultats expérimentaux et de publications. Dans le domaine industriel, la commande tolérante aux défauts à été implantée dans divers applications militaires et civiles pour accommoder l'effet des défauts dans différents domaines tels que les systèmes de contrôle des avions, la technologie d'espace, les véhicules sous-marins, les centrales nucléaires et aussi les systèmes de guidage des missiles.

II.2. POSITIONNEMENT DU PROBLEME

L'idée de base d'un système de commande tolérant aux défauts est de maintenir la stabilité et les performances du système à commander, en dépit de la présence des défauts qui peuvent le faire dévier de son point de fonctionnement nominal. En d'autre terme, la stratégie de ce type de commande a pour but de concevoir un dispositif de commande particulier permettant de conserver la maîtrise du comportement dynamique du système commandé non seulement en l'absence de défauts, mais également en présence de composants défectueux. Dans ce dernier cas, ce dispositif peut permettre aussi de conduire le système de commande vers un autre point de fonctionnement, proche du point de fonctionnement nominal, c'est le cas de fonctionnement en mode dégradé.

Les techniques classiques de commande (commande linéaire, quadratique, robuste ou adaptative) restent insuffisantes et il est nécessaire de faire appel à des techniques qui prennent en considération l'interaction entre le système de détection des défauts et la loi de commande.

Le principe de reconfiguration est illustré sur la figure (II.1). La procédure est très simple. L'apparition d'un défaut provoque un changement du point de fonctionnement nominal de système, défini par les variables entrées/sorties (u_0, y_0) vers un autre point de fonctionnement (u_f, y_f) .

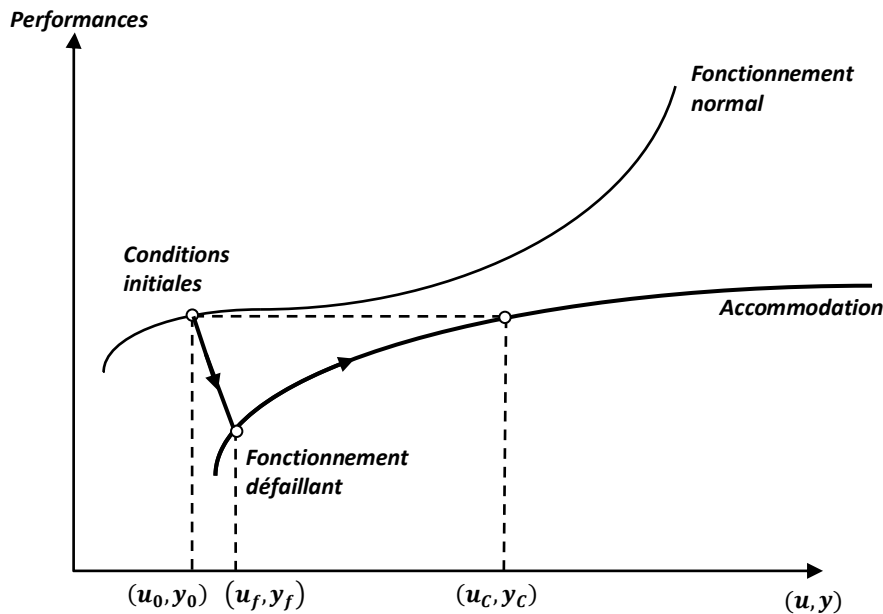


Figure II.1 : Problème de la commande tolérante aux défauts.

Dans ce cas précis, le but de la commande tolérante est de concevoir une nouvelle loi de commande qui prend en considération les variations dues à l'apparition du défaut et permet d'amener le système, soit vers l'état de fonctionnement nominal, soit vers un autre point de fonctionnement défini par (u_c, y_c) pour lequel les performances ne sont pas les mêmes qu'initialement mais en sont acceptables [50].

II.3. DIAGNOSTIC ET LA TOLERANCE AUX DEFAUTS

La plupart des travaux de recherche dans le domaine de la commande tolérante aux défauts ont été développés en supposant l'existence d'un module de diagnostic fonctionnant d'une façon intégrative avec un module de reconfiguration des lois de commande.

Le diagnostic s'intègre dans une tâche plus générale qui est la surveillance, cette dernière a pour but de détecter et localiser les défauts qui peuvent affecter les performances à partir des informations disponibles sur le système (commande, mesures). Il est même nécessaire, dans certains cas, de déterminer la nature de défaut. Donc, il est alors possible de générer des alarmes.

La tolérance aux défauts représente une fonction supérieure au module de diagnostic. La structure d'un système tolérant aux défauts permet de garantir la stabilité et le bon fonctionnement du système à commander dans la présence de défauts.

La figure (II.2) représente un schéma général d'un système tolérant aux défauts muni d'un système de diagnostic. Il comprend :

- Un système commandé muni de son correcteur, de ses actionneurs et de ses capteurs,
- Une module de détection et diagnostic de défauts surveille le système et donne au superviseur des informations sur l'état du système,
- Un superviseur qui, en cas de défaut, exploite ces informations et redéfinit les consignes et les modifications à apporter aux lois de commande tolérante aux défauts.

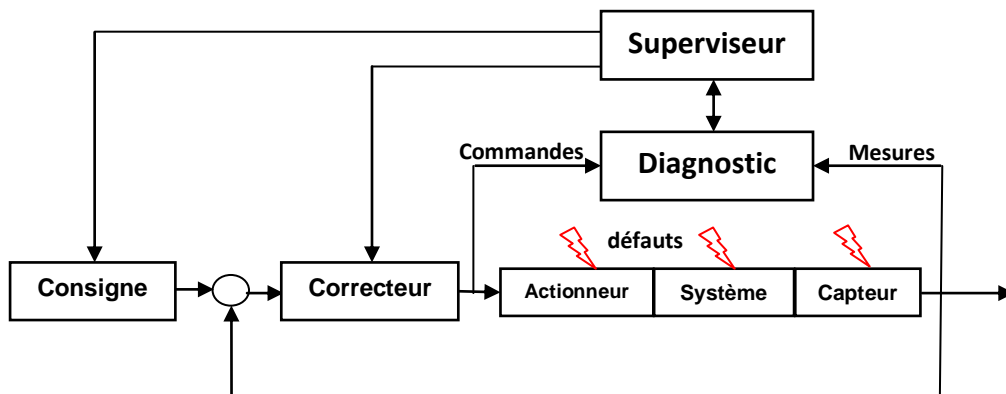


Figure II.2 : Système muni d'un module de diagnostic et d'une commande tolérante aux défauts.

II.4. FONDEMENTS ET MOTIVATIONS

Un système tolérant aux défauts représente une stratégie de synthèse de système de commande à très haute fiabilité. Ce terme regroupe les différentes techniques récentes développées afin d'accroître la fiabilité et la disponibilité et réduire les risques pour des systèmes opérant dans des environnements hostiles. Les systèmes tolérants aux défauts sont conçus pour être capables de s'accommoder avec les défauts dès leurs apparitions et avec une réactivité maximale, de sorte qu'un défaut mineur n'évolue pas en défaut critique au niveau du système global.

Les systèmes tolérants aux défauts sont utilisés dans différentes applications telles que les réacteurs nucléaires, aéronautique, réseaux mobiles de communications...etc. Par exemple, dans les centrales nucléaires, l'occurrence d'un défaut peut générer une catastrophe écologique/humaine. Une action de commande appropriée pourrait être, dans ce cas, l'arrêt du système en situation d'urgence.

On prend par exemple la catastrophe de Tchernobyl. Cet accident est un accident nucléaire qui s'est produit le 26 Avril 1986 dans la centrale nucléaire Lénine en Ukraine. Une explosion énorme au niveau d'une centrale nucléaire, suivie d'une fusion progressive du réacteur N°4. La raison principale de cette tragédie a été l'absence d'un mécanisme d'accommodation aux défauts [9].

Un autre exemple est la fusée Ariane 5. Cette dernière a explosé en plein vol, le 4 Juin 1996, 37 secondes après son décollage. Cela a été causé par un défaut software au niveau de l'Unité de Référence Inertiel (URI) qui a fourni des Informations erronées au système de contrôle. Ces dernières étaient relatives à la trajectoire et à la vitesse de la fusée [10].

Les conséquences catastrophiques de ces pannes auraient pu être évitées, ou du moins atténuées, si le système de commande était conçu de manière à intégrer un certain degré de tolérance aux défauts [40,1].

II.5. CLASSIFICATION DES APPROCHES FTC

De façon générale, Les systèmes de commande tolérante aux défauts peuvent être divisés en deux classes :

- Les systèmes de commande tolérante aux défauts active (*Active Fault Tolerant Control system, AFTCS*).
- Les systèmes de commande tolérante aux défauts passive (*Passive Fault Tolerant Control system, PFTCS*).

Les méthodes passives sont équivalentes aux méthodes de synthèse de loi de commande robuste. Les régulateurs dans ce cas sont synthétisés de façon à être robuste à certains défauts [49,23]. Les méthodes actives de la commande tolérante aux défauts (FTC) intègrent un mécanisme de diagnostic pour détecter et localiser les défauts pouvant affecter le système. Si un défaut est détecté par le module de diagnostic, la stratégie de commande tolérante aux défauts est activée via un mécanisme de reconfiguration.

II.5.1. Approches passives

Les méthodes passives s'appuient sur l'idée de synthèse d'une commande permettant de rendre le système insensible à certains défauts connus a priori. De façon générale, ces approches sont basées sur les techniques de synthèse d'une loi de commande robuste par rapport aux incertitudes paramétriques et perturbations extérieures (commande H_∞ , commande à mode glissant...). Ce type d'approche ne nécessite aucun schéma de détection de défauts ni aucune reconfiguration de loi de commande et ne requière pas d'information en ligne sur le défaut.

Ce point dernier représente un avantage par rapport aux systèmes actifs, puisque l'on obtient une certaine autonomie. Le système continuera à travailler avec la même commande et la même structure. Néanmoins, le domaine d'application reste limité à cause de ses inconvénients [60,33] :

- Pratiquement, l'utilisation seule de la commande robuste ne suffit pas. Un régulateur non intelligent, sans connaissance sur l'occurrence de défaut et sans utilisation de module de diagnostic, le système passif a alors une faible capacité de tolérance aux défauts.
- Assurer la robustesse vis à vis des défauts, ne peut se faire sans réduire les performances nominales du système. Les défauts sont des phénomènes qui arrivent rarement, il n'est pas raisonnable de dégrader les performances nominales du système pour un ensemble très réduit de défauts.

II.5.2. Approches actives

Au contraire des méthodes passives, les méthodes actives de commande tolérante aux défauts utilisent les techniques d'ajustement en temps réel en reconfigurant les lois de commande tout en préservant la stabilité et les performances du système. L'utilisation de l'un de ces méthodes permet alors de traiter les défauts imprévus mais nécessite une technique de détection et d'isolation des défauts capable aussi de fournir des informations précises sur les défauts éventuels (instant d'apparition, type, amplitude...).

La figure (II.3) présente un schéma typique de systèmes de commande tolérante aux défauts avec quatre principaux composants [54,33] :

- un bloc de diagnostic de défauts (*Fault Detection and Diagnosis, FDD*) donnant des informations en temps réel,
- un mécanisme de reconfiguration,
- un régulateur reconfigurable
- et une référence appliquée au système.

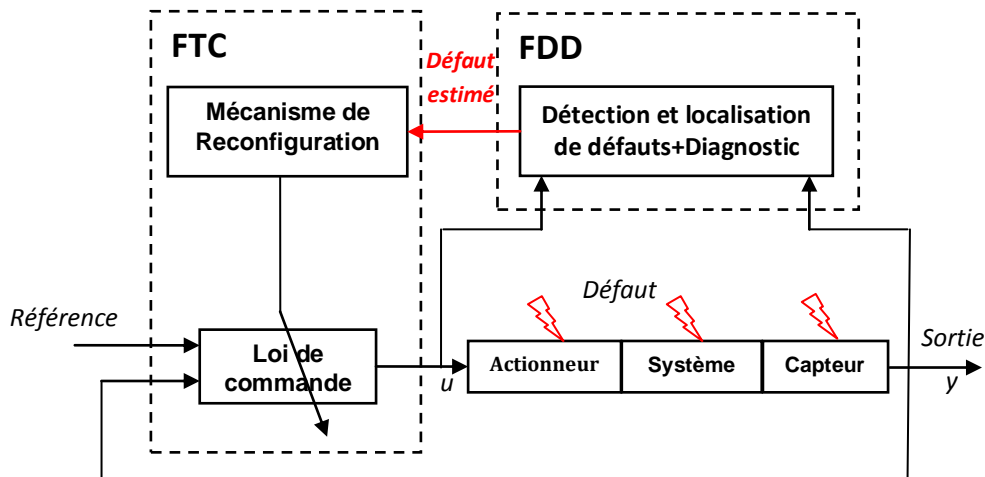


Figure II.3 : Schéma de principe d'une loi de commande FTC active.

Comme il est montré dans la figure (II.4), les méthodes actives sont généralement classifiées en deux sous classes : la reconfiguration du système et l'accommodation de défauts [62].

Les approches dites de reconfiguration du système sont utilisées dans le cas où les relations entrées/sorties entre la loi de commande et le système à contrôler sont modifiées à travers un changement des paramètres et de la structure de la loi de commande, c'est-à-dire la structure du système à contrôler est modifiée de façon à compenser l'effet du défaut (utilisation de redondance matérielle, logicielle...).

Lorsque l'on parle d'accommodation de défauts, les relations entrées/sorties entre la loi de commande et le système à contrôler restent inchangées. On distingue deux méthodes : les méthodes à base de projection et les méthodes de re-conception en ligne. Dans le cadre de la première méthode, il s'agit de sélectionner la loi de commande adéquate à partir d'un ensemble de correcteurs pré-calculés hors ligne. Les méthodes de re-conception en ligne impliquent un calcul en ligne des paramètres du correcteur (commande reconfigurable) ou une commande restructurable qui représente une nouvelle synthèse de la loi de commande modifiant ainsi la structure et les paramètres de cette dernière.

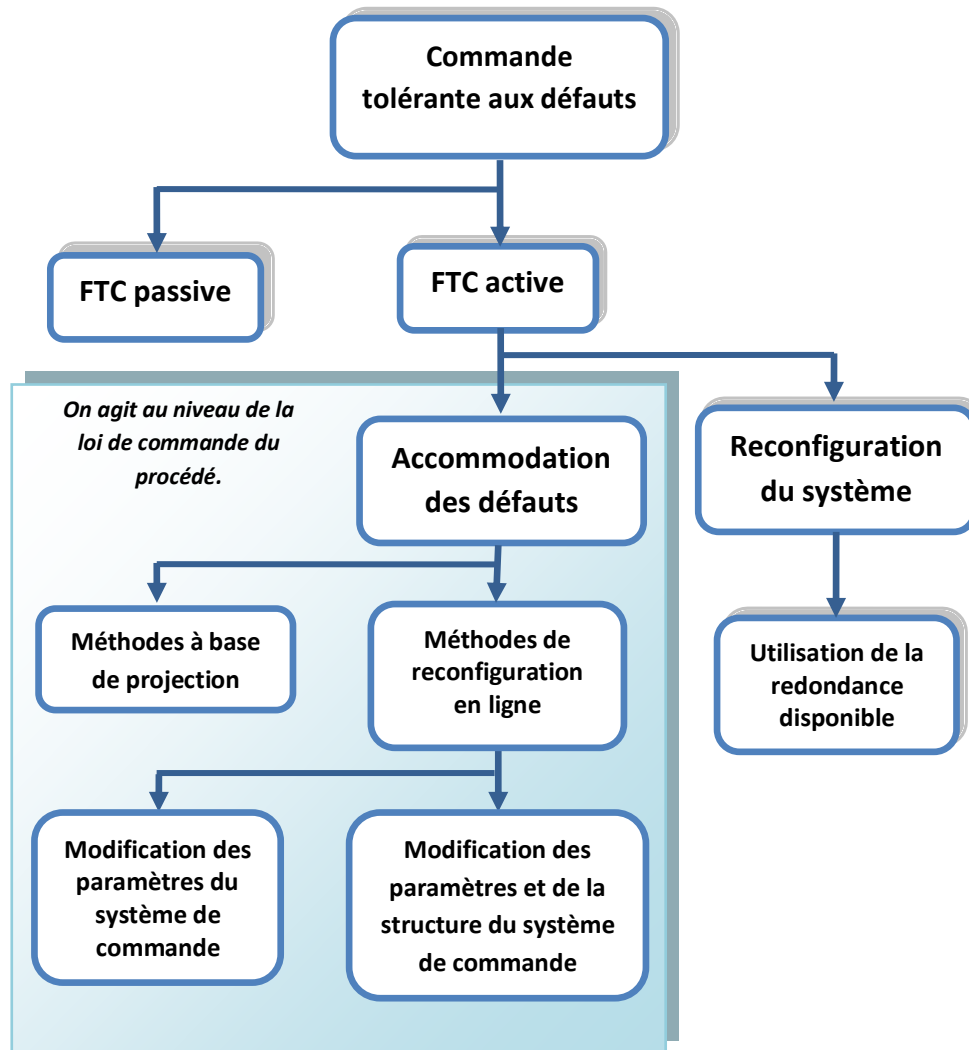


Figure II.4 : Classification des approches FTC.

II.6. DIFFERENTES METHODES FTC

II.6.1. Techniques FTC passives

Dans ces méthodes, le système est conçu de façon à tolérer un nombre limité de défauts supposés connus avant la phase de synthèse de la loi de commande. Dans la littérature, on trouve plusieurs outils de synthèse de la loi de commande robuste. De nombreuses études utilisant les techniques de synthèse de loi de commande robuste basées sur la minimisation d'un critère, ont été menées pour la conception de stratégies passives de la commande tolérante aux défauts.

Parmi les méthodes passives de commande tolérante aux défauts, nous présentons brièvement les techniques suivantes :

- La commande linéaire gaussienne (LQG).
- La synthèse H_∞ .
- Inégalités Matricielles Linéaires (LMI).
- Modes glissants.

Une méthodologie basée sur la minimisation d'un critère LQG, est utilisée dans [32], pour synthétiser le correcteur de commande tolérante aux défauts. Les auteurs proposent de modéliser l'effet des défauts sur le système par un processus aléatoire. Pareillement dans [49], l'idée est d'utiliser la para-métrisation de Youla comme degré de liberté pour atteindre les performances FTC requises. Le problème est résolu à l'aide de la technique «loop shaping» de la commande robuste H_∞ . La minimisation d'un critère H_∞ est aussi le point initial autour duquel les travaux de [73] et [75] sont développés. La méthode de synthèse mise en jeu est basée sur les inégalités matricielles linéaires (LMI), ou sur la résolution d'équations algébriques de Riccati [74]. La méthodologie proposée dans [41]. et [42] consiste en la synthèse d'un régulateur à quatre degrés de liberté [48,69]. La méthodologie implique donc la gestion d'un compromis entre les performances de régulation et de diagnostic. Même si l'idée de départ est séduisante d'un point de vue conceptuel, ce type de solution reste pour le moins discutable car cela conduit à détériorer les performances du système commandé en fonctionnement nominal.

II.6.2. Techniques FTC active

Les méthodes de commande tolérante actives sont plus développées dans la littérature que les méthodes passives grâce à leurs meilleures performances et leurs capacités de traiter une large classe de défauts. Une vue d'ensemble des techniques utilisées est donnée dans la suite :

1) Méthode de la pseudo inverse

La méthode pseudo inverse « *pseudo-inverse method* » (*PIM*) est l'une des méthodes les plus citées dans le domaine de commande active tolérante aux défauts. Cette méthode a été développée initialement dans [27]. Le principe repose sur la minimisation d'une norme de Frobenius conduisant à la détermination du gain de la commande.

Soit le système en boucle fermée, dont le modèle nominal est défini par une représentation d'état de la forme :

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \quad (\text{II.1})$$

Avec $A \in \mathfrak{R}^{n \times r}$, $B \in \mathfrak{R}^{n \times m}$ et $C \in \mathfrak{R}^{q \times r}$ représentent les matrices du système dans le cas nominal.

Sous l'hypothèse que le vecteur d'état est connu, on considère que le système est commandé par un contrôleur par retour d'état $u = -k.x$, $K \in \mathfrak{R}^{m \times n}$. Le système en boucle fermée est :

$$\begin{cases} \dot{x} = (A - BK).x \\ y = C x \end{cases} \quad (\text{II.2})$$

L'apparition d'un défaut conduit à un changement du modèle. Le système en défaut se représente de la manière suivante :

$$\begin{cases} \dot{x}_f = A_f x_f + B_f u_f \\ y_f = C_f x_f \end{cases} \quad (\text{II.3})$$

L'idée de base de cette méthode est de modifier la loi de commande par retour d'état de telle sorte que la dynamique du système défaillant en boucle fermée soit approximativement égale à celle du système nominal en boucle fermée. Nous considérons une nouvelle matrice de gain K_f :

$$u_f = -K_f x_f \quad (\text{II.4})$$

telle que :

$$A - BK = A_f - B_f K_f \quad (\text{II.5})$$

$$\text{d'où :} \quad K_f = B_f^+(A_f - A + B K) \quad (\text{II.6})$$

B_f^+ Étant la matrice pseudo inverse de B_f .

Cette approche a l'avantage d'être simple ce qui la rend très appropriée à une implémentation en ligne, et d'autre part, la possibilité d'exprimer le changement dans toutes les matrices d'état en cas de défaut. Le principal inconvénient relatif à cette méthode réside dans le fait que la loi de commande optimale ne garantit pas toujours la stabilité en boucle fermée du système défaillant [27]. Pour palier ce problème, la méthode pseudo-inverse modifiée a été

développée [26].. D'autres modifications et améliorations de cette méthode ont été proposées dans la littérature (voir par exemple [16 ,63].).

2) Méthode de placement de structure propre

La méthode de placement de structure propre est une méthode plus intuitive que la méthode du pseudo inverse. Pour la synthèse de la loi de commande tolérante aux défauts, cette méthode a été introduite initialement dans [6] et a fait l'objet de plusieurs publications (voir par exemple [35,68]). L'objectif de cette méthode est de ramener la structure propre (c'est à dire les vecteurs et les valeurs propres) de la matrice A du modèle en boucle fermée en cas de défauts à la structure propre nominale. L'idée principale est d'affecter exactement certaines valeurs propres dominantes et en même temps de minimiser l'écart entre les vecteurs propres correspondants.

Dans le cas de retour d'état, on définit λ_i et v_i , $i = 1, 2, \dots, n$, les valeurs propres de la matrice d'état du système en boucle fermée et les vecteurs propres correspondants respectivement. Le rôle est de calculer le nouveau gain de retour d'état K_r pour le modèle en défaut (II.3), solution du problème d'optimisation suivant :

$$\begin{cases} \text{Trouver } K_r & \text{tel que} \\ (A_f + B_f K_r)v_i^f = \lambda_i v_i^f & i = 1, 2, \dots, n \\ \text{et } v_i^f = \arg_{v_i^f} \min \|v_i - v_i^f\|_{W_i}^2 \end{cases} \quad (\text{II.7})$$

Où $\|v_i - v_i^f\|_{W_i}^2 = (v_i - v_i^f)^T W_i (v_i - v_i^f)$ et W_i est une matrice de pondération définie positive servant de degré de liberté supplémentaire.

L'avantage de cette approche comparée à la méthode de la pseudo inverse, réside dans le fait que la solution, si elle existe, assure la stabilité de la boucle fermée du système défaillant. En revanche, l'inconvénient vient du fait que les erreurs de modèles ne sont pas aisément incorporables dans le calcul d'optimisation [64].

3) Loi de commande par séquençement

La loi de commande par séquençement de gain a été largement appliquée au domaine de l'aéronautique où un correcteur linéaire invariant unique ne peut remplir l'ensemble des objectifs de performance et de robustesse sur la totalité du domaine de fonctionnement (situation normale et défaillante).

Un système muni d'une loi de commande séquencée peut être vu comme un système avec une commande de retour d'état dont les gains sont ajustés par compensation. Dans les applications d'aéronautique, l'utilisation des mécanismes de FDI, l'estimation d'état pour reconfiguration du régulateur et les lois de commande pré-calculées sont les principaux dispositifs de la loi de commande par séquencement.

A partir d'une structure d'une loi de commande tolérante fixe, cette méthode permet de modifier la valeur des gains du correcteur en fonction de certains paramètres physiques variant avec le temps (vitesse, altitude, masse, etc.) et/ou en fonction d'un résultat de diagnostic [60,33].

4) Approche par modèle de référence

L'approche par modèle de référence est l'une des méthodes de commande active tolérante aux défauts qui permet la synthèse d'une nouvelle loi de commande telle que les performances du système défaillant commandé s'approche le plus possible de celles d'un modèle de référence, au sens d'un critère. Généralement, la méthode considère un modèle de référence de la forme :

$$\begin{cases} \dot{x}_m = A_m x_m + B_m r_m \\ y_m = x_m \end{cases} \quad (\text{II.8})$$

Où r_m correspond au signal de trajectoire de référence, la commande par retour d'état u définie par :

$$u = K_r r + K_x x \quad (\text{II.9})$$

Donc, le but est de calculer les matrices de gain K_r et K_x . Le système défaillant est donné par la représentation d'état suivante :

$$\begin{cases} \dot{x} = A_f x + B_f u \\ y = C_f x \end{cases} \quad (\text{II.10})$$

où $u \in \mathfrak{R}^m$ correspond aux entrées de commande, $y \in \mathfrak{R}^p$ aux sorties mesurées, et $x \in \mathfrak{R}^n$ aux états du système et où A_f , B_f et C_f sont des matrices d'état dépendant du vecteur de paramètre f qui correspond à un vecteur de paramètres dont la variation autour de sa valeur nominale traduit l'effet des défauts considérés. On cherche alors les matrices K_r et K_x telles que le système défaillant (II.10) coïncide au modèle de référence (II.8) en boucle fermée.

Il vient alors que :

$$\begin{cases} K_x = (C_f B_f)^{-1} \cdot (A_m - C_f A_f) \\ K_r = (C_f B_f)^{-1} \cdot B_m \end{cases} \quad (\text{II.11})$$

Il est clair d'après le résultat (II.11) que la solution existe si le système est carré c'est-à-dire $Dim(y) = Dim(u)$ et que l'inverse de la matrice $C_f \cdot B_f$ existe. Lorsque les matrices du système (II.11) sont inconnues, elles peuvent être remplacées par des valeurs estimées \hat{C}_f, \hat{B}_f , provenant de la méthode indirecte [12]. Cette méthode dernière ne garantit pas toujours la stabilité en boucle fermée. Pour éviter ce problème, une technique dite directe (ou implicite) a été utilisée pour calculer directement les matrices de gain des correcteurs K_r et K_x à partir d'une méthode adaptative [12,30].

5) Méthode basée sur banc de régulateurs

La méthode basée sur un banc de régulateurs est l'une des méthodes à accommodation active où seuls les paramètres de la commande sont modifiés suite à l'occurrence d'un défaut. Le principe de cette méthode est basé sur l'existence d'un banc de régulateurs pré-calculés. Chaque régulateur représente un mode de fonctionnement. Un régulateur pour le mode de fonctionnement nominal et un régulateur pour chaque mode défaillant.

Un superviseur constitué d'un ensemble d'estimateurs, permet de sélectionner le régulateur associé au mode de fonctionnement actif. Ces estimateurs permettent la reconstruction des sorties du système quelque soit son mode de fonctionnement (défectueux ou non). Après avoir évalué les performances de chaque mode, le régulateur concerné est sélectionné comme illustré à la figure (II.4).

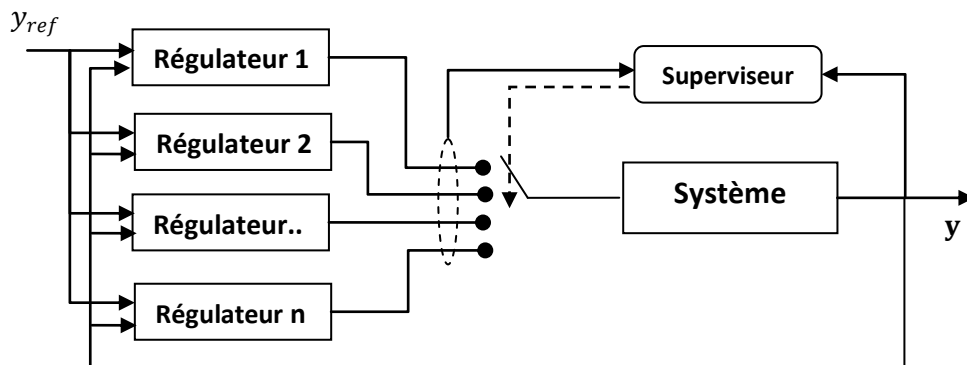


Figure II.5 : Structure du système tolérant aux défauts actif à base de banc de régulateurs.

Cette approche était le sujet de plusieurs travaux notamment [43,13]. Dans le cas où le système est représenté avec un nombre de modes de défaillances relativement petit ou les modèles sont parfaitement connus, la commutation entre de modèles multiples possède l'avantage d'être très rapide et stable. Dans le cas d'un nombre important de défauts, cette approche ne semble pas raisonnable car il faut également un nombre important de régulateurs à synthétiser.

6) Commande adaptative

La commande adaptative est une approche convenable pour résoudre le problème d'accommodation aux défauts de type interne, grâce à leur capacité d'adaptation automatique aux changements des paramètres du système. Il est remarquable que ce type de méthode ne contienne pas de module de détection des défauts. Par conséquent, un défaut interne agit sur le système provoque des variations des paramètres. L'identification en ligne de ces derniers permet alors la modification des paramètres du régulateur à structure fixe. Dans le domaine de l'aéronautique, ces méthodes ont fait l'objet de beaucoup d'études. Généralement, la présence de bruits n'est pas prise en compte dans les différentes situations étudiées. Aussi elles ne font souvent intervenir que des défauts peu sévères.

7) Commande prédictive

La philosophie de la commande prédictive se résume sur la résolution répétée à chaque pas de temps d'un problème de commande optimale. En d'autre terme, le principe de cette technique est d'utiliser un modèle dynamique du processus à l'intérieur du contrôleur en temps réel afin d'anticiper le futur comportement du système. Notant que la commande prédictive permet de gérer facilement les différentes contraintes de commande avec un nombre limité de paramètres de synthèse. L'avantage de cette méthode dans le domaine de commande tolérante aux défauts est de pouvoir modifier en ligne les différentes contraintes de commande de façon à garantir un niveau de performance acceptable [33]. Tout le potentiel de la commande prédictive à résoudre le problème de l'accommodation aux défauts a été montré dans [38 ,39,34].

8) Linéarisation par retour ou principe d'identification en ligne

Les régulateurs linéaires fonctionnent généralement correctement pour des petites variations de l'état ou de variables. Le concept de linéarisation par bouclage (feedback linearization) peut être utilisé pour compenser les effets liés aux non linéarités. Ces techniques ont été utilisées en aéronautique. Les défauts sont identifiés en estimant les paramètres des équations du mouvement de l'avion en utilisant les moindres carrés récursifs. Les paramètres estimés sont alors utilisés pour mettre à jour les nouveaux paramètres du régulateur. Étant donné qu'un avion a de nombreux paramètres à surveiller, des difficultés peuvent survenir quant à l'identification des paramètres.

9) Commande multi-modèle

La représentation multi-modèles est une approche possible pour développer des stratégies de commande tolérante, dans le cas où un modèle linéaire ne peut pas couvrir l'intégralité du domaine de fonctionnement du système (système non linéaire ou variant dans le temps). Ces méthodes sont basées sur un ensemble fini de modèles linéaires reliés par des fonctions d'activation. Il est alors possible dans ce cas d'appliquer les techniques linéaires pour les systèmes non linéaires.

La loi de commande globale est déterminée à partir de r lois de commandes calculées pour toutes les situations possibles du système, décrites par un ensemble de r modèles. Les autres situations prennent en compte l'apparition d'un défaut particulier, entraînant le système en dehors de sa zone de fonctionnement nominale. Les matrices de gain K_i de la commande sont calculées à l'avance pour chacun des modes de fonctionnement.

On peut déterminer la commande locale u_i par la relation suivante :

$$u_i = -K_i \cdot \hat{X}_i \quad (\text{II.12})$$

Où \hat{X}_i est l'estimation de l'état du système fournie par le $i^{\text{ème}}$ filtre. Une unité de calcul des probabilités de Bayes permet de calculer les probabilités $P(H_i|r_i)$ associées à chaque modèle possible par la relation suivante :

$$P(H_i|r_i) = \frac{P(H_i) P(r_i|H_i)}{\sum_{i=1}^n [P(H_i) P(r_i|H_i)]} \quad (\text{II.13})$$

$P(r_i|H_i)$: désigne la probabilité conditionnelle de l'innovation r_i issue du $i^{\text{ème}}$ filtre, et $P(H_i)$ la probabilité a priori du modèle H_i . La relation suivante montre la loi de commande globale appliquée au système :

$$U = \sum_{i=1}^r u_i P(r_i|H_i) \quad (\text{II.14})$$

Cette méthode exige le calcul a priori des gains des régulateurs correspondant à chaque situation du système. Une méthode fondée sur le principe d'interaction a été développée pour des défauts de type capteur et actionneur [79], Cette technique est basée sur une estimation d'état reconfigurée permettant d'éviter la modification du gain de la commande par retour d'état pour n'importe quelle situation du système.

10) Système expert

Un système expert se définit comme une application informatique capable de prendre des décisions ou de résoudre des problèmes dans un domaine précis. C'est un programme faisant appel à l'intelligence artificielle. La capacité des systèmes experts à établir un raisonnement proche de celui d'un humain en fait un support solide pour résoudre le problème de la reconfiguration.

Le système expert est composé d'une base de connaissance qui contient toute les informations relatives à l'état du système sous forme de paramètres et de règles, et d'un élément d'inférence qui cherche les valeurs des paramètres pour le test des différentes règles. Cet élément dernier gère les différentes composantes de la base de connaissances qui sont : la commande, la détection de défauts, le diagnostic, l'estimation de l'amplitude des défauts et la reconfiguration. Une fois le modèle du système en défaut connu, la phase de reconfiguration sélectionne l'action à prendre pour rétablir l'équilibre.

11) Approches basées sur la modélisation LPV

L'approche basée sur la modélisation LPV (Linéaire à Paramètres Variants) en utilisant des méthodes d'optimisation LMI est l'une des approches de commande tolérante aux défauts actives. Un régulateur LPV, $K(\hat{\theta}_f)$ est synthétisé pour garantir la stabilité et les performances d'un système représenté par un modèle LPV noté $P(\hat{\theta}_f)$ où $\theta_f \in \mathcal{R}^{q\theta}$ représente un vecteur de paramètres susceptible de varier avec le temps $\forall \theta_f \in \Theta, \Theta$ représente le domaine de variation de

θ_f et $\hat{\theta}_f$ est un vecteur de paramètre estimé. Par exemple, une loi de commande active tolérante aux défauts sur un pendule inversé à été appliquée dans [72].

12) Neuro-Floue

Egalement, les méthodes fondées sur les réseaux de neurones et la logique floue ont attiré l'attention des chercheurs intéressés par la commande tolérante aux défauts. Ces méthodes ont l'avantage d'être applicables sur les systèmes non linéaires qui sont généralement modélisées par des modèles flous de type Takagi-Sugeno représentés par des règles de la façon suivante :

$$\text{If } z(t) \in M_j, \text{ then } \begin{cases} \dot{x}(t) = A_j x(t) + B_j u(t) \\ y(t) = C_j x(t) + D_j u(t) \end{cases}$$

Les capacités d'apprentissage de ces méthodes permettent d'adapter le modèle et le contrôleur après l'apparition d'un défaut dans le système.

II.7. CONCLUSION

Dans ce chapitre, nous avons donné un rappel des principales commandes tolérantes aux défauts. Les deux méthodes excitantes de commande tolérante aux défauts ont été rappelées : la commande tolérante aux défauts passive et active. Les méthodes passives sont basées sur les techniques de conception d'une loi de commande robuste. En revanche, les méthodes actives permettent de traiter des défauts imprévus mais nécessitent une commande reconfigurable et un module de diagnostic.

Chapitre III

Modélisation par Réseaux de Neurones Artificiels et la logique floue

- III.1. Introduction*
 - III.2. Modélisation par réseaux de neurones*
 - III.3. Modélisation par logique floue*
 - III.4. Modélisation neuro-floue*
 - III.5. Conclusion*
-

III.1. INTRODUCTION

La plupart des processus dans l'industrie sont caractérisés par des comportements non linéaires et variables dans le temps. La modélisation des systèmes non linéaires est devenue un outil très important qui peut être utilisé pour améliorer les performances de commande et d'atteindre un comportement robuste et tolérant aux défauts.

Parmi les différentes techniques de modélisation non linéaire, les méthodes basées sur les techniques d'intelligence artificielle sont progressivement mis en place non seulement dans le milieu académique, mais aussi dans des applications industrielles.

L'utilisation des réseaux de neurones et des systèmes d'inférence floue pour la modélisation des systèmes de commande dans le domaine de diagnostic des défauts a connu un essor important au cours de ces dernières années. Les réseaux de neurones et les systèmes d'inférence floue sont capables de générer des modèles des comportements à partir des données entrées/sorties des systèmes dynamiques. L'un des avantages de ces techniques réside essentiellement, dans leur capacité à modéliser des comportements non linéaires et permettant aussi de modéliser n'importe quel système complexe sans aucune connaissance à priori du fonctionnement interne de ce dernier.

Dans ce chapitre, nous allons représenter un aperçu des méthodes de modélisation neuronale, floue et neuro-floue pour la modélisation des systèmes non linéaires, et nous citons aussi les méthodes de construction de ces modèles.

III.2. MODELISATION PAR RESEAUX DE NEURONES

Les réseaux de neurones artificiels sont des outils très intéressants car ils représentent des outils capables d'effectuer des opérations de modélisation, de prédiction et de classification...etc. Comme leur nom indique, le neurone artificiel est une modélisation mathématique simplifiée d'un neurone biologique, c'est-à-dire, le fonctionnement d'un neurone artificiel est basé sur le principe de fonctionnement d'un neurone biologique.

L'avantage principal des réseaux de neurones par rapport aux autres outils est leur capacité d'apprendre par des ensembles des données entrées/sorties, ce qui représente une des caractéristiques importantes. Leur utilisation est principalement guidée par leurs propriétés suivantes : la capacité d'apprentissage, le parallélisme dans le traitement, adapté aux non-linéarités des systèmes et la rapidité de traitement.

La qualité du modèle neuronal dépend des différentes caractéristiques de ces réseaux : le choix de la méthode d'apprentissage (supervisé ou non supervisé), et le choix de la structure (la topologie, le nombre de neurones, l'algorithme d'apprentissage...).

Dans ce qui suit, nous présentons les différentes stratégies de modélisation et les différents types de modèle fréquemment utilisés dans la tâche de la modélisation.

III.2.1. Architectures neuronales utilisées en modélisation

Nous utilisons les réseaux de neurones artificiels comme outils de modélisation du diagnostic car ils intègrent des mécanismes d'adaptation, d'apprentissage et de raisonnement nécessaires à la problématique du diagnostic.

1) Le Perceptron Multi-Couche 'PMC' ou 'MLP':

Le premier réseau de neurones que nous allons étudier s'appelle le perceptron multicouche « *Multi Layer Perceptron MLP* ». Ce réseau comporte une couche d'entrée, une ou plusieurs couches cachées à fonction d'activation de type "sigmoïde" et une couche de sortie. Le traitement de l'information se fait de la couche d'entrée vers la couche de sortie, c'est-à-dire à propagation avant.

L'architecture du réseau multicouche est très simple et elle peut être utilisée dans plusieurs types d'applications telles que pour les tâches de modélisation et de classification. La figure (III.1) représente un schéma simplifié d'un réseau MLP comportant une couche d'entrée, une couche cachée et une couche de sortie.

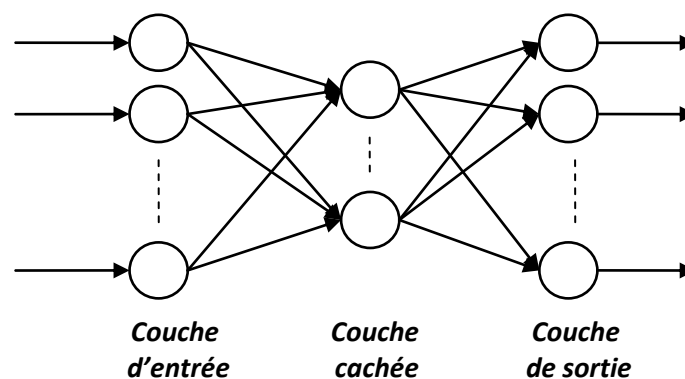


Figure III.1 : Schéma simplifié d'un perceptron Multicouche.

2) Réseau à Fonctions de Base Radiales 'RBF':

Le Réseaux à Fonction de Base Radiale (*Radial Basis Function*-RBF) est basé sur une architecture similaire à celle des réseaux MLP, mais ne comporte généralement que deux couches : une couche cachée et une couche de sortie.

La couche cachée, constituée des (noyaux) neurones RBF effectue une transformation non linéaire de l'espace d'entrée et la couche de sortie calcule une combinaison linéaire des sorties de la couche cachée. Chaque noyau élémentaire calcule la distance entre l'entrée et son centre qu'il passe ensuite dans une non linéarité concrétisée par une fonction d'activation $\phi(\)$ qui est généralement de type gaussienne. La valeur que prend la sortie du noyau gaussien est d'autant plus importante que l'entrée est plus proche de son centre et tend vers zéro, lorsque la distance entrée-centre devient importante. La sortie du réseau RBF est donnée par :

$$y_i = \sum_{k=1}^{N_1} w_{kj} \phi_k(\|x - c_k\|) \quad (III.1)$$

$$\phi(\xi) = \exp\left(-\frac{\xi^2}{2\eta^2}\right) \quad (III.2)$$

$\| \ \|$ dénote la norme euclidienne, x le vecteur d'entrée, c_k est le centre associé au noyau k . N_1 le nombre de noyaux de la couche cachée et w_{kj} les poids associés à la couche de sortie. Le paramètre η permet de contrôler la vitesse de décroissance de la fonction ϕ .

D'une manière pareille comme le réseau MLP, le réseau RBF possède aussi la propriété d'approximation, c'est à dire que le réseau est capable d'approximer n'importe quelle fonction continue avec une précision donnée pourvu qu'on fournisse un nombre de neurones suffisant.

L'apprentissage des réseaux RBF est composé d'une règle supervisée pour l'adaptation des poids et une règle d'apprentissage non supervisée pour l'adaptation des centres. Le choix de la position des centres et le nombre de neurones reste généralement arbitraire.

Les réseaux de neurones de type MLP et RBF sont capables d'approximer n'importe quelle fonction non linéaire. La différence porte sur l'architecture qui est figée en deux couches pour les RBF et peut comporter plusieurs couches pour le MLP. Le réseau RBF souffre d'une difficulté dans le calcul de la distribution optimale des centres. Si les centres ne sont pas identifiés avec une grande précision, il se produira des dégradations dans les performances du système. En outre, la complexité de calcul augmente exponentiellement avec le nombre d'entrée, augmentant ainsi la difficulté du problème. Cependant, l'addition d'une couche cachée

supplémentaire peut améliorer la généralisation du MLP; il peut aussi réduire le nombre de neurones requis et, donc, être une solution plus désirable [29].

Les avantages les plus importants que l'on peut attribuer à une application de diagnostic par réseaux de neurones sont la modélisation et l'estimation des fonctions non linéaires par apprentissage et la fusion de données et la généralisation.

Les réseaux MLP et RBF sont généralement les deux architectures neuronales les plus utilisées pour des tâches de diagnostic. Il existe évidemment de nombreuses autres variantes de réseaux de neurones utilisés dans le domaine de diagnostic tel que : les réseaux de neurones dynamiques, réseaux CPN (*Counter Propagation Networks*), réseaux GMDH (*Group Method of Data Handling*), réseaux CMAC (*Cerebellar Model Articulation Controller*) [51].

3) Réseaux récurrents :

Un réseau récurrent possède la même structure qu'un réseau multicouche munie de rétroactions. Les connexions rétroactives peuvent exister entre tous les neurones du réseau sans distinction, ou seulement entre certains neurones (par exemple, les neurones de la couche de sortie et les neurones de la couche d'entrée ou les neurones de la même couche). L'importance de ces réseaux est qu'ils permettent d'apprendre la dynamique de systèmes.

La figure ci-dessous montre un réseau multicouche récurrent, qui utilise un vecteur d'entrée qui contient les copies des activations de la couche de sortie du réseau.

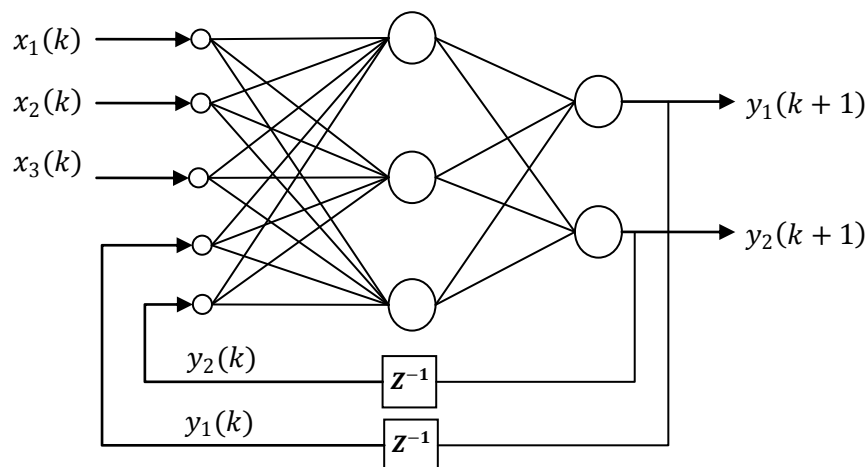


Figure III.2 : Réseau de neurones récurrent.

III.2.2. Stratégies de modélisation neuronale

Généralement, quand un modèle physique est connu, il est possible de concevoir un modèle de connaissances qui peut donner une représentation parfaite du système. Ainsi, les différents types de modèles sont distingués en fonction de la quantité de connaissances préalables physiques considérées. Alors, il est intéressant de concevoir un modèle de connaissance, mais en fonction de la complexité du système il est très difficile de l'obtenir. Dans ce cas, il est nécessaire d'utiliser les connaissances empiriques en utilisant les mesures expérimentales.

On peut distinguer deux principales stratégies de modélisation qui utilisent des réseaux de neurones : la première appelée l'approche par « boîte noire », quand le processus est représenté avec un réseau neuronal approprié. La deuxième appelée l'approche par « boîte grise » (appelé aussi modèle hybride). Ce modèle se situe entre la boîte noire et le modèle de connaissance qui contient à la fois des équations résultant d'une théorie, et des équations purement empiriques, résultant d'une modélisation de type « boîte noire ».

1) Modèle "boîte noire" :

Les modèles "boîte noire" sont construits essentiellement sur la base de mesures effectuées sur les entrées et les sorties du processus à modéliser. La modélisation consiste alors à utiliser, pour représenter les relations entre les entrées et les sorties, des équations (algébriques, différentielles, ou récurrentes) paramétrées, et à estimer les paramètres, à partir des mesures disponibles, de manière à obtenir la meilleure précision possible avec le plus petit nombre possible de paramètres ajustables. Le modèle « boîte noire » constitue la forme la plus primitive de modèle mathématique, c'est-à-dire, ce modèle est réalisé uniquement à partir des données expérimentales ou d'observations.

2) Modèle "boîte grise" ou "hybride" :

Lorsque des connaissances exprimables sous forme d'équations, sont disponibles, mais insuffisantes pour concevoir un modèle de connaissance satisfaisant, on peut avoir recours à une modélisation "boîte grise" (ou modélisation semi-physique) qui prend en considération à la fois les connaissances et les mesures. Une telle démarche peut concilier les avantages de l'intelligibilité d'un modèle de connaissance avec la souplesse d'un modèle comportant des paramètres ajustables [5].

III.2.3. Conception d'un réseau de neurones

Le problème de modélisation par réseaux de neurones peut être vu comme un processus d'identification de système. La construction d'un modèle neuronal comporte quatre grandes étapes essentielles.

1) La création et le choix de la base de données :

Avant d'utiliser un modèle, il faut tout d'abord créer une base de données contenant toutes les valeurs à utiliser. D'une façon générale, la création de deux bases de données est nécessaire pour effectuer la procédure d'apprentissage par l'une de ces données et l'autre pour effectuer le teste de réseau obtenu et déterminer ses performances. Il est souvent préférable de disposer une troisième base de données pour mieux vérifier la phase d'apprentissage et pour accomplir la validation de ce réseau.

2) Choix de la structure :

Cette deuxième étape consiste à choisir la structure du modèle qui peut représenter la dynamique du système, l'architecture du modèle et ses entrées. L'un des problèmes de l'utilisation des réseaux de neurones consiste dans le choix de sa topologie. La détermination de l'architecture optimale est restée un problème ouvert parce qu'il n'existe pas de méthodes générales pour le choix du nombre de neurones sur chaque couche cachée ainsi que le nombre de ces dernières. Cependant, un réseau à une seule couche cachée est dans la majorité des cas suffisant. Le choix d'un réseau est très important et peut par exemple dépendre de la tâche à effectuer (modélisation, classification, commande de processus,...) et de la nature des données.

3) Apprentissage :

Lorsque la structure du réseau de neurones est fixée, il faut choisir un processus d'apprentissage, par lequel les poids vont être ajustés de manière à satisfaire un critère d'optimisation. L'apprentissage des réseaux de neurones artificiels est une phase qui permet de déterminer ou de modifier les paramètres du réseau, afin d'adopter un comportement désiré. Plusieurs algorithmes d'apprentissage ont été développés depuis la première règle d'apprentissage de Hebb en 1949. La méthode d'apprentissage la plus utilisée est la rétro-propagation, qui est généralement plus économe que les autres en terme de nombres d'opérations arithmétiques à effectuer pour évaluer le gradient. Généralement, il existe trois types d'apprentissage, supervisé, non supervisé ou par renforcement.

4) Validation :

Cette dernière étape permet de mettre en évidence si le modèle identifié est représentatif des comportements entrées/sorties du système. Après la phase d'apprentissage, le réseau de neurones est entraîné. Donc il est nécessaire de le tester sur une base de données différente de celles utilisées pour l'apprentissage. Si les performances ne sont pas satisfaisantes, il faudra soit modifier l'architecture du réseau, soit modifier la base d'apprentissage.

La figure (III.2) résume les étapes principales de la construction d'un modèle neuronal.

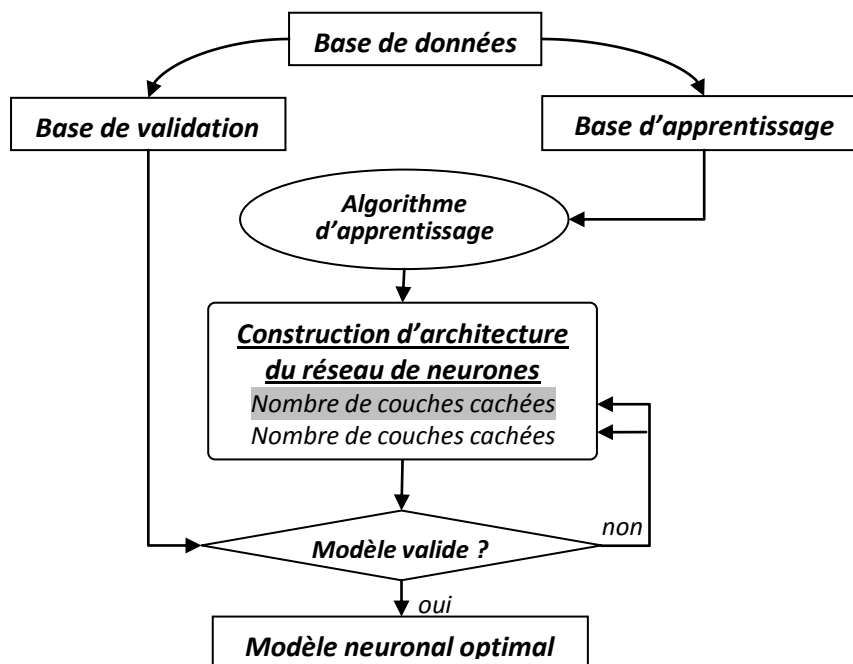


Figure III.3 : Schéma général de construction d'un modèle neuronal.

III.3. MODELISATION PAR LOGIQUE FLOUE

La logique floue se définit comme une description mathématique d'un processus basée sur la théorie des ensembles flous développée en 1965 par Lotfi Zadeh [30]. On peut dire aussi que la logique floue est un outil permet la représentation des connaissances sous une forme proche du langage naturel. C'est-à-dire, cet outil se propose de remplacer les variables booléennes de la logique classique par des variables floues.

La structure de base d'un système basée sur la logique floue est composée de quatre éléments principaux : Un premier bloc représente une base de connaissance, qui contient à la fois

un ensemble de règles floues, connu sous le nom de la base de règles, et un ensemble de fonctions d'appartenance connue sous le nom de la base de données. Le deuxième bloc est le bloc de fuzzification qui réalise la transformation de valeurs numériques en informations symboliques floues. Un troisième bloc représente un moteur d'inférence qui applique un mécanisme de raisonnement flou afin d'obtenir une sortie floue. Enfin, le dernier bloc est le bloc de défuzzification, qui représente l'opération inverse de la fuzzification. La défuzzification fait correspondre à chaque résultat flou calculé, la sortie précise adéquate [58].

La modélisation floue d'un système est intéressante par rapport aux autres méthodes, parce que, dans certains cas, les techniques de modélisation floue permettent d'interpréter les connaissances contenues dans un modèle, ainsi on veut extraire des connaissances à partir de données numériques, en les exprimant dans un langage proche du naturel. La modélisation floue peut être considérée comme une modélisation boîte grise.

Les modèles flous peuvent être considérés comme des modèles logiques qui utilisent une représentation de la connaissance sous forme des règles du type « Si ...Alors... ». Ce type de règles permet d'établir des relations qualitatives entre les variables d'entrée et de sortie du système.

D'une manière générale, les systèmes flous sont constitués d'une collection de fonctions d'appartenance et de règles floues. Les modèles flous basés sur des règles, permettent d'utiliser n'importe quelle information sous forme des expressions du langage naturel, ce qui facilite la formalisation de la connaissance et l'interprétabilité du modèle. De plus, les modèles flous ont une structure mathématique capable d'approximer les systèmes non linéaires complexes avec une bonne précision. Notant que les algorithmes d'apprentissage sont aussi utilisés pour la modélisation floue. Dans ce qui suit, nous représentons les différents types de modèles flous.

III.3.1. Différents types de modèles flous

Comme mentionné plus haut, les systèmes flous sont basés sur une représentation de connaissance sous forme des règles floues qui permettent de représenter les relations entre les variables d'entrée et de sortie dont l'expression générale est de la forme :

« Si *prémisse*, Alors, *conséquent* ».

On peut considérer la prémisse (antécédent) comme une description linguistique qui indique les conditions de validité du phénomène représenté. En revanche, le conséquent (conclusion)

représente le comportement associé aux conditions de validité décrite par la prémisse. Selon cette structure et les applications de la logique floue en commande des systèmes, on distingue généralement trois grandes familles de modèles flous à base de règles :

- les modèles flous linguistiques (ou de Mamdani),
- les modèles flous relationnels,
- les modèles flous de Takagi-Sugeno.

Généralement, les modèles flous sont basés sur une collection de règles « si...alors... ». Dans le cadre des applications de la logique floue en commande des systèmes, les types de modèles les plus souvent utilisés sont ceux de Mamdani et de Takagi-Sugeno. Les prémisses de règles dans ces deux modèles sont exprimées symboliquement, la différence est dans l'expression des conclusions de règles. Les modèles flous relationnels peuvent être considérés comme une généralisation du modèle linguistique. Dans ces modèles derniers, il est possible d'associer une proposition antécédente spécifique avec plusieurs propositions conséquentes différentes par une relation floue.

III.3.1.1. Modèle flou de type Mamdani

Nous avons vu que les systèmes d'inférence floue (SIF) sont composés par une collection de plusieurs règles. Le premier type de ces systèmes est le système d'inférence floue avec une structure Mamdani ou appelé aussi modèle linguistique. La forme générale des règles de cette structure Mamdani est représentée sous la forme :

$$R_i: \quad Si \ x \ est \ A_i \ Alors \ y \ est \ B_i \quad i = 1, \dots, r \quad (III.3)$$

où R_i dénote la i ème règle du modèle est r est le nombre de règles que contient la base de règle. $x \in \mathfrak{R}^P$ est la variable d'entrée (antécédent) et $y \in \mathfrak{R}$ est la variable de sortie (conséquent). Les termes linguistiques représentés par A_i et B_i sont définis par des ensembles flous caractérisés par des fonctions d'appartenance $\mu_{A_i}(x): \mathfrak{R}^P \rightarrow [0, 1]$ et $\mu_{B_i}(x): \mathfrak{R} \rightarrow [0, 1]$.

On peut voir dans ce cas que le système flou est représenté par une représentation qualitative qui décrit le comportement du système en utilisant une description linguistique.

Le mécanisme d'inférence dans le modèle linguistique est basé sur la règle compositionnelle d'inférence [76] Pour le cas des systèmes décrits par plusieurs variables, les

propositions antécédentes des règles sont des combinaisons logiques, à l'aide des opérateurs flous, de propositions floues élémentaires.

Pour un modèle à p variables d'entrées et une variable de sortie, les règles s'écrivent alors sous la forme :

$$R_i: \text{ Si } x_1 \text{ est } A_{i1} \text{ et } x_2 \text{ est } A_{i2} \text{ et ... et } x_p \text{ est } A_{ip} \text{ Alors } y \text{ est } B_i \quad (III.4)$$

L'opérateur « et » flou est la forme conjonctive la plus couramment utilisée en modélisation des processus réels multivariés. Le calcul d'un degré d'activation de la règle souvent noté par β_i . Par exemple, pour un système à trois variables d'entrée $\{x_1, x_2, x_3\}$ et une sortie y (système MISO), la règle floue peut s'écrire sous la forme suivante :

$$R_i: \text{ Si } x_1 \text{ est } A_{i1} \text{ et } x_2 \text{ est } A_{i2} \text{ ou } x_3 \text{ n'est pas } A_{i3} \text{ Alors } y \text{ est } B_i \quad (III.5)$$

Dans ce cas le degré d'activation de la règle est calculé en utilisant les opérateurs d'intersection, d'union et de complément. Alors, l'expression de degré d'activation est donné par :

$$\beta_i = \mu_{A1}(x_1) \wedge \mu_{A2}(x_2) \vee (1 - \mu_{A3}(x_3)) \quad (III.6)$$

où l'opérateur minimum (\wedge) représente dans ce cas la conjonction (et), l'opérateur maximum (\vee) représente la disjonction (ou), et $(\mu - 1)$ est le complément (négation : n'est pas).

Pour le cas des entrées numériques précises et en utilisant la t-norme, le schéma de raisonnement peut être simplifié, aboutissant à celui bien connu dans la littérature appelé le max-min ou inférence de Mamdani.

Le mécanisme d'inférence de Mamdani, pour le cas de la règle composée définie par l'expression (III.5), peut être résumé en trois étapes :

Etape 1 : Calculer pour chaque règle i le degré d'activation β_i des propositions antécédentes :

$$\beta_i = \mu_{A_{i1}}(x_1) \wedge \mu_{A_{i2}}(x_2) \wedge \dots \wedge \mu_{A_{ip}}(x_p)$$

Etape 2 : Calculer l'ensemble flou de sortie β'_i pour chaque règle, en utilisant la t-norme du minimum :

$$\mu_{B'_i}(y) = \beta_i \wedge \mu_{B_i}(y), \quad \forall y \in Y$$

Etape 3 : Calculer l'ensemble flou de sortie agrégée en prenant le maximum de chaque conclusion β'_i : $\mu_{\beta'}(y) = \max \mu_{B'_i}(y), \quad \forall y \in Y, i = 1, \dots, r$

III.3.1.2. Modèle flou relationnel

Le modèle flou relationnel peut être tout simplement considéré comme une généralisation du modèle linguistique. Ce modèle est constitué d'une règle floue R décrivant le lien existant entre les différents éléments du système. Dans d'autre terme, il est possible avec ce type des modèles d'associer une proposition antécédent spécifique avec plusieurs propositions conséquents différentes via une relation floue. Notons que la règle floue R est discrète, c'est-à-dire construire sur des univers discrétisés (numériques ou symboliques) et les interfaces de fuzzification et défuzzification ne sont pas intégrées pour ce type des modèles [45].

Pour mieux comprendre, nous considérons d'abord le modèle linguistique flou qui se compose des règles suivantes :

$$R_i: \text{Si } x_1 \text{ est } A_{i1} \text{ et ... et } x_n \text{ est } A_{in} \text{ Alors } y \text{ est } B_i, i = 1, 2, \dots, k \quad (III. 7)$$

$x \in X$ et $y \in Y$ l'entrée et la sortie d'un modèle statique. Soit A, B représentent des collections des ensembles flous M et N définis sur les domaines X et Y , respectivement, c'est-à-dire :

$$A = \{A_1, A_2, \dots, A_M\}, B = \{B_1, B_2, \dots, B_N\},$$

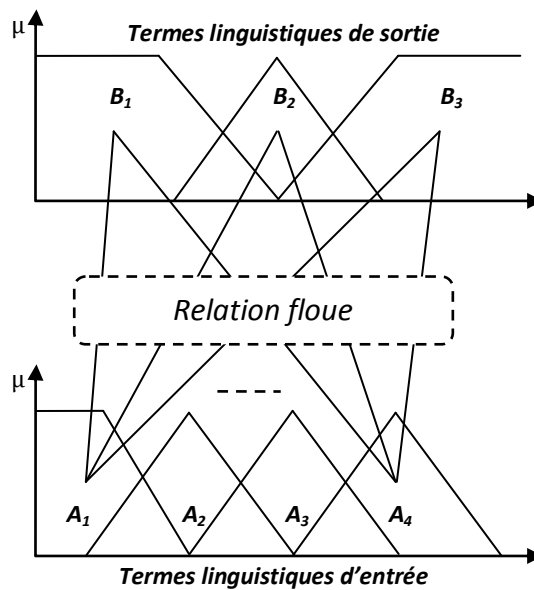


Figure III.4: Modèle flou relationnel.

D'après la figure (III.3), une association $R : A \rightarrow B$ à été définie par la relation floue $R = [r_{ij}]_{M \times N}$, où chaque A_i est relié à chaque B_j avec un poids défini par l'élément r_{ij} de la relation.

Nous pouvons ajuster par cette pondération, les conséquences des règles sans changer les ensembles flous de référence (termes linguistiques). Nous avons vu que pour les modèles linguistiques, les conséquences sont bien déterminées, en revanche chaque conséquence du modèle flou relationnel peut prendre une valeur aléatoire sur le domaine y . Cet avantage donne plus de paramètres libres, mais elle cause des problèmes au niveau de l'identification [15].

III.3.1.3. Modèle flou de type Takagi-Sugeno

Une deuxième modèle flou utilisé pour l'approximation d'une classe générale de systèmes non linéaires est celui proposé par Takagi et Sugeno. Comme celui de Mamdani, ce type de modèle est construit à partir d'une base de règles "Si...Alors...", les conséquences de règles sont des fonctions strictes des entrées du système flou. Ces conséquents peuvent s'exprimer par exemple, sous la forme d'une constante, d'un polynôme ou de manière plus générale d'une fonction ou d'une équation différentielle dépendant des variables associées à l'antécédent.

D'une manière générale, les règles d'un modèle de type Takagi-Sugeno (TS) sont représentées sous la forme :

$$R_i: \text{ Si } x \text{ est } A_i \text{ Alors } y_i = f_i(x), \quad i = 1, \dots, r \quad (III.8)$$

où R_i dénote la i ème règle du modèle est r est le nombre de règles que contient la base de règle. $x \in \mathfrak{R}^P$ est le variable d'entrée (antécédent) et $y \in \mathfrak{R}$ est la variable de sortie (conséquent). A_i est le sous-ensemble flou de l'antécédent de l' i ème règle.

Les fonctions f_i sont choisies comme des fonctions paramétrées appropriées, avec la même structure pour chaque règle où seuls les paramètres varient.

Dans ce cas, le modèle affine est obtenu par :

$$R_i: \text{ Si } x \text{ est } A_i \text{ Alors } y_i = a_i^T x + d_i, \quad i = 1, \dots, r \quad (III.9)$$

où $a_i \in \mathfrak{R}^P$ est un vecteur de paramètres et d_i est un scalaire. Ce modèle est appelé le modèle affine Takagi-Sugeno.

Un cas particulier de la fonction du conséquent s'obtient quand $d_i = 0$. Dans ce cas, le modèle est appelé modèle homogène Takagi-Sugeno de la forme :

$$R_i: \text{ Si } x \text{ est } A_i \text{ Alors } y_i = a_i^T x, \quad i = 1, \dots, r \quad (III.10)$$

Un autre cas particulier de la fonction du conséquent s'obtient quand $a_i = 0$. Les conséquents dans le modèle prennent la forme d'une constante. Dans ce cas, le modèle est obtenu par :

$$R_i: \text{ Si } x \text{ est } A_i \text{ Alors } y_i = d_i, \quad i = 1, \dots, r \quad (III.11)$$

L'obtention de la sortie du modèle, dans la modélisation Takagi-Sugeno, est réalisée à partir d'une combinaison des opérations d'inférence et de défuzzification. La sortie finale sera obtenue en utilisant la formule générale suivante :

$$y = \frac{\sum_{i=1}^K \mu_{A_i}(x) y_i}{\sum_{i=1}^K \mu_{A_i}(x)} \quad (III.12)$$

Les modèles affines de Takagi-Sugeno permettent d'approcher les systèmes non linéaires pas une combinaison de plusieurs modèles locaux linéaire. Le modèle affine de Takagi-Sugeno, avec une structure commune du conséquent, peut être exprimé comme un modèle pseudo-linéaire avec des paramètres dépendants des entrées :

$$y = \frac{\sum_{i=1}^K \mu_{A_i}(x) (a_i^T x + b_i)}{\sum_{i=1}^K \mu_{A_i}(x)} = a^T(x) x + b(x) \quad (III.13)$$

$$y = \frac{\sum_{i=1}^K \mu_{A_i}(x) (a_i^T x + b_i)}{\sum_{i=1}^K \mu_{A_i}(x)} = \left(\sum_{i=1}^K \gamma_i(x) a_i^T \right) x + \sum_{i=1}^K \gamma_i(x) b_i \quad (III.14)$$

où : $\gamma_i = \frac{\mu_{A_i}(x)}{\sum_{i=1}^K \mu_{A_i}(x)}$

K représente le nombre de règles dans la base de règles,

$\mu_{A_i}(x)$ est le degré d'activation de la i ème règle.

Le modèle TS peut être vu comme une correspondance entre l'espace de l'antécédent et une région convexe (polytope) dans l'espace des paramètres d'un système quasi-linéaire. Cette propriété a certains avantages qui facilitent l'analyse des systèmes flous TS dans le cadre des systèmes poly-topiques [14].

La représentation mathématique des modèles de Takagi-Sugeno constitue une représentation très intéressante des systèmes non linéaires, car il permet de représenter tout système non linéaire, quelle que soit sa complexité, par une structure simple en s'appuyant sur des modèles linéaires interpolés par des fonctions non linéaires.

III.3.2. Construction d'un modèle flou

Pour construire des modèles flous, deux sources d'information commune sont la connaissance préalable et les données (mesures de processus). La connaissance préalable peut être de nature approximative (connaissance qualitative, heuristiques), qui provient généralement des experts, à savoir, les concepteurs de processus, opérateurs,...etc. En ce sens, les modèles flous peuvent être considérés comme des systèmes experts flous simples. La construction des modèles flous à partir de données implique les méthodes basées sur la logique floue et le raisonnement approximatif. Deux approches principales peuvent être distinguées pour l'intégration des connaissances et des données dans un modèle flou :

- Aucune connaissance préalable sur le système n'est utilisée pour formuler les règles, et un modèle flou est construit à partir des données. Il est prévu que les règles extraites et les fonctions d'appartenance peuvent fournir une interprétation a posteriori du comportement du système.
- La connaissance d'experts est traduite dans une collection de règles "si-alors". De cette façon, une structure de certains modèles est créée. Les paramètres de cette structure peuvent être affinés en utilisant des données entrées/sorties.

Ces techniques, bien sûr, peuvent être combinées, en fonction de l'application particulière. Dans la conception des modèles, deux éléments de base sont distingués : la structure et les paramètres du modèle [8]. Dans la suite, nous résumons les étapes utilisées afin d'obtenir un modèle flou.

a. La détermination de la structure de base de règle :

En effet, la forme des règles dépend du type de modèle et des variables d'entrée/sortie supposées déterminées.

b. La détermination du nombre et type des fonctions d'appartenance pour chaque variable :

Ce choix détermine la granularité du modèle. Aussi dans le but de modélisation et le détail des connaissances disponibles, va influencer ce choix.

c. La détermination de type du mécanisme d'inférence, les opérateurs conjonctifs et la méthode de défuzzification :

Ces choix sont limités par le type de modèle flou.

d. La validation du modèle :

Après la structure est fixe, la performance d'un modèle flou peut être affinée en ajustant ses paramètres. Les paramètres réglables de modèles linguistiques sont les paramètres de l'antécédent, les fonctions d'appartenance de conséquent et les règles. Les modèles Takagi-Sugeno ont des paramètres dans les fonctions d'appartenance de l'antécédent et dans les fonctions de conséquent [45,8].

III.4. MODELISATION NEURO-FLOUE

Les techniques modernes de l'intelligence artificielle ont trouvé une application dans presque tous les domaines de la connaissance humaine. Les deux techniques de réseaux de neurones et la logique floue sont souvent appliquées en ensemble pour résoudre les problèmes, où les techniques classiques ne fournissent pas une solution facile et précise. Le terme neuro-floue est né de la combinaison de ces deux techniques. Les systèmes neuro-flous sont des systèmes flous formés par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones. La technique d'apprentissage opère en fonction de l'information locale et produit uniquement des changements locaux dans le système flou d'origine [57].

Généralement, les systèmes neuro-flous ont été appliqué dans de nombreux domaines en raison de leurs avantages. On peut citer parmi ces caractéristiques : la conception à base de règles, l'application aux systèmes complexes et incertains non-linéaires, les capacités d'apprentissage et le calcul parallèle et la tolérance aux défauts.

L'obtention et la détermination des fonctions d'appartenance et des règles appropriées pour un système flou est souvent une étape difficile et l'exécution d'un système flou peut prendre beaucoup de temps. Donc dans ce cas là, il faut trouver des idées qui permettent d'appliquer des algorithmes d'apprentissage pour les systèmes flous. Les réseaux de neurones, qui ont des algorithmes d'apprentissage très efficaces, avait été présenté comme une alternative à automatiser ou de soutenir le développement des systèmes flous.

En général, toutes les combinaisons de techniques basées sur les réseaux de neurones et la logique floue peuvent être appelées systèmes neuro-flous. Les principaux avantages des

techniques floues sont l'approche naturelle de la modélisation et la bonne interprétabilité de la description, en employant des règles linguistiques. Aussi, les réseaux de neurones autorisent de disposer des algorithmes permettant l'apprentissage automatique de ces paramètres. On peut aussi noter que les systèmes neuro-flous peuvent être utilisés comme des approximateurs universels.

- **Définition**

Une définition des systèmes neuro-flous est donnée dans [47]. Les systèmes neuro-flous sont des systèmes flous formés par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones. La technique d'apprentissage opère en fonction de l'information locale et produit uniquement des changements locaux dans le système flou d'origine.

III.4.1. Types des systèmes neuro-flous

Généralement, depuis les années quatre vingt, il ya plusieurs types pour combiner les réseaux de neurones et les systèmes flous. Ces types peuvent être classées en fonctionnelle et structurelle, suivant leur architecture et la configuration recherche entre le système d'inférence floue et les réseaux de neurones.

III.4.1.1. Système neuro-flou coopératif

Dans le moyen le plus simple, un système coopératif peut être considéré comme un préprocesseur dans lequel le mécanisme d'apprentissage de réseaux de neurones artificiels détermine les fonctions d'appartenance de système d'inférence floue ou les règles floues à partir des données d'apprentissage. Une fois les paramètres du système d'inférence floue sont déterminés, les réseaux de neurones artificiels vont au fond. La règle de base est couramment déterminée par une approche de clustering (cartes auto-organisation) ou un algorithme clustering flou. Les fonctions d'appartenance sont généralement approchées par un réseau de neurones à partir des données d'apprentissage.

III.4.1.2. Système neuro-flou concurrent

Dans un système Neuro-flou concurrent, le réseau de neurones aide le système d'inférence floue en permanence afin de déterminer les paramètres nécessaires particulièrement si les variables d'entrée du contrôleur ne peuvent pas être mesurées directement. Dans certains cas, les

sorties de système d'inférence floue ne pourraient pas être directement applicable au processus. Dans ce cas le réseau de neurones peut jouer le rôle d'un post-processeur de système d'inférence floue de sorties. La figure suivante représente les systèmes neuro-flous coopératif et Concurrent [3,2].

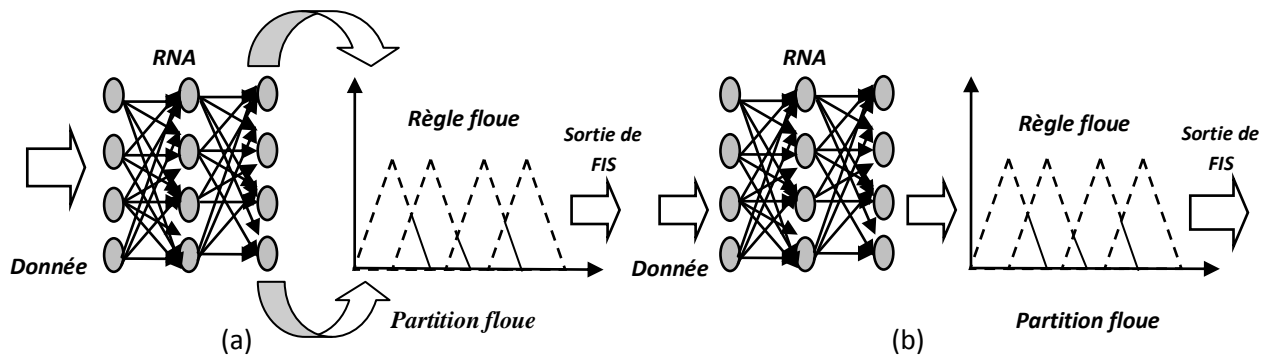


Figure III.5 : Systèmes neuro-flous, a) Coopératif, b) concurrent.

III.4.1.3. Système neuro-flou hybride

Dans une architecture Neuro-Floue hybride, les algorithmes d'apprentissage des réseaux de neurones sont utilisés pour déterminer les paramètres de système d'inférence floue. Les systèmes Neuro-Flous hybrides partagent les structures de données et la représentation des connaissances. Une manière commune d'appliquer un algorithme d'apprentissage à un système flou est pour représenter-la dans une architecture spéciale. Toutefois, les algorithmes d'apprentissage classiques de réseaux de neurones (descente de gradient) ne peuvent pas être appliqués directement à un système d'inférence comme une fonction parce que les fonctions utilisées dans le système d'inférence sont habituellement non différentiables. Ce problème peut être abordé en utilisant des fonctions différentiables dans le système d'inférence ou en n'utilisant pas l'algorithme d'apprentissage des réseaux de neurones standard [2].

III.4.2. Architectures neuro-floues

1. Systèmes Neuro-Flous intégrés type Mamdani :

Un système de Mamdani neuro-flou utilise une technique d'apprentissage supervisé (apprentissage par rétropropagation) pour apprendre les paramètres des fonctions d'appartenance. L'architecture du système neuro-flou de Mamdani contient cinq couches.

- Une couche d'entrée :

Le rôle de cette première couche est de transmettre les valeurs d'entrée à la couche suivante directement. Chaque nœud dans cette couche correspond à une seule variable d'entrée.

- Une couche de fuzzification :

Chaque nœud dans cette deuxième couche correspond à un seul label linguistique à l'une des variables d'entrée dans la couche d'entrée. En d'autres termes, le lien de sortie représente la valeur d'appartenance, qui précise le degré auquel une valeur d'entrée appartient à être un ensemble flou, est calculée dans la deuxième couche. Un algorithme de classification décidera du nombre initial et le type de fonctions d'appartenance à attribuer pour chacune des variables d'entrée. Les formes finales des fonctions d'appartenance seront réglées avec précision durant l'apprentissage du réseau.

- Une couche de règles antécédentes :

C'est la troisième couche. Un nœud dans cette couche représente la partie antécédente d'une règle. Habituellement, un opérateur T-norme est utilisé dans ce nœud. La sortie d'un nœud de la troisième couche représente le sens de la règle floue correspondante.

- Une couche de règles conséquentes :

Ce nœud a deux tâches : Combiner les antécédents de règles et déterminer le degré avec lequel ils appartiennent au label linguistique de sortie. Le nombre de nœuds dans cette couche est égal au nombre de règles.

- Une couche de défuzzification :

Dans cette dernière couche le nœud effectue la combinaison de toutes les parties conséquentes des règles utilisant un opérateur S-norme et calcule enfin la sortie non floue après défuzzification [71].

2. Systèmes Neuro-Flous intégrés type Takagi-Sugeno :

Les systèmes neuro-flous de type Takagi-Sugeno utilisent une combinaison mélange de l'algorithme de rétro-propagation pour apprendre les fonctions d'appartenance et la méthode d'estimation de moindres carrés pour déterminer les coefficients des combinaisons linéaires dans les conclusions de règles. Une étape dans la procédure d'apprentissage se compose de deux parties : Dans la première partie les modèles d'entrée sont propagées, et les paramètres d'une conclusion optimale sont estimés par une procédure des moindres carrées, tandis que les paramètres antécédent (fonctions d'appartenance) sont supposées fixes pour le cycle courant à travers l'ensemble d'apprentissage. Dans la seconde partie, les entrées sont propagées encore une fois et, pendant cette itération, la rétro-propagation est utilisée pour modifier les paramètres antécédents, pendant que les paramètres conclusion restent fixes. Cette procédure est ensuite répétée.

- Les couches 1, 2, 3 :

Leurs rôles sont similaires aux couches 1,2 et 3 du système de type Mamdani.

- Couche 4 (normalisation sens de la règle) :

Chaque nœud dans cette couche calcule le rapport du sens de la $i^{\text{ème}}$ règle par rapport à la somme des sens de toutes les règles :

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2, \dots \quad (III.15)$$

- Couche 5 (couche conséquent de règle) :

Chaque nœud i dans cette couche est doté d'une fonction nœud :

$$\bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i) \quad (III.16)$$

où \bar{w}_i est la sortie de la couche 4, et $\{p_i, q_i, r_i\}$ est l'ensemble des paramètres. Les paramètres conséquents sont déterminés par l'algorithme des moindres carrés.

- La couche 6 (couche inférence de règle) :

L'unique nœud de cette couche calcule la sortie globale par addition de tous les signaux entrants :

$$\text{Sortie globale} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (III.17)$$

Dans les sections suivantes, nous discutons brièvement les différents systèmes neuro-flous intégrés qui réalisent la complémentarité des réseaux de neurones et des systèmes d'inférence floue de type Mamdani ou des systèmes d'inférence floue de type Takagi Sugeno. Les types de systèmes Neuro-Flous principaux sont : GARIC, FALCON, ANFIS, NEFCON, SONFIN, et beaucoup d'autres [71].

3. Système ANFIS :

L'architecture ANFIS (*Adaptive-Network-based Fuzzy Inference System*) est peut-être le premier système neuro-flou intégré. Ce système met en œuvre un système d'inférence floue de type Takagi Sugeno et il comporte cinq couches.

La première couche cachée est responsable de la cartographie de variable d'entrée relativement à chacun des fonctions d'appartenance, c'est-à-dire, cette couche d'entrée est une couche permet la "fuzzification" des variables. L'opérateur T-norme est appliqué dans la deuxième couche cachée pour calculer les antécédents des règles. La troisième couche cachée normalise les résultats fournis par la couche précédente, suivies par la quatrième couche cachée où les conséquents des règles sont déterminées. La couche de sortie calcule la sortie globale comme étant la somme de tous les signaux qui arrivent à cette couche.

L'ANFIS utilise l'apprentissage par rétro-propagation afin de déterminer les paramètres des fonctions d'appartenance d'entrée et la méthode des moindres carrés moyens pour déterminer les paramètres conséquents. Chaque étape de l'algorithme d'apprentissage itératif comporte deux parties. Dans la première partie, la rétro-propagation est utilisée pour la mise à jour des paramètres antécédents et les moindres carrés moyens pour mettre à jour les paramètres conséquents, pendant que les paramètres des antécédents sont considérés fixes. Dans la deuxième partie, les modèles d'entrée sont propagés à nouveau et à chaque itération, l'algorithme d'apprentissage par rétro-propagation est utilisé pour modifier les paramètres des antécédents, pendant que les conséquents restent fixes.

4. L'architecture FALCON :

FALCON (*Fuzzy Adaptive learning Control Network*) a une architecture de cinq couches. Il ya deux nœuds linguistiques (neurones) pour chaque variable de sortie. Une pour les données d'apprentissage (sortie désirée) et l'autre pour la sortie de FALCON. La première couche cachée

est responsable de la fuzzification de chaque variable d'entrée. Chaque neurone dans cette couche représentant une fonction d'appartenance à un ensemble flou. La deuxième couche cachée définit les parties antécédentes des règles floues suivie par les parties conséquences des règles dans la troisième couche cachée. Ce système utilise un algorithme hybride d'apprentissage comportant l'apprentissage non supervisé pour localiser les fonctions d'appartenance initiale et la base de règles initiale et l'apprentissage supervisé pour optimiser l'ajustement des paramètres du fonction d'appartenance pour générer les sorties désirées.

5. L'architecture NEFCON :

NEFCON (*Neural Fuzzy CONTroller*) est conçu pour mettre en application le système d'inférence floue type Mamdani. Il est constitué de 2 couches dont les poids sont les ensembles flous et les règles floues. Avec la même utilisation antécédente supposée ont partagé les poids, qui sont représentés par des ellipses dessinées autour des raccordements. Elles assurent l'intégrité de la base de règle. La couche d'entrée assure la tâche de l'interface de fuzzification, la logique d'inférence est représentée par les fonctions de propagation, et la couche de sortie est l'interface de défuzzification. L'apprentissage du modèle de NEFCON est basé sur un mélange de l'apprentissage non supervisé et supervisée (rétro-propagation). NEFCON peut être employé pour apprendre des règles initiales, si aucune connaissance du système n'est disponible ou même pour optimiser une base manuellement définie de règle [2].

6. L'architecture NEFCLASS :

Le système NEFCLASS (*Neuro-Fuzzy Classification*) est utilisé généralement en classification, il est constitué de 3 couches : une couche d'entrée avec les fonctions d'appartenance, une couche cachée représentée par des règles et une couche de sortie définissant les classes. Ce modèle est facile à mettre en application, parce qu'il évite l'étape de défuzzification, tout en étant précis dans le résultat final, avec une rapidité bien supérieure aux autres architectures.

7. L'architecture NEFPROX :

Ce modèle (NEFPROX : *Neuro Fuzzy function apPROXimator*) est obtenu par l'association des deux architectures : NEFCLASS et NEFCON, il est utilisé dans différentes applications comme la classification et l'approximation de fonctions. Le NEFCLASS utilise un algorithme supervisé pour définir les règles floues. En revanche, le NEFCON utilise un algorithme d'apprentissage non supervisé avec le calcul de l'erreur de sortie. Les deux modèles emploient la rétro-propagation afin de définir les sous ensembles flous [57].

Parmi tous les systèmes neuro-flous, le modèle ANFIS présente la plus faible erreur moyenne quadratique. Ceci est probablement dû à l'implémentation de règles type Takagi-Sugeno dans le modèle ANFIS comparé au système flou type Mamdani. Comparé au modèle ANFIS, NEFPROX est beaucoup plus rapide, mais ANFIS donne des meilleurs résultats en approximation [57].

III.4.3. METHODOLOGIE DE MODELISATION NEURO-FLOUE

Comme il est montré dans [36], la modélisation neuro-floue est généralement composée de trois phases essentielles :

- Création du modèle flou initial, qui consiste en la collection des règles floues,
- Sélection des variables d'entrées significatives, en se basant sur le modèle flou initial,
- Détermination de nombre optimal des règles floues.

La figure suivante montre le schéma général de la modélisation neuro-floue [57] :

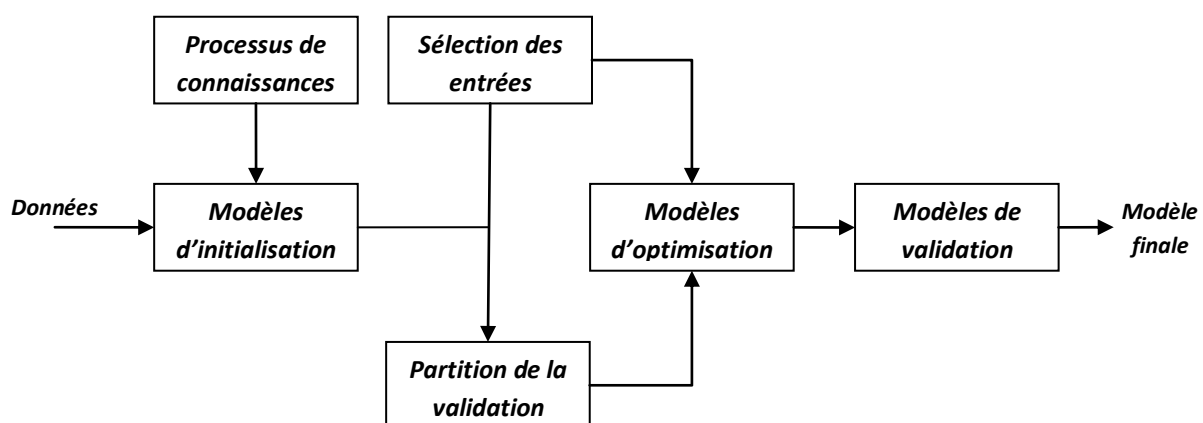


Figure III.6 : Schéma général de la modélisation neuro-floue.

III.5. CONCLUSION

Dans ce chapitre, nous avons étudié les différentes stratégies de modélisation des systèmes non linéaires et nous avons montré aussi qu'il est possible de les représenter avec les réseaux de neurones, la logique floue et les réseaux neuro-floues.

Nous avons vu que les systèmes d'inférence floue peuvent raisonner avec l'information imprécise, et expliquer leurs décisions mais ne peuvent pas automatiquement acquérir les règles qu'ils l'utilisent pour prendre ces décisions. Par ailleurs, les réseaux de neurones sont bons à reconnaître des modèles mais ne sont pas intéressants pour expliquer comment ils atteignent leurs décisions. Ces limites ont été une raison derrière la création des systèmes hybrides intelligents dans une manière à vaincre les limitations d'une seule technique.

Le chapitre suivant présente quelques perspectives de ces outils à travers la proposition des outils de diagnostic et de commande basés sur l'utilisation des structures neuronales, floues et/ou neuro-floues.

Chapitre IV

Diagnostic des défauts neuro-flous

- IV.1. Introduction*
 - IV.2. Modèles dynamiques non linéaires*
 - IV.3. Diagnostic des défauts neuro/flous*
 - IV.4. Exemple de diagnostic*
 - IV.5. Conclusion*
-

IV.1. INTRODUCTION

Dernièrement, les techniques d'intelligence artificielle ont connu une utilisation large en ce qui concerne la modélisation, le diagnostic et la commande des systèmes industriels, puisque elles permettent de pallier la complexité des systèmes.

D'une manière générale, ces techniques, tels que les réseaux de neurones et les systèmes d'inférence floue, sont relativement bien adaptées aux problèmes du diagnostic et de commande, car ils peuvent se caractériser par la capacité de traiter une grande quantité d'informations, des données non homogènes (symboliques) et des données incomplètes. Comme nous l'avons rappelé précédemment, les réseaux de neurones et les systèmes d'inférence floue, permettent aussi de bien modéliser la dynamique des systèmes non linéaires, ce qui rend ces techniques capables d'assister la supervision totale de ces systèmes à suivi.

Par la suite, nous présentons, dans un premier temps, un bref rappel sur les différents modèles dynamiques non linéaires, et dans un deuxième temps, nous détaillons les différentes étapes de diagnostic et de commande tolérante aux défauts où ces étapes sont réalisées avec les techniques de réseaux de neurones, les systèmes d'inférence floue et les systèmes neuro-flous.

IV.2. MODELES DYNAMIQUES NON LINEAIRES

En général, un système dynamique est décrit par des équations différentielles s'il est à temps continu ou par des équations aux différences s'il est à temps discret. Dans la pratique, il est rare qu'un système complexe puisse être absolument décrit par un modèle de connaissance. On a souvent recours aux modèles entrées-sorties de type « boîtes noires », pour lesquels aucune connaissance sur le système n'est nécessaire, mais des mesures sur les variables régissant le fonctionnement du système sont indispensables et en quantité suffisante. Le problème de modélisation devient alors un problème de régression non linéaire.

Pour modéliser un système donné, on distingue différents types de modèles non linéaires. Selon le choix du vecteur de régression, différentes structures du modèle non linéaire émergent. Chacune des structures non linéaires NFIR, NARX, NOE et NARMAX est une solution éventuelle.

Tous ces modèles peuvent être exprimés comme suit :

$$y_m(k) = f[\varphi(k)] \tag{IV.1}$$

Où : u, y : les entrées et les sorties du système, respectivement.

$e = y - \hat{y}$: représente l'erreur de prédiction ou résidu.

n, m : les retards entrées et sorties, respectivement.

$f(\cdot)$: la fonction non linéaire du réseau.

$\varphi(k)$: le vecteur de régression.

y_m : la sortie du modèle.

1) Structure NARX

Le modèle NARX (*Nonlinear AutoRegressive with eXogenous input*) permet de représenter des systèmes dynamiques non linéaires dont la sortie dépend des entrées passées et des sorties mesurées passées.

$$\varphi(k) = [u(k-1), \dots, u(k-m), y(k-1), \dots, y(k-n)]^T \quad (IV.2)$$

2) Structure NFIR

Le modèle NFIR (*Nonlinear Finite Impulse Response*) permet de représenter des systèmes non linéaires dont la sortie ne dépend que des entrées à l'instant courant et/ou aux instants antérieurs.

$$\varphi(k) = [u(k-1), u(k-2), \dots, u(k-m)]^T \quad (IV.3)$$

3) Structure NARMAX

NARMAX désigne (*Nonlinear AutoRegressive Moving Average with eXogenous input*). Ce modèle permet de représenter un système dynamique non linéaire par une estimation de la sortie à un instant t à partir des entrées passées, des sorties mesurées passées et des erreurs de prédiction passées.

$$\varphi(k) = [[u(k-1), \dots, u(k-m), y(k-1), \dots, \\ \dots, y(k-n), e(k-1), \dots, e(k-n)]^T \quad (IV.4)$$

4) Structure NOE

Le modèle NOE (*Nonlinear Output Error*) permet de représenter des systèmes dynamiques non linéaires par estimation de la sortie à un instant t à partir des entrées antérieures et des sorties antérieures estimées.

$$\varphi(k) = [u(k-1), \dots, u(k-m), y_m(k-1), \dots, y_m(k-n)]^T \quad (IV.5)$$

La figure (IV.1) illustre les différentes structures des modèles non linéaires.

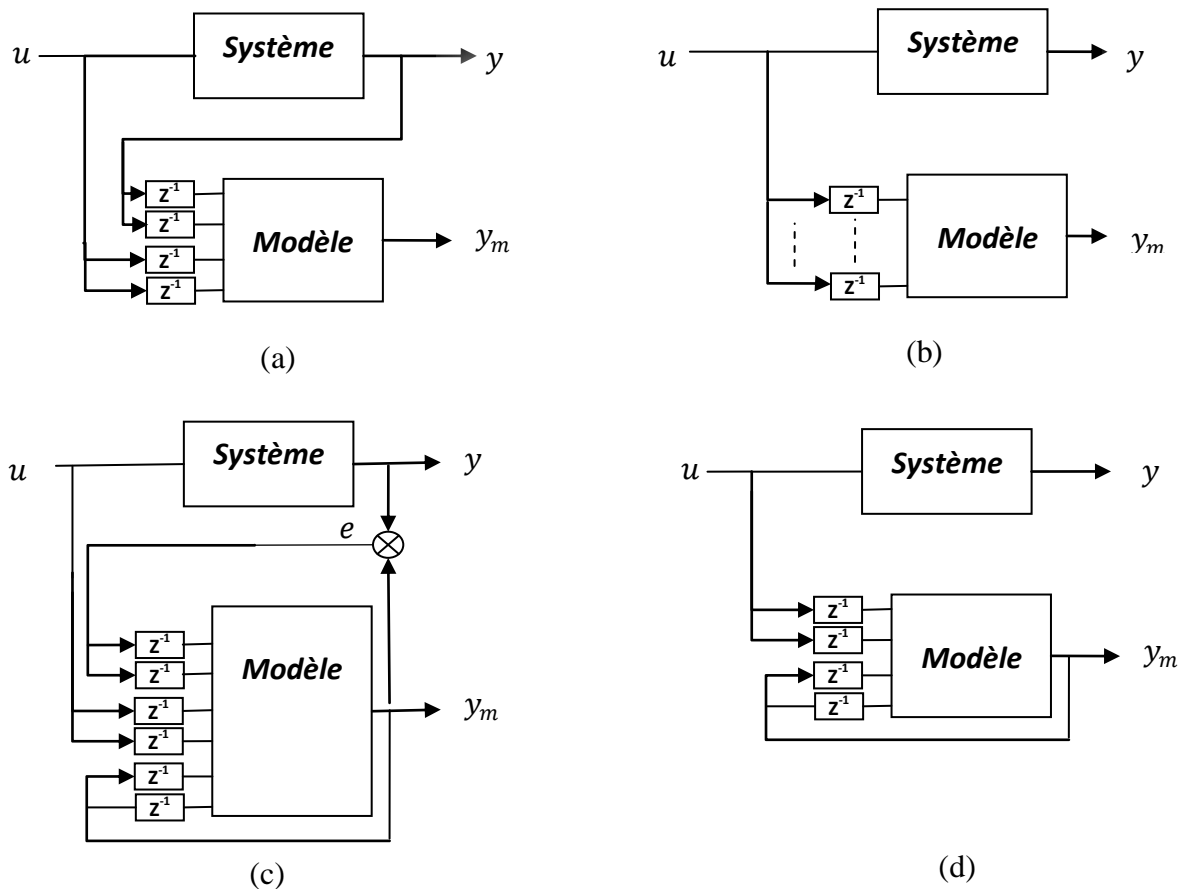


Figure IV.1 : Les structures des modèles, (a): NARX, (b): NFIR, (c): NARMAX et (d): NOE.

Généralement, les structures où il ya des relations algébriques pures entre la prédiction (la sortie estimée) et les mesures précédentes des entrées et des sorties, dans ce cas, la stabilité est accomplie. Cet avantage est important dans le cas non linéaire puisque la recherche de la stabilité est plus complexe ici que dans les systèmes linéaires.

Particulièrement, la structure NARX représente le meilleur choix quand le système est déterministe ou faiblement bruité. Par contre, est plus difficile d'analyser les propriétés de stabilité pour les modèles NARMAX et NOE.

Donc, dans ce cas dernier, il serait judicieux de considérer la stabilité comme une propriété locale. Il n'en demeure pas moins qu'un modèle NARMAX soit stable dans certains régimes et instable dans d'autres. Dans l'un ou l'autre cas, le problème en pratique va dépendre du comportement du système [65,67].

IV.3. DIAGNOSTIC DES DEFAUTS NEURO/FLOUS

Généralement, le diagnostic est une tâche très complexe et malheureusement, les techniques analytiques classiques souvent ne peuvent pas fournir les solutions acceptables aux problèmes de conception. Ceci explique pourquoi les techniques de l'intelligence artificielle telle que les réseaux de neurones et la logique floue deviennent de plus en plus populaires dans des applications industrielles de diagnostic des défauts. L'utilisation de techniques d'Intelligence Artificielle permet d'obtenir des résultats interprétables et offre une information utile pour la décision d'actions à effectuer en cas de défauts.

Comme il est mentionné dans le premier chapitre, la tâche de diagnostic est composée en deux étapes : une étape de génération des résidus et une étape de prise de décision. L'étape de génération des résidus est une étape qui permet de générer des signaux résiduels avec les entrées et les sorties disponibles à partir de système à suivi. Le processus de génération de résidus est basé sur une comparaison entre le comportement observé du système et le comportement de référence attendu (prédit par un modèle). En revanche, l'étape de prise de décision consiste à évaluer les résidus dans le but de classer les défauts détectés.

Le résidu devrait être proche de zéro dans les conditions normales (pas de défauts). Au contraire, en présence de défauts, la valeur de ce résidu sera non nulle. Avec des erreurs de modélisation et de bruits de mesures, un résidu ne peut pas rester identiquement à zéro en l'absence de défauts. Il en résulte que la précision du modèle décrivant le comportement du système est essentiel dans la détection de défaut à base modèle.

Une fois les résidus générés, ils doivent être évalués pour déterminer la présence ou non d'un défaut. L'étape d'évaluation du résidu doit permettre de répondre aux questions suivantes :

- Y a-t-il un défaut ?
- Si oui, quel défaut est présent ?

L'obtention d'un générateur de résidu dans le cas des systèmes non linéaires par les méthodes quantitatives classiques n'est pas une chose aisée. Il serait alors plus judicieux d'utiliser d'autres méthodes capables de modéliser des fonctions non linéaires et de générer les fonctions résidus. L'utilisation des méthodes d'intelligence artificielle, tels que les réseaux de neurones, systèmes experts, systèmes flous et les systèmes neuro-flous ont été utilisés avec succès dans le modèle de base.

La figure (IV.2) illustre le principe le plus général pour la génération et l'évaluation des résidus.

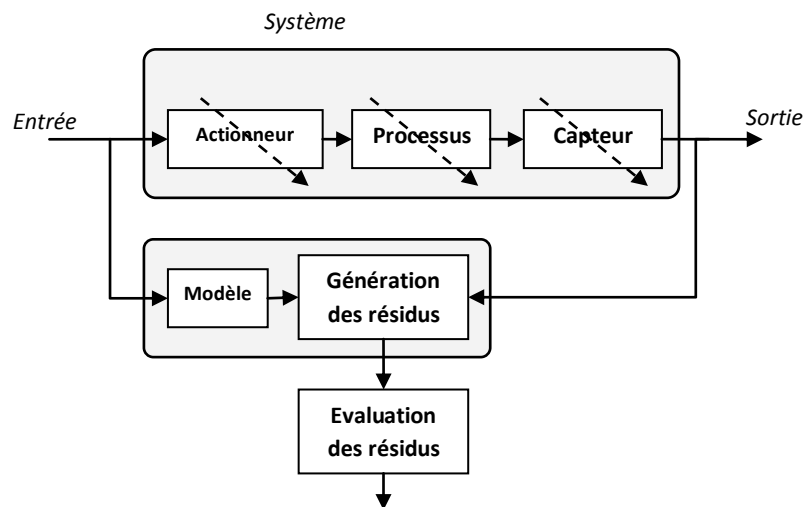


Figure IV.2 : Principe de génération et d'évaluation des résidus.

Dans ce que suit, nous présentons avec détails l'utilisation des réseaux de neurones, la logique floue et des réseaux neuro-flous dans le but de générer les fonctions résidus.

IV.3. 1. Génération des résidus neuro-flous

L'inconvénient major des méthodes analytiques utilisées dans le domaine de diagnostic est le fait que l'utilisation d'un modèle mathématique précise est nécessaire. Le modèle mathématique utilisé dans les méthodes de diagnostic et d'isolation des défauts "FDI" traditionnelles peuvent être très sensibles aux erreurs de modélisation, la variation des paramètres, le bruit et les perturbations. Pour éviter certaines des difficultés d'utilisation des modèles mathématiques, il est très important de faire choisir des algorithmes FDI plus applicables à des systèmes réels.

IV.3. 1. 1. Génération des résidus à base de modèle neuronal

Les techniques d'intelligence artificielle tels que les réseaux de neurones, la logique floue, les systèmes experts...etc., peuvent être utilisés à la fois pour générer des résidus et isoler les défauts.

Toutefois, aucun modèle mathématique du système n'est nécessaire pour mettre en œuvre un réseau de neurones. L'apprentissage en ligne permet de changer le système de diagnostic facilement quand des modifications sont apportées dans le processus physique, le système de commande ou les paramètres. Les réseaux de neurones ont la capacité de prendre des décisions intelligentes en cas de bruit ou des données corrompues. Les réseaux de neurones peuvent fonctionner simultanément sur des données qualitatives et quantitatives, et ils sont facilement applicables aux systèmes multivariables.

L'une des plus importantes classes de méthodes FDI, en particulier consacrée aux processus non linéaires, est l'utilisation des réseaux de neurones artificiels. Il existe de nombreuses structures neuronales qui peuvent être utilisées. Les réseaux de neurones peuvent être aussi utilisés à deux étapes : l'étape de génération des résidus et l'étape d'évaluation des résidus.

Dans la première étape, les réseaux de neurones sont utilisés comme des modèles de prédiction. Une caractéristique importante d'un modèle de prédiction basé sur les réseaux de neurones est qu'il est automatiquement «apprendre» la dynamique du système non linéaire au cours du processus d'apprentissage réalisé au cours de plusieurs cycles d'apprentissage, avec des données d'apprentissage provenant de données historiques sur les entrées-sorties du système.

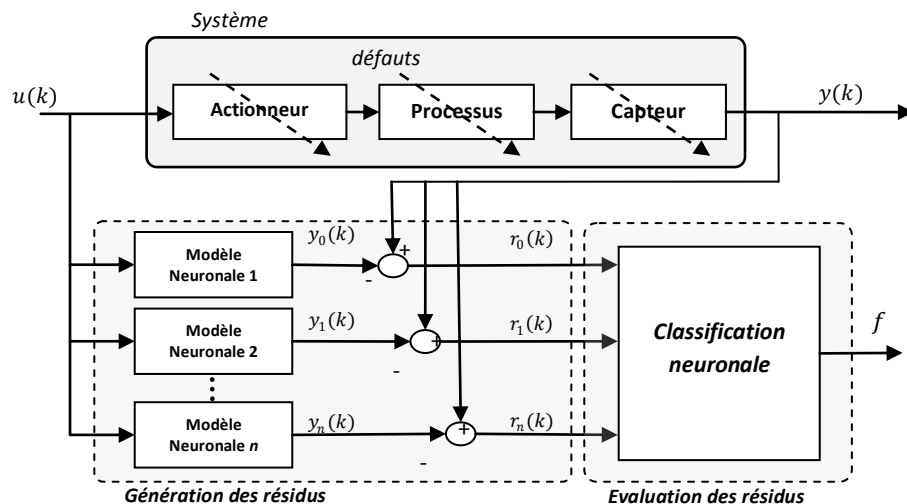


Figure IV.3 : Génération et évaluation des résidus par réseaux de neurones.

Les modèles prédictifs basés sur les réseaux de neurones ont des avantages par rapport aux méthodes de prédiction traditionnelles et les méthodes d'estimation. À ce jour, de nombreuses structures neuronales avec des caractéristiques dynamiques ont été développées. Ces structures sont efficaces dans la modélisation des processus non linéaires.

En général, toutes les structures connues peuvent être divisées en deux groupes :

- Réseaux de neurones avec des dynamiques externes.
- Réseaux de neurones avec des dynamiques internes.

1) Réseaux de neurones avec des dynamiques externes

Les réseaux de neurones avec dynamiques externes sont des réseaux de type statique et peuvent être utilisés pour approximer n'importe quelle fonction non-linéaire. Par conséquent, la modélisation par réseaux de neurones des systèmes de commande devrait tenir compte la dynamique des processus ou des systèmes considérés. Les réseaux de neurones les plus couramment utilisés pour la modélisation des processus sont de type «perceptron multicouche MLP».

Par exemple, un système dynamique non linéaire peut être représenté par des réseaux de neurones de type MLP. La structure du réseau dépend de la classe du modèle définie par son vecteur de régression. Pour un modèle NARX, le vecteur de régression est donné par (IV. 2).

La figure (IV.4) représente l'implantation d'un modèle NARX avec une structure MLP.

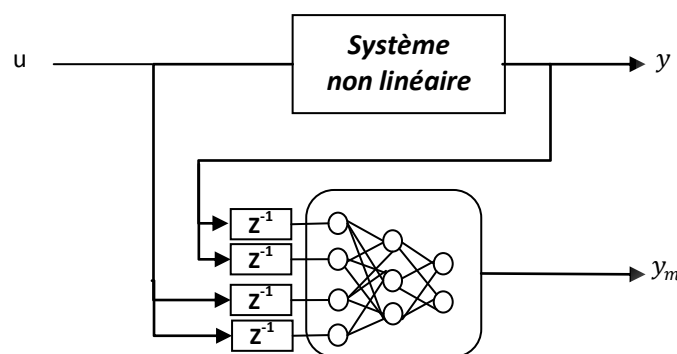


Figure IV.4 : Exemple d'implantation d'un modèle NARX par un réseau MLP.

2) Réseaux de neurones avec des dynamiques internes

La caractéristique principale de la dynamique de réseau de neurones est le fait qu'elle possède la mémoire par la présentation de récurrence globale ou locale. Donc, dans ce cas là, il est possible de mémoriser les informations et les utiliser plus tard. En général, les réseaux de neurones globalement récurrents, en dépit de leur utilisation, ont quelques inconvénients. Ces architectures souffrent d'un manque de stabilité. Pour un ensemble donné de valeurs initiales, les activations de sortie des neurones linéaires peuvent se développer illimitée.

Une autre solution, qui fournit le comportement dynamique du modèle neuronal, est un réseau conçu à l'aide des modèles neuronaux dynamiques. Ces réseaux ont une architecture qui est quelque part entre deux types d'architectures, une architecture feed-forward et une architecture globalement récurrente.

Différents modèles neuronaux dynamiques peuvent être conçus dont les modèles dynamiques les plus connus sont : ceux avec retour local des informations d'activation, avec retour local des informations de la synapse, retour local des informations de sortie et les filtres RII. Rappelons par exemple: le réseau d'Elman, et le réseau de Jordan [52,61].

IV.3. 1. 2. Génération des résidus à base de modèle flou

Les techniques floues ont reçu une attention particulière en raison de leur mise en œuvre rapide robuste, leur capacité à intégrer a priori des connaissances, leurs performances de reproduction de correspondances non linéaires, et leur capacité de généralisation.

La description de certains systèmes non linéaires peut être très difficile à réaliser au moyen des équations analytiques. L'utilisation de la théorie des systèmes flous est un outil naturel pour traiter des conditions non linéaires et incertaines. L'utilisation des modèles flous augmente la capacité du système de diagnostic à travailler avec des systèmes caractérisés par des informations incomplètes et bruitées.

Le but de modélisation floue reste de concevoir un modèle flou en vue de la génération de résidus. Le concept général de la génération de résidus reste aussi le même que pour les modèles analytiques. Il consiste à comparer les sorties du procédé à leurs estimées, mais dans ce cas, les estimées sont calculées par un modèle flou.

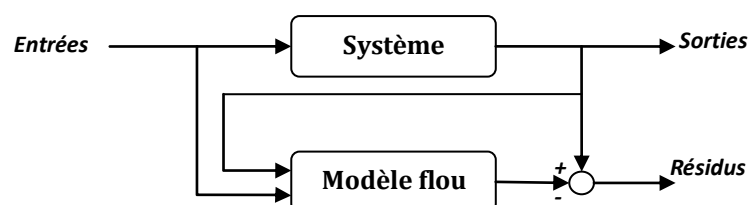


Figure IV.5 : Génération des résidus à base de modèle flou.

D'une manière générale, les modèles flous sont des ensembles de règles floues qui se différencient de par leurs structures (nombre de règles, type de variables, nombre et type de fonctions d'appartenance) et de la technique d'identification utilisée pour estimer les paramètres des règles et des fonctions d'appartenance.

- Un modèle flou utilisé pour la génération de résidus est celui proposé par Takagi et Sugeno. La structure *NARX*, constitue le meilleur choix pour la structure des modèles non linéaires. Un système peut être représenté comme un modèle multi-entrées/multi-sorties (MIMO), et peut être aussi décomposé en plusieurs modèles, multi entrées/une sortie (MISO) ou une entrée/une sortie (SISO).

$$y_m(k) = f[x(k)] \quad (IV.6)$$

Où $x(k) \in \mathfrak{R}^n$ est l'état du système, et contient les entrées et les sorties précédentes. Nous considérons dans ce qui suit des modèles MISO. Les règles de modèle de type Takagi-Sugeno affine sont représentées sous la forme :

$$\begin{aligned} R_i: & \text{ Si } x_1 \text{ est } A_{i1} \text{ et } \dots \text{ et } x_n \text{ est } A_{in}, \\ & \text{ Alors } y_i = a_{i1}x_1 + \dots + a_{in}x_n + a_{i(n+1)} \end{aligned} \quad (IV.7)$$

La sortie finale est obtenue en utilisant la formule générale suivante :

$$\hat{y} = \frac{\sum_{i=1}^K \mu_{Ai}(x) y_i}{\sum_{i=1}^K \mu_{Ai}(x)} \quad (IV.8)$$

$i = 1, \dots, r$ et r le nombre des règles floues.

$\mu_{Ai}(x)$ est le degré d'activation de la i ème règle.

- L'approche la plus fréquente de modèles flous pour la génération de résidus concerne les modèles relationnels flous. Ces derniers ne sont pas utilisés comme des modèles parallèles mais comme des observateurs de sortie tels que la sortie estimée est calculée par la relation suivante :

$$\hat{y}(k) = R \circ X(k) \quad (IV.9)$$

où X est une combinaison de toutes les prémisses de règles calculées et R la matrice relationnelle correspondant à l'ensemble des règles, " \circ " étant un opérateur de composition T-conorme / T-norme selon le principe de combinaison – projection.

La résolution de l'équation floue donnée par l'équation (IV.9) consiste à trouver dans le cas de N observations, une solution vérifiant simultanément les N équations. L'ensemble des solutions est généralement vide. L'approche classique consiste alors à trouver une solution R pour laquelle la sortie mesurée est proche de la sortie estimée au sens de la minimisation d'un critère $J(R)$ donné.

Celui-ci peut par exemple être l'erreur quadratique [53] :

$$J(R) = \frac{1}{2} \sum_{k=1}^N [y(k) - \hat{y}(k)]^2 \quad (IV.10)$$

avec N le nombre d'éléments de l'ensemble d'apprentissage.

IV.3. 1. 3. Génération des résidus à base de modèle neuro-flou

Les réseaux neuro-flous sont nés de l'association des réseaux de neurones avec la logique floue, de manière à tirer profits des avantages de chacune de ces deux techniques. La caractéristique principale des modèles neuro-flous pour le diagnostic industriel est la capacité de modéliser des processus non linéaires et de traiter dans un même outil des connaissances numériques et symboliques d'un système [70]. Dans des applications de diagnostic, on trouve principalement des modèles neuro-flous hybrides, pour lesquels les réseaux de neurones et les systèmes flous sont combinés de manière homogène.

Les réseaux neuro-flous apparaissent comme des outils puissants combinant des grandes capacités d'approximation pour la modélisation des systèmes dynamiques non linéaires. Pour développer un système de diagnostic à l'aide des modèles neuro-flous, plusieurs méthodes sont disponibles, mais, malheureusement on trouve quelques points négatifs. Tout d'abord au niveau de la conception de ces systèmes, il est nécessaire d'avoir des connaissances a priori suffisantes sur le système à diagnostiquer pour déterminer les variables d'entrées, les fonctions d'appartenance, ainsi que les règles. Donc, il serait intéressant d'avoir des précisions sur la possibilité d'effectuer un diagnostic en ligne.

La plupart des applications rencontrées sont basées sur l'établissement d'un diagnostic à partir de la classification de résidus, elles nécessitent donc de pouvoir établir un modèle du système. De plus, elles nécessitent un réseau par résidus ce qui rend le système relativement complexe et ne permet de diagnostiquer qu'un nombre restreint de défauts. Il serait donc intéressant d'employer ces techniques compte tenu de leurs capacités en s'affranchissant complètement d'un modèle du système à diagnostiquer.

Les réseaux neuro-flous hiérarchiques peuvent être utilisés pour résoudre le problème de dimensionnalité en décomposant le système en une série des systèmes MISO et/ou des systèmes SISO appelés systèmes hiérarchiques. Les règles locaux utilisent des sous-ensembles de l'espace d'entrée et sont activées à des règles plus hauts niveaux.

Les critères sur lesquels construit un modèle neuro-flou sont fondés sur les exigences du modèle de diagnostic de défaut et les caractéristiques du système. Les vecteurs sont obtenus en comparant les sorties du processus et les sorties du modèle. La valeur résiduelle doit être égale à zéro dans l'état sans défaut et vice versa.

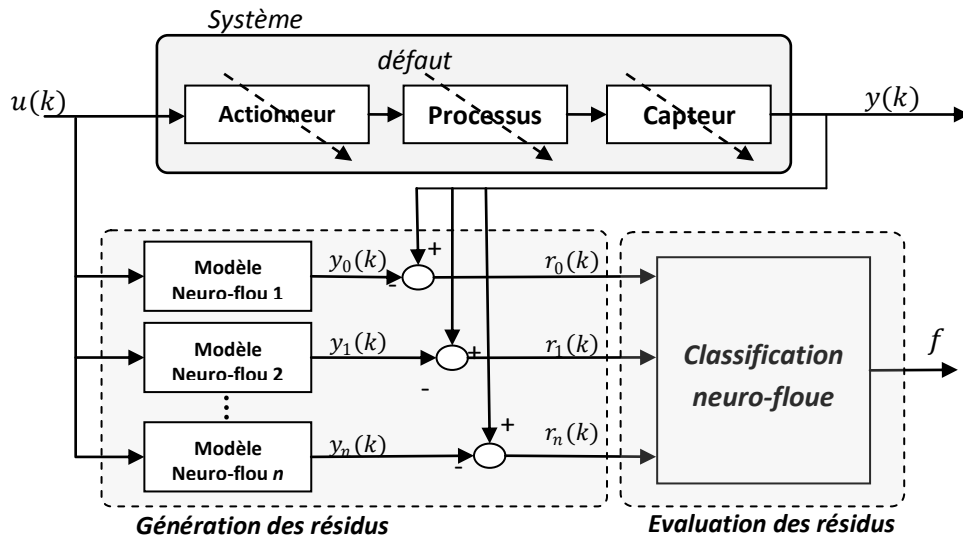


Figure IV.6 : Génération et évaluation des résidus neuro-flous.

IV.3. 2. Evaluation des résidus

Une fois les résidus générés, la deuxième étape consiste à évaluer ces résidus dans le but de classifier les défauts détectés. Pour cela nous allons intéresser, à des techniques d'évaluation ou de classification à base de réseaux de neurones, de logique floue, et neuro-flou.

IV.3. 2. 1. Evaluation des résidus à base de modèle neuronal

L'évaluation des résidus est un processus de prise de décision logique qui transforme les connaissances quantitatives à des états qualitatives. Il peut également être considéré comme un problème de classification. Une variété d'approches et de techniques bien établies (seuil, seuil adaptatif, méthodes statistiques et de classification) peuvent être utilisées pour l'évaluation des résidus. Parmi ces approches, les méthodes de classification floue et neuronale sont très attractives et les plus fréquemment utilisées dans les systèmes FDI.

Les réseaux de neurones sont des outils de l'intelligence artificielle, souvent utilisés pour des tâches de classification. L'avantage de l'utilisation des réseaux de neurones dans le cas de la classification est qu'ils peuvent construire des frontières de décision linéaires ou non linéaires entre les différentes classes d'une manière non paramétrique et offrent une méthode pratique pour résoudre les problèmes complexes de classification.

Différentes structures de réseaux de neurones statiques peuvent être appliquées, y compris le perceptron multicouche, les réseaux de Kohonen, les réseaux RBF et les structures de réseaux multiples.

1) Structure de réseaux multiples

Dans de nombreux cas, un seul réseau de neurones d'une taille finie, par exemple, un perceptron multicouche ou un réseau RBF, ne garantit pas l'architecture nécessaire ou sa capacité de généralisation n'est pas suffisante.

Pour améliorer la qualité de classifieurs neuronaux, des systèmes de réseaux multiples sont proposées. L'idée du système de réseaux multiples est de combiner de manière indépendante n réseaux pour n points de fonctionnement et de classer un modèle d'entrée donnée par un module de décision.

La décomposition d'un problème de classification complexe peut être réalisée en utilisant de façon indépendante des classifieurs neuronaux conçus de telle manière que chacun des ces classifieurs d'être capable de reconnaître seulement quelques classes. Le module de décision en connecter avec la porte logique dans le cadre de décider quel classifieur neuronal doit être sélectionné.

En outre, chaque classifieur neuronal peut avoir une structure optionnelle, mais la seule condition est que chaque classifieur neuronal devrait avoir le même nombre de sorties.

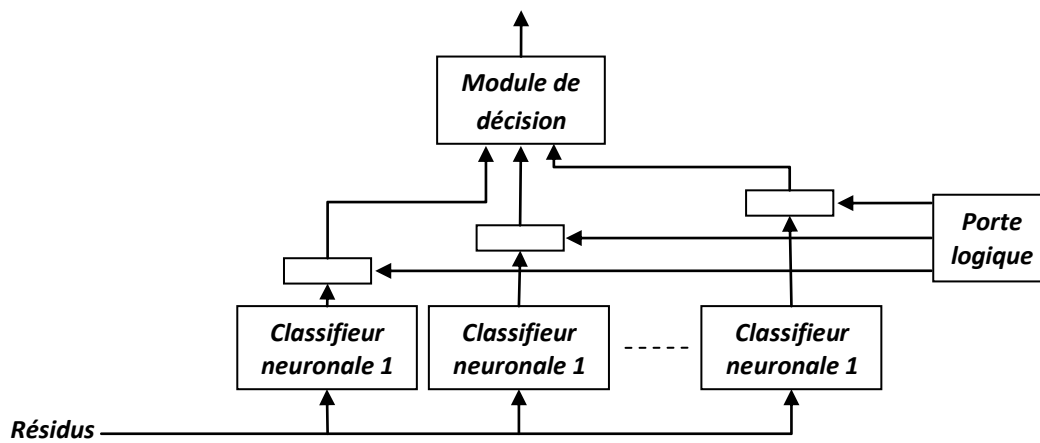


Figure IV.7 : Structure de réseaux multiples.

2) Classifieur de Kohonen

Le réseau de Kohonen est une carte d'auto-organisation dans lequel les paramètres sont adaptés par une procédure d'apprentissage non supervisé basé uniquement sur les modèles d'entrée. Contrairement aux méthodes standards d'apprentissage supervisé, les paramètres non supervisés utilisent les signaux d'entrée pour extraire des connaissances à partir des données.

Chaque neurone de la couche d'entrée est relié à tous les neurones de la carte, dans la plupart des applications, les cartes sont à deux dimensions. A chaque cellule est associé un voisinage qui dépend de la géométrie du maillage de la carte et auquel correspondent des mécanismes d'interaction latérale. Cette interaction latérale montre que les neurones proches apportent une activation, alors que les neurones plus éloignés ont une action inhibitrice.

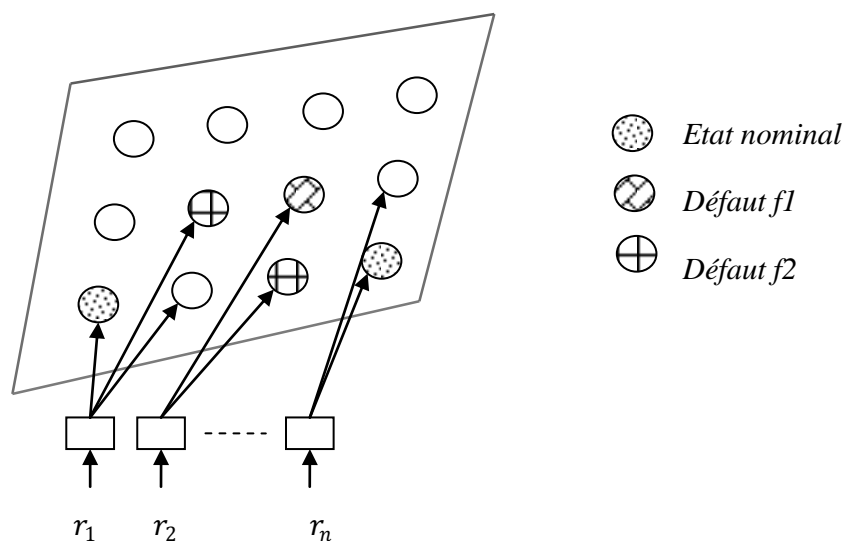


Figure IV.8 : Classifieur de Kohonen en deux dimensions.

Un exemple du processus d'apprentissage est représenté dans la figure (IV.8), où trois états sont considérés (deux défauts $f1$, $f2$ et l'état normal). Si, après le processus d'apprentissage, il ya des régions qui se chevauchent, alors la taille de la carte (le nombre de neurones) devrait être augmenté et le processus de formation répété depuis le début.

Les réseaux de Kohonen ont trouvé de nombreuses applications pour la classification, mais l'inconvénient majeur est d'exiger un temps relativement long pendant la phase d'apprentissage [61].

IV.3. 2. 2. Evaluation des résidus à base de modèle flou

L'utilisation la plus fréquente de la logique floue dans les méthodes FDI est dans l'étape d'évaluation des résidus. Il y a trois principales approches dans le processus de décision : le seuil adaptatif flou, la classification floue et le raisonnement flou.

a) Classification floue

La classification floue est une idée vient directement de l'approche de reconnaissance de forme. Dans cette approche dernière, le cadre de la modélisation de l'imprécision (incertitude) est différent que ce soit pour construire l'espace de représentation ou pour prendre la décision. Les différentes classes floues sont construites par apprentissage en utilisant une base de données qu'est effectuée en hors ligne. L'apprentissage permet aussi de déterminer les centres de classes les plus significatifs et ceci pour chaque défaut étudié. Cette méthode est très utile quand on a plusieurs résidus ou dans le cas où aucune connaissance experte sur le système n'est disponible.

L'espace de classification est donc défini d'une façon graduelle qui est très intéressante pour le diagnostic en ligne. La mise en œuvre d'une méthode de classification floue implique deux étapes : la construction des fonctions d'appartenance et la définition des règles de décision.

- a. Sélectionner une partition floue initiale des N éléments dans K classes en sélectionnant la matrice d'appartenance U de dimension $N \times K$. Un élément u_{ij} de cette matrice représente le degré d'appartenance de l'élément x_i dans la classe c_j . Typiquement, $u_{ij} \in [0, 1]$.
-

- b. Minimiser la valeur de la fonction critère floue, en modifiant de façon itérative la matrice U de manière à modifier l'assignation des éléments aux classes et ainsi faire décroître le critère.
- c. Répéter l'étape 2 jusqu'à ce que la valeur de U ne change pas significativement.

Avec cette méthode, chaque classe est un ensemble flou de tous les éléments. Le « *C-Means Flou* » (FCM) est le plus connu des algorithmes de classification floue.

b) Seuil adaptatif flou

Dans cette deuxième méthode, l'évaluation des résidus a été développée comme un problème de seuillage des résidus. L'utilisation des seuils fixes introduit l'inconvénient de mal gérer le compromis détection/fausses alarmes. Pour éviter ces inconvénients, un système basé sur la logique floue est utilisé pour obtenir des seuils adaptatifs, qui varient selon les conditions opératoires du processus. Cette méthode est très utile quand un modèle linéaire est utilisé pour la génération de résidu, alors que le processus réel est non linéaire, elle diminue le taux de fausses alarmes dues aux non linéarités.

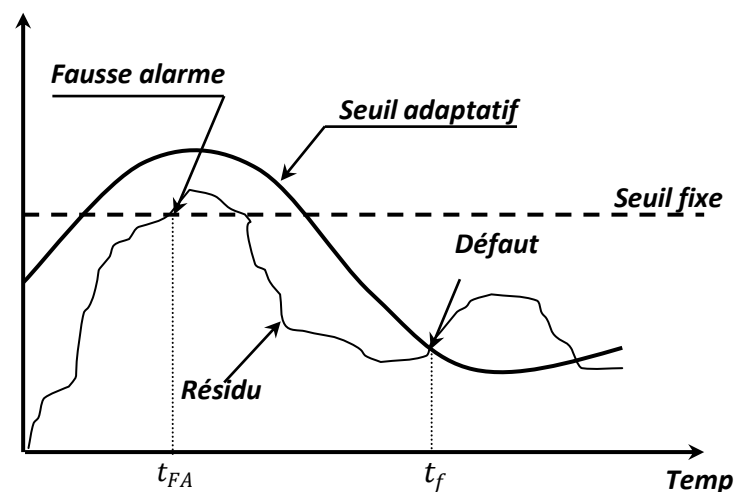


Figure IV.9 : Test d'un seuil adaptatif.

Le concept flou permet de déterminer des relations floues qui sont exprimées par des implications floues au lieu d'utiliser des relations analytiques. Le seuil est adapté suivant les variations des entrées u et des sorties y en terme de règles entre les ensembles flous qui sont spécifiés par des fonctions.

La relation résultante pour le seuil adaptatif flou est donnée par :

$$J(u, y) = J_0 + \Delta J(u, y) \quad (IV.11)$$

Où $J_0 = J_0(u_0, y_0)$ est le seuil nominal constant dû à l'influence des perturbations et des bruits.

$\Delta J(u, y)$ représente les effets des erreurs de modélisation.

Les quatre étapes principales pour la sélection d'un seuil adaptatif flou peuvent être définies comme suit :

- a. L'observation des relations entre les fausses alarmes et les conditions caractéristiques de procédés.
- b. La formulation des règles, qui sont organisées suivant la structure r : "Si ... Alors ...".
- c. Le choix des variables et des fonctions d'appartenances floues appropriées.
- d. Définition d'une table de règles floues basées sur les étapes 2, 3.

Après l'organisation initiale des fonctions d'appartenance et d'une base de règles floues, de nouvelles connaissances peuvent être insérées, en changeant les règles ou en introduisant de nouvelles variables floues si nécessaire [7].

Le diagramme structurel de la procédure d'évaluation utilisant la technique du seuil adaptatif flou est donné par la figure (IV.9) :

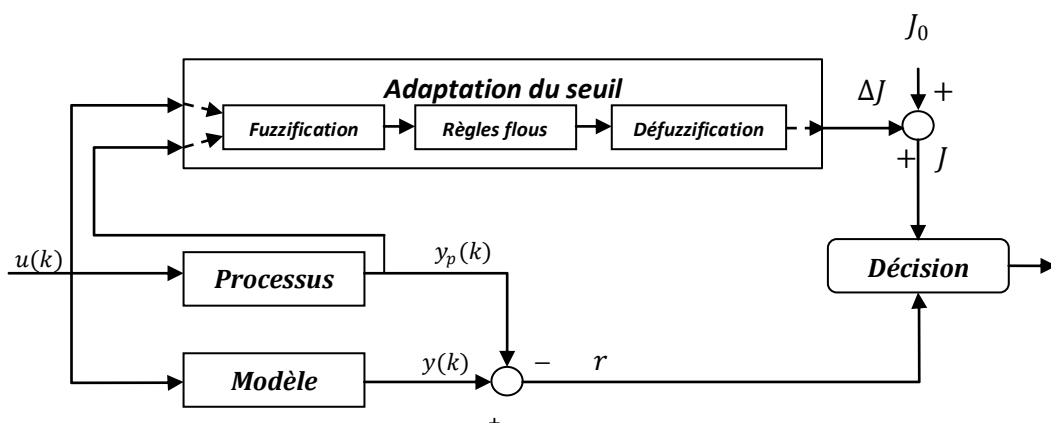


Figure IV.10 : Evaluation des résidus par un seuil adaptatif flou.

c) Raisonnement flou

L'avantage principal du raisonnement flou, est qu'il peut principalement introduire des informations heuristiques dans le schéma d'analyse. Les schémas des raisonnements flous sont aussi faciles à comprendre, car ils sont très similaires à la manière avec laquelle les humains résolvent des problèmes d'analyse. L'idée de base de l'utilisation de l'inférence floue pour l'analyse des résidus, vient du fait que chaque résidu est zéro, positif ou négatif par rapport à un certain degré. Un schéma flou comporte trois étapes :

▪ La fuzzification :

C'est la transformation des valeurs données brutes en des valeurs d'entrée floues. Pour cela, on détermine pour chaque entrée et sortie sa fonction d'appartenance floue.

▪ L'inférence :

Cette étape fait la détermination de la base des règles qui sont formées pour déterminer les conditions sous lesquelles le défaut existe et sous lesquelles le système est non défaillant.

Par exemple :

- ✓ Si le résidu 1=0 et le résidu 2=0, alors le système est non défaillant.
- ✓ Si le résidu 1 est >0 et le résidu 2 <0, alors le défaut 1 est présent.

Si les règles ne reflètent pas l'expérience d'un opérateur, alors elles peuvent être difficilement validées.

▪ La défuzzification :

La construction de valeurs de sorties brutes à partir des ensembles d'inférence. La sortie du processus de la décision logique, est une valeur qui donne le degré avec lequel un défaut est présent dans le système, plutôt qu'un constat simple de défaut/non défaut. Le degré peut être une indication aussi bien de la taille du défaut présent, que de la certitude avec laquelle un défaut est présent dans le système. Une telle sortie est donnée pour chaque défaut considéré. L'utilisation de la méthode de raisonnement flou. L'absence de méthodes formelles de conception représente un des inconvénients majeurs pour réaliser des schémas FDI.

IV.3. 2. 3. Evaluation des résidus à base de modèle neuro-flou

L'utilisation des approches neuro-floues, dans l'étape d'analyse des résidus et de décision est dictée par la capacité d'apprentissage des réseaux de neurones et la souplesse offerte par la logique floue. Différentes associations de réseaux de neurones et de la logique floue peuvent être utilisées pour analyser les résidus. Il existe ainsi trois architectures neuro-floues :

1) Première architecture neuro-floue :

Dans cette architecture, les réseaux de neurones et les systèmes flous sont associés en série ou en parallèle, ce qui permet de construire plusieurs possibilités d'association :

- On peut avoir un réseau de neurones qui fonctionne en amont du système flou. Les variables d'entrées d'un système à commande floue sont, par exemple, déterminées à partir de la sortie d'un réseau de neurones, ou bien par un réseau de neurones qui effectue une tâche de classification ou de reconnaissance de forme, et qui est suivi par un système d'aide à la décision floue.
- On peut aussi avoir un réseau de neurones qui fonctionne en aval du système flou, dans le but d'ajuster les sorties d'un système flou à de nouvelles connaissances obtenues, les variables de sorties étant les erreurs sur les variables de sortie du système flou.

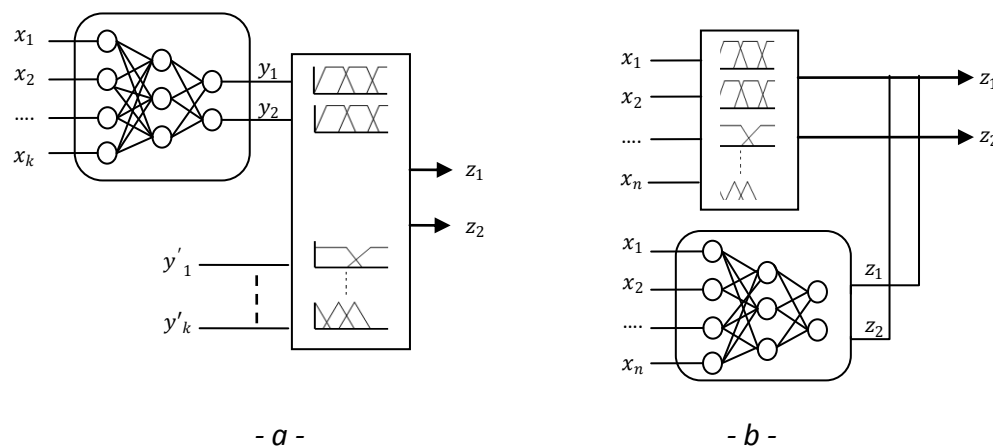


Figure IV.11 : Première architecture des réseaux Neuro-Flous.

a) association en série, b) association en parallèle.

2) Seconde architecture neuro-floue :

Dans cette architecture, un système d'inférence floue est représenté sous la forme d'un réseau multicouche, dans lequel les poids correspondent aux paramètres du système. Le type de règle et les méthodes d'inférence, l'agrégation et le défuzzification choisies déterminent l'architecture du réseau.

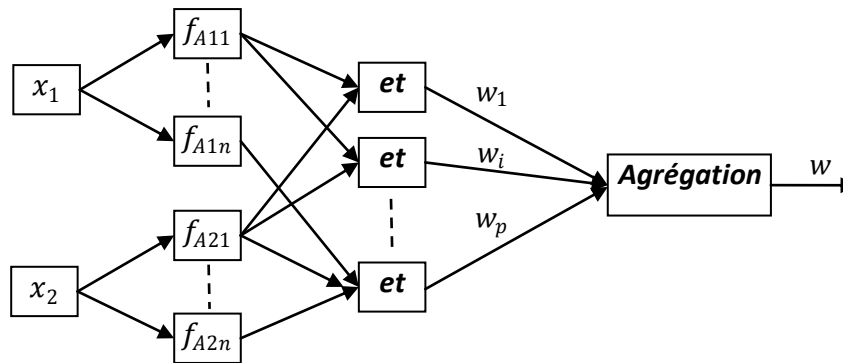


Figure IV.12 : Seconde architecture neuro-floue.

3) Troisième architecture neuro-floue :

Dans cette architecture, les réseaux de neurones sont utilisés pour remplacer chacune des composantes d'un système flou. Ces réseaux servent à l'apprentissage des fonctions d'appartenance, au calcul de l'inférence, à la réalisation de la phase d'agrégation et de défuzzification. Ils peuvent réaliser l'extraction des règles floues, en analysant la corrélation qui existe entre les entrées et les sorties du réseau de neurones. Ces approches sont capables de résoudre des problèmes importants de la logique floue, tels que la détermination et l'apprentissage des fonctions d'appartenances, la détermination des règles et l'adaptation à l'environnement du système.

Nous prenons, par exemple, le schéma suivant qui appartient à la troisième architecture neuro-floue. Ce schéma peut être utilisé dans un problème de diagnostic.

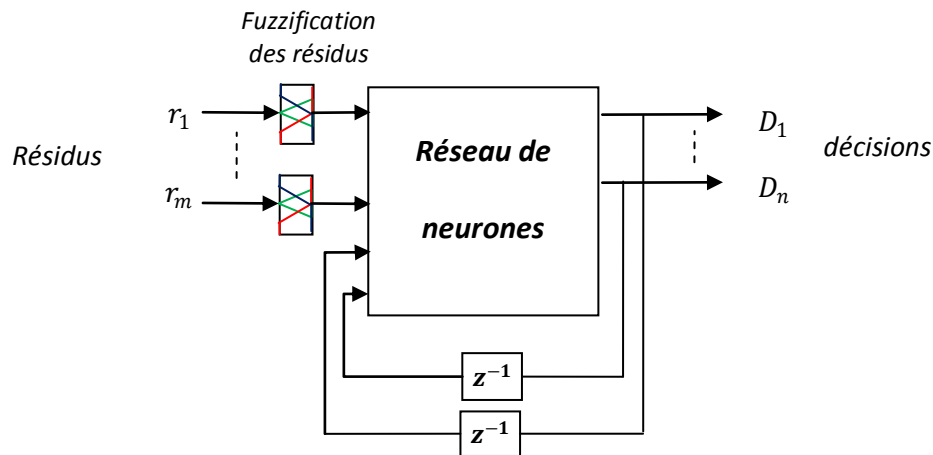


Figure IV.13 : Analyse des résidus basés sur un système neuro-flou.

Pour réaliser cette tâche, plusieurs étapes sont nécessaires.

- **Fuzzification des résidus :**

Dans cette première étape, les résidus sont fuzzifiés. Il faut prendre pour chaque résidu des fonctions d'appartenance qui sont généralement de type triangles ou trapèzes.

- **L'inférence :**

Dans cette étape, nous utilisons un réseau de neurones pour établir une base de règles. Ce réseau est de type récurrent, parce que nous considérons les résidus fuzzifiés et les décisions précédentes comme entrées de réseau et les décisions comme des sorties.

- **Apprentissage :**

Avant d'appliquer en ligne ce réseau, une phase d'apprentissage est nécessaire. On utilise une base de données qui doivent contenir tous les scénarios possibles de défauts et les décisions correspondantes.

IV.4. EXEMPLE DE DIAGNOSTIC

Afin d'illustrer ce que nous venons d'exposer précédemment, nous considérons un système non linéaire proposé dans [46,59]. Ce système est défini comme suit :

$$y_p(k+1) = f(y_p(k), y_p(k-1), y_p(k-2), u(k)) \quad (IV.12)$$

Où :

$$f(x_1, x_2, x_3, x_4) = \frac{x_1(k).x_2(k).x_3(k).x_4(k).[x_3(k) - 1] + x_4(k)}{1 + x_1^2(k) + x_2^2(k)} \quad (IV.13)$$

On peut écrire ce système sous la forme suivante :

$$x_1(k+1) = x_2(k) \quad (IV.14)$$

$$x_2(k+1) = x_3 \quad (IV.15)$$

$$x_3(k+1) = \frac{x_1(k).x_2(k).x_3(k).u(k).[x_3(k) - 1] + u(k)}{1 + x_1^2(k) + x_2^2(k)} \quad (IV.16)$$

Où la période d'échantillonnage prise est de $T_e = 0.1 \text{ sec}$.

$u(k)$: L'entrée du système.

Comme nous avons vu précédemment, le diagnostic des défauts passe par deux étapes principales : la génération et l'évaluation des résidus. Dans la première étape, nous essayons d'abord d'obtenir un modèle du comportement afin de générer les résidus. L'objectif de cette étape est de trouver un modèle qui soit capable de caractériser le comportement du procédé.

- **Obtention d'un modèle du comportement.**

Pour modéliser ce système, nous proposons un modèle basé sur la structure NOE (voir IV.2.4). Nous utilisons aussi la technique neuro-floue qui donne des meilleurs résultats en modélisation. Suivant l'architecture du système, on a choisi un modèle composé de trois entrées et une sortie : $u(k)$, $x_1(k)$ et $x_2(k)$ comme entrées et $x_3(k)$ comme sortie.

Essai 1 :

Nous avons voulu montrer l'effet des incertitudes de modélisations sur les performances du système de détection, en absence de défauts. Ceci est représenté sur les figures (IV.14) jusqu'à (IV.16).

Ces figures illustrent les résultats de simulation effectuées dans le cas d'un système sain, mais soumis à des incertitudes de modélisation. Il faut noter que les résidus varient et s'éloignent de zéro. Ces derniers sont donc sensibles aux incertitudes de modélisation. Ce qui engendre des fausses alarmes et des détections erronées.

Les figures montrent l'entrée, la sortie et l'erreur de modélisation correspondant en respect des changements sur la commande du système. Nous avons réalisé, dans un premier temps, une expérimentation (en l'absence de défaut) pour tester la fiabilité du système.

L'entrée du système est donné par :

$$u(k) = 0.3 \cdot \sin\left(k \cdot \frac{\pi}{40}\right) + 0.4 \cdot \sin\left(k \cdot \frac{\pi}{30}\right) \quad (IV.17)$$

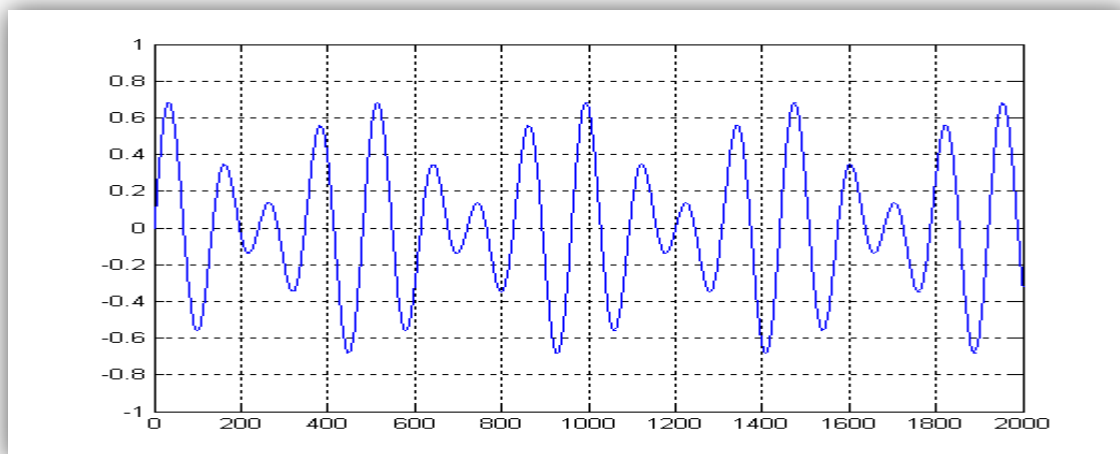


Figure IV.14 : L'entrée du système.

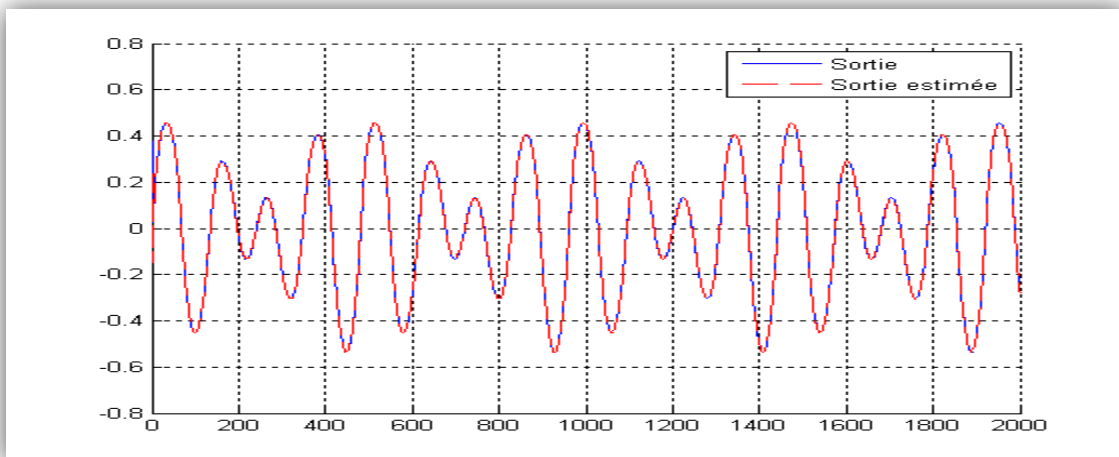


Figure IV.15 : La sortie et la sortie estimée.

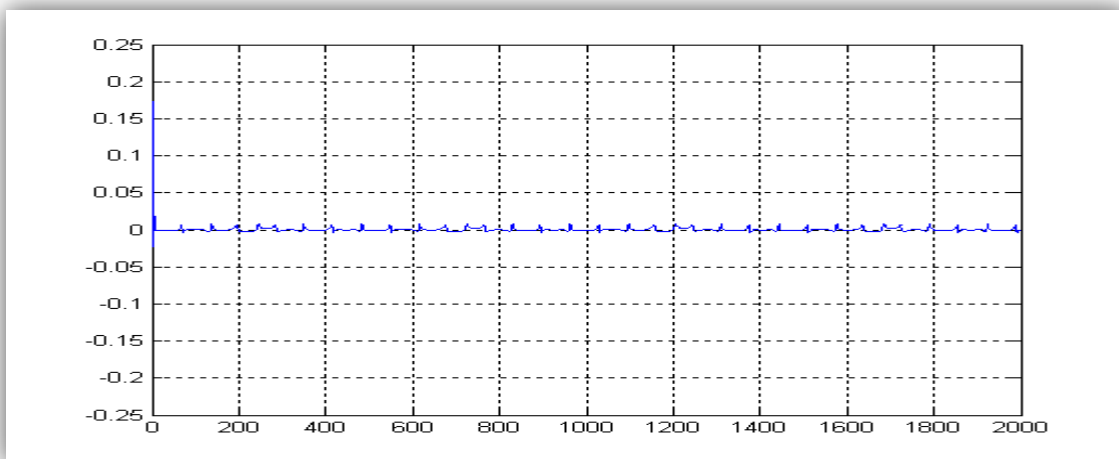


Figure IV.16 : Erreur de modélisation (résidu dans le cas sans défaut).

Donc, cette première étape permet la génération des résidus. Ensuite nous passons à la deuxième étape qui représente l'étape d'évaluation des résidus.

- **L'évaluation des résidus**

Dans cette étape, nous avons proposé un modèle de raisonnement flou, pour classer les défauts. On note que la sortie de ce modèle de décision flou prend soit la valeur 0 dans le cas d'absence de défauts ou la valeur 1 dans le cas de présence de défauts.

Pour montrer l'efficacité de système de diagnostic proposé, nous allons simuler des différents types de défauts dans différents temps et les résultats de simulation sont montrés dans ce que suit. Dans tous les scénarios simulés, l'objectif souhaité en terme de référence est fixé à 0.5.

Essai 2 : En absence de défauts.

Cet essai montre l'entrée et la sortie du système nominal, la sortie du modèle, le résidu et la décision dans le cas d'absence de défaut. Les résultats sont illustrés sur les figures (IV. 17) jusqu'à (IV. 20) respectivement.

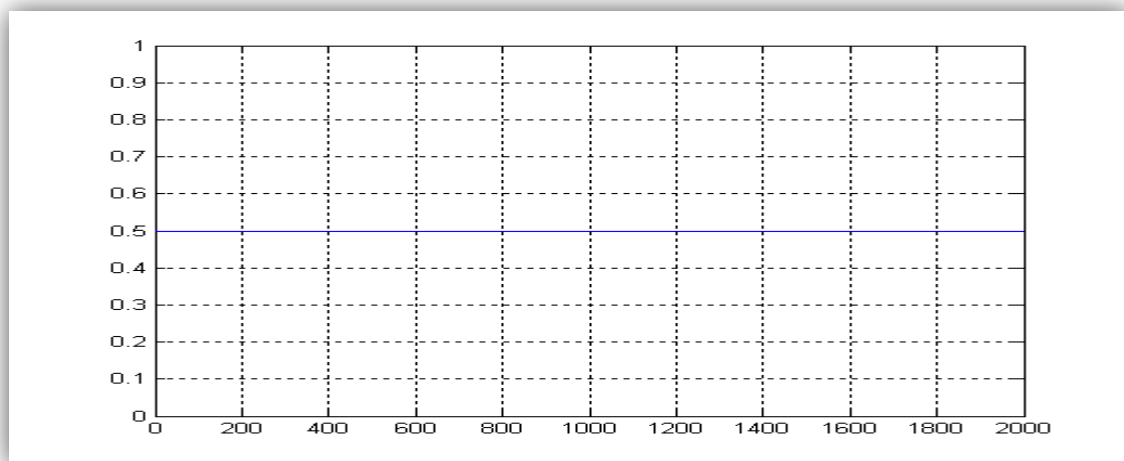


Figure IV.17 : L'entrée du système.

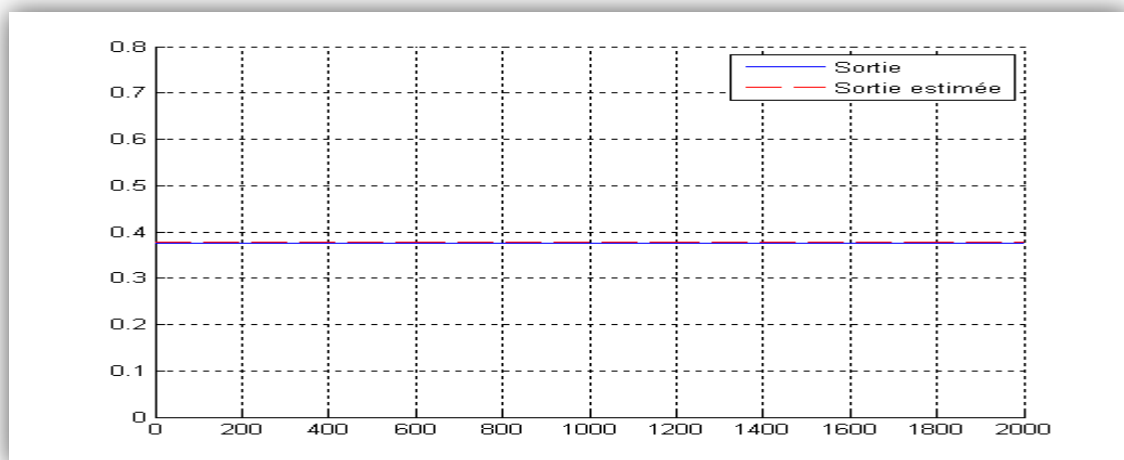


Figure IV.18 : La sortie et la sortie estimée.

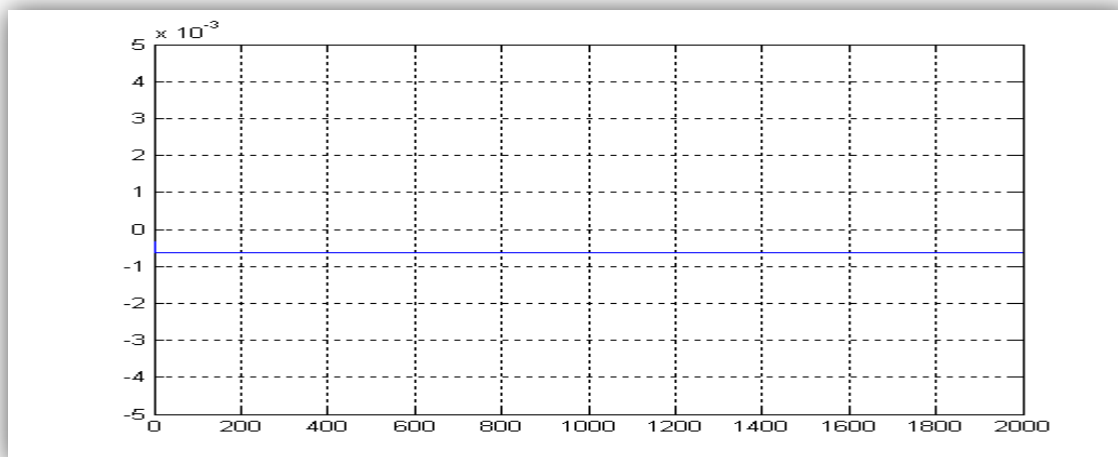


Figure IV.19 : Le résidu.

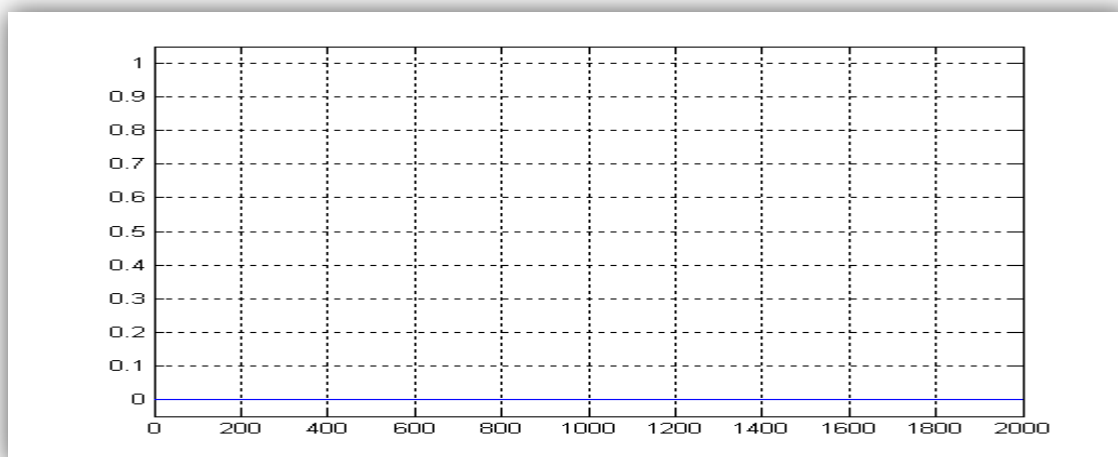


Figure IV.20 : La décision.

On voit clairement d'après les figures d'essai précédentes (IV, 17), (IV, 18), (IV, 19) et (IV, 20) que la sortie du système et la sortie du modèle sont presque identiques et le résidu et la décision restent nulles dans le cas d'absence de défauts.

Essai 3 :

<i>Emplacement du défaut</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
L'entrée du système	$k = 500$	50%

Maintenant on va simuler une perte d'efficacité d'amplitude de 50 % de l'entrée depuis l'instant $k = 500$ et les résultats obtenus sont illustrés respectivement sur les figures (IV, 21), (IV, 22), (IV, 23) et (IV, 24).

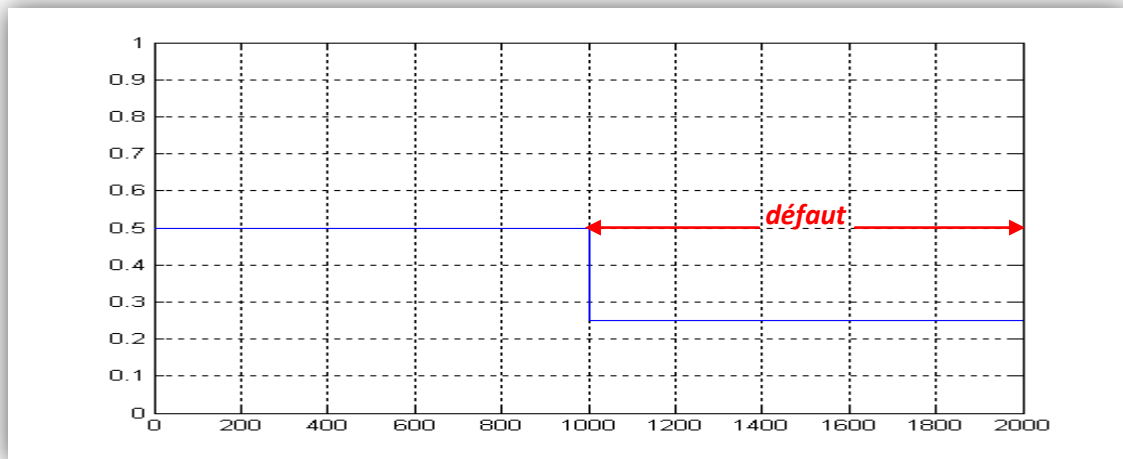


Figure IV.21 : L'entrée du système.

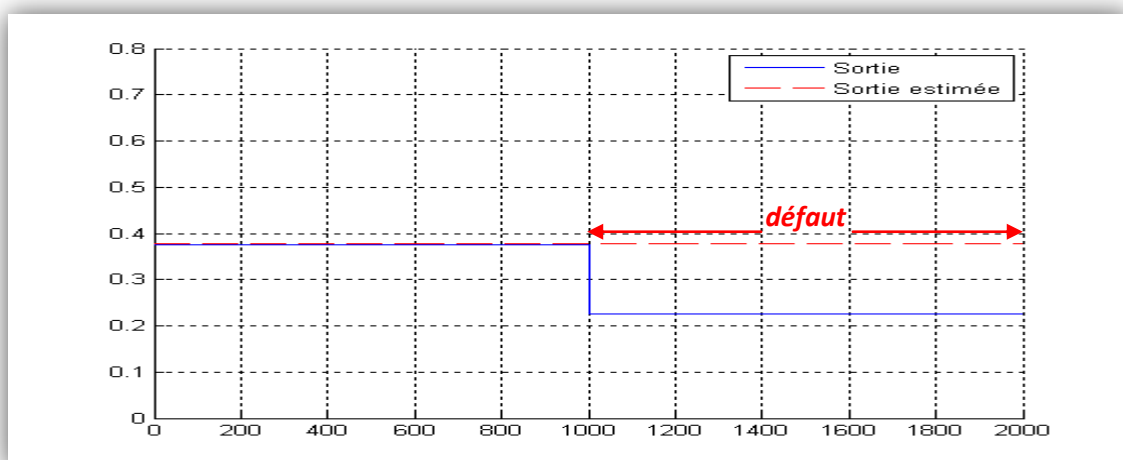


Figure IV.22 : La sortie et la sortie estimée.

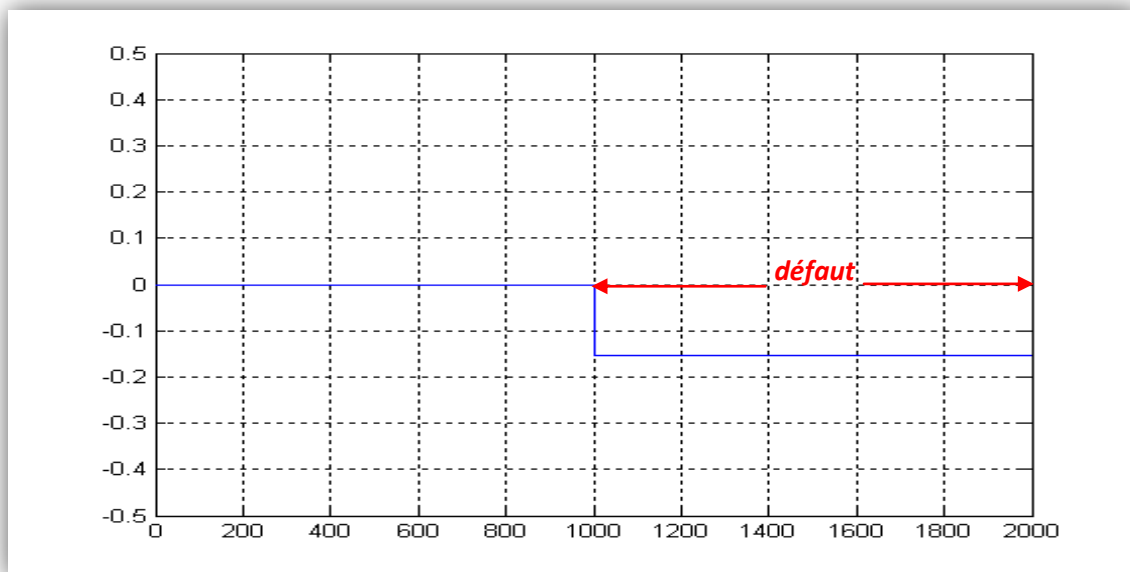


Figure IV.23 : Le résidu.

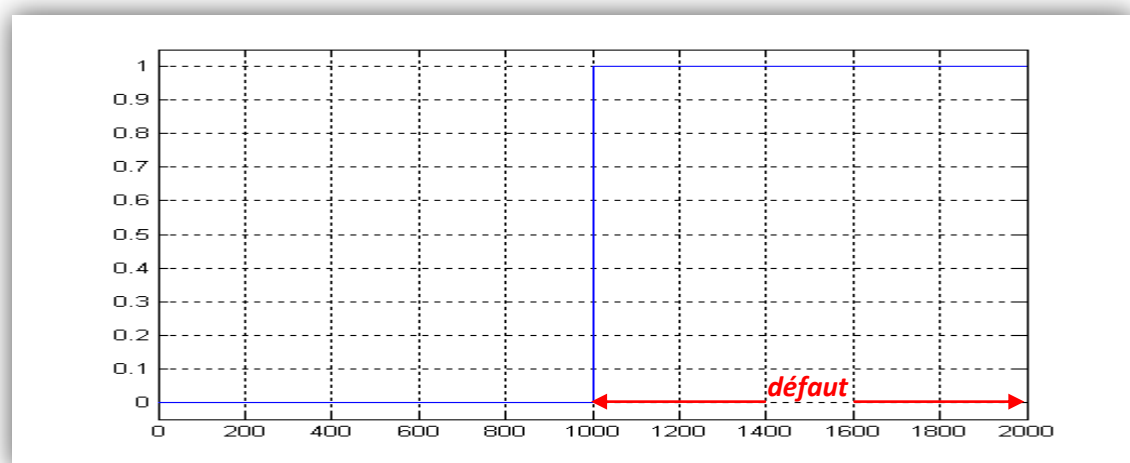


Figure IV.24 : La décision.

On voit d'après la figure (IV.23) que la sortie a une valeur constante jusqu'à l'instant $k = 1000$. Après cet instant, la valeur de la sortie prend une autre valeur à cause du défaut qui est appliqué sur l'entrée du système. La figure (IV,24) montre que la décision aussi prend la valeur 1 après cet instant.

Essai 4 :

<i>Emplacement du défaut</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
L'entrée du système	$k = 700$ jusqu'à $k = 1100$	90 %

Avec un défaut sur l'entrée, nous simulons une perte d'efficacité de 90 % entre l'instant $k = 700$ et $k = 1100$. Les résultats obtenus sont constatés sur les figures (IV.25) jusqu'à (IV.28).

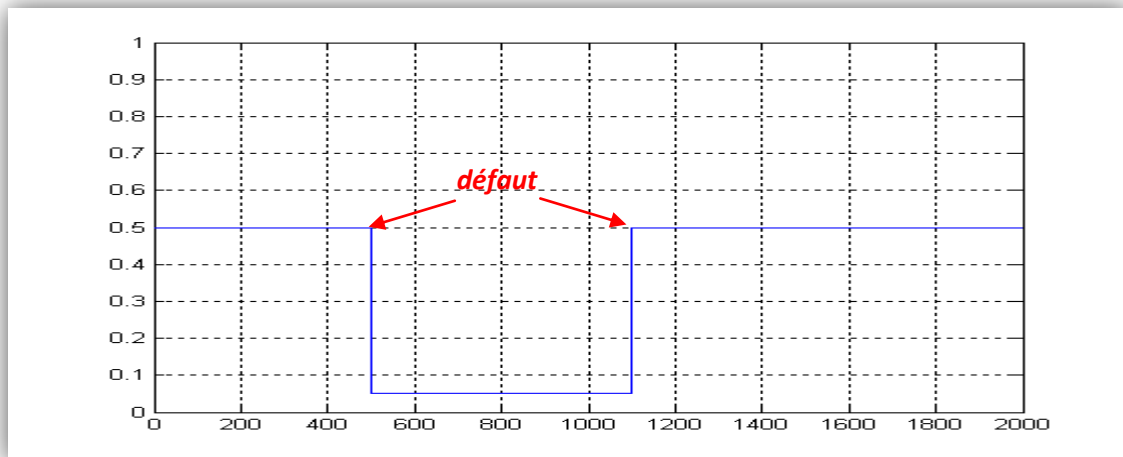


Figure IV.25 : L'entrée du système.

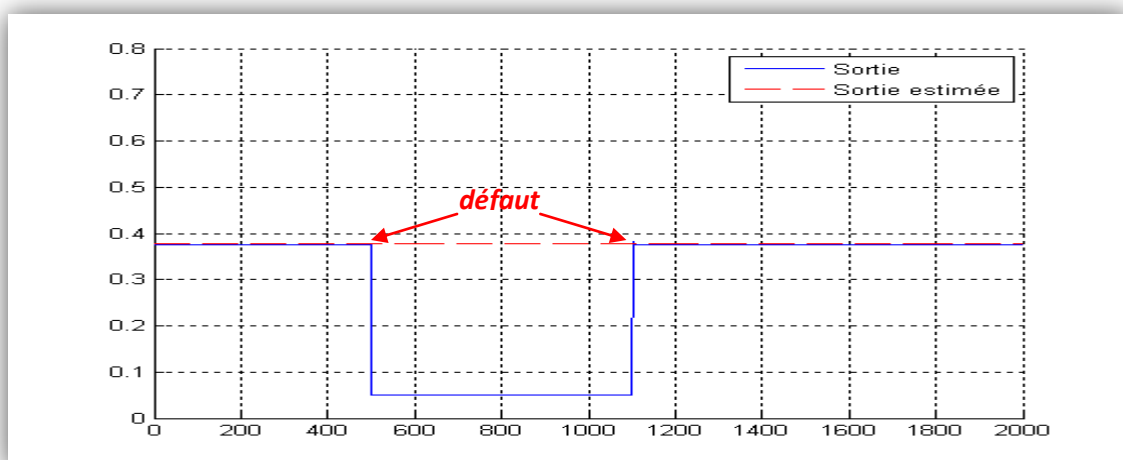


Figure IV.26 : La sortie et la sortie estimée.

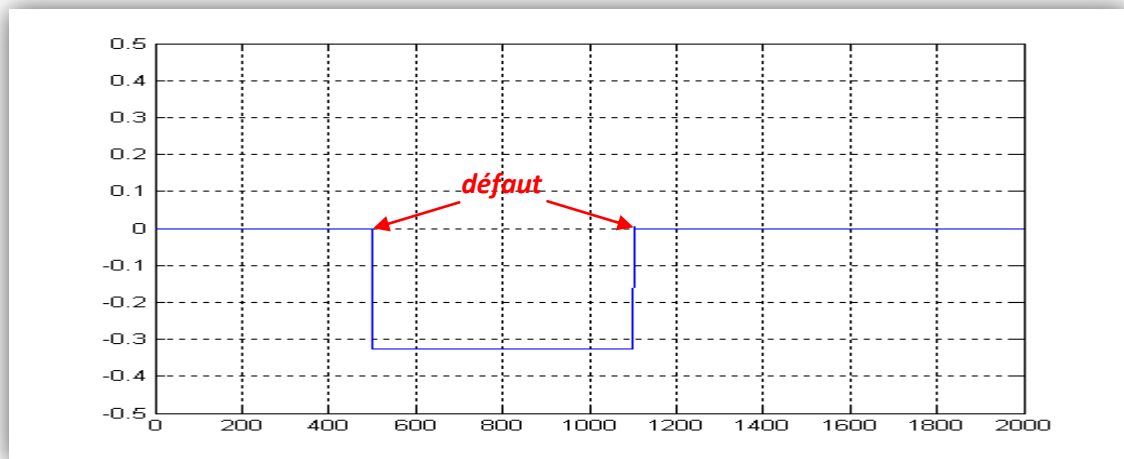


Figure IV.27 : Le résidu.

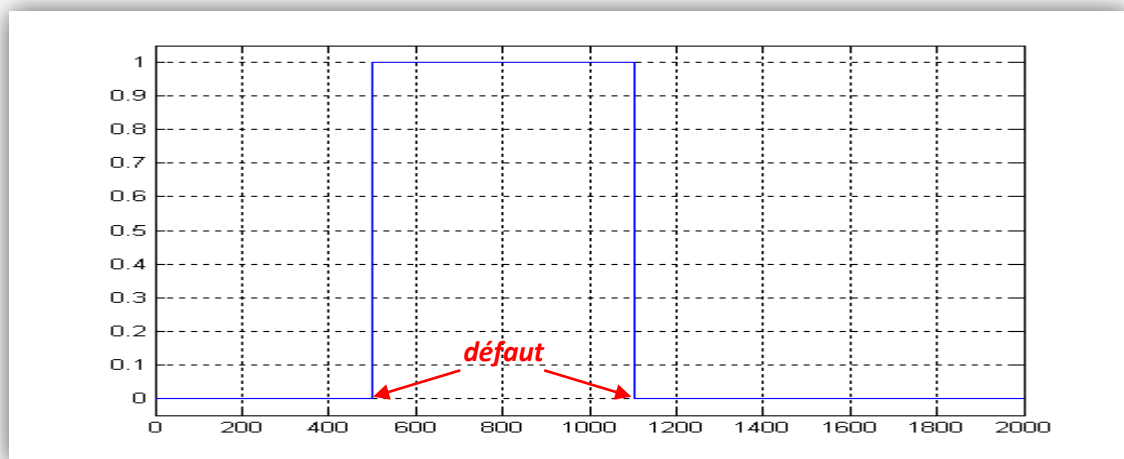


Figure IV.28 : La décision.

La figure(IV. 27) montre que la sortie change sa valeur entre l'instant $k = 500$ et l'instant $k = 1100$. Ce changement est dû à la présence de défaut sur l'entrée du système entre ces instants. La décision prend la valeur 1 entre ces instants comme il est illustré sur la figure (IV. 28).

Cet essai montre que le système de diagnostic à une bonne décision contre les défauts d'amplitude important.

Essai 5 :

<i>Emplacement du défaut</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
L'entrée du système	$k = 800$ jusqu'à $k = 1200$	10 %

Une autre perte d'efficacité est simulée sur l'entrée du système mais dans ce cas avec un défaut d'amplitude de 10 % entre l'instant $k = 800$ et $k = 1200$. Les résultats obtenus sont illustrés sur les figures (IV.29) jusqu'à (IV.32).

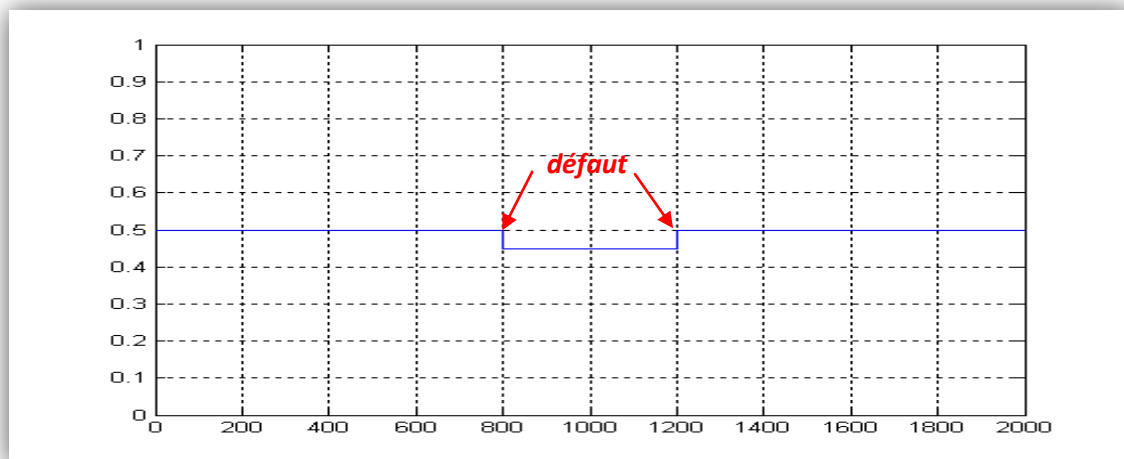


Figure IV.29 : L'entrée du système.

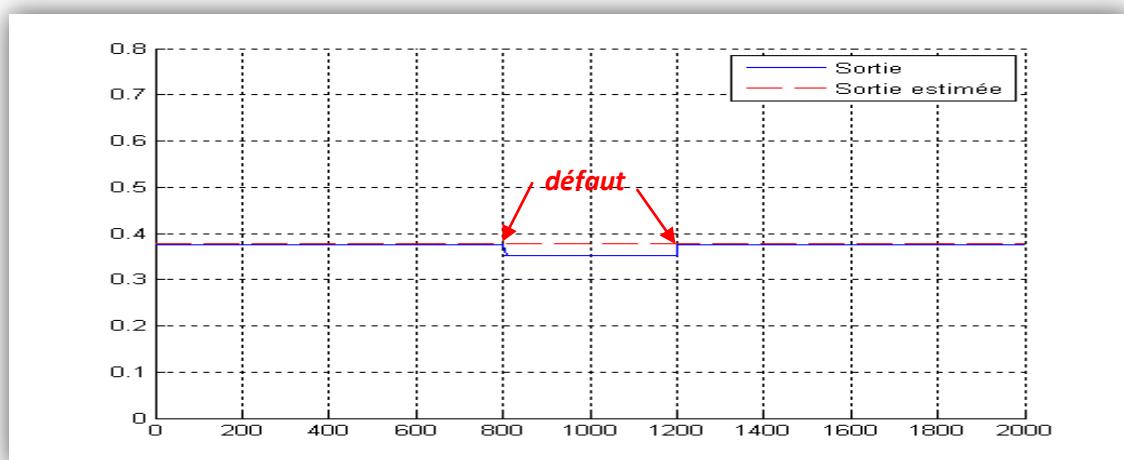


Figure IV.30 : La sortie et la sortie estimée.

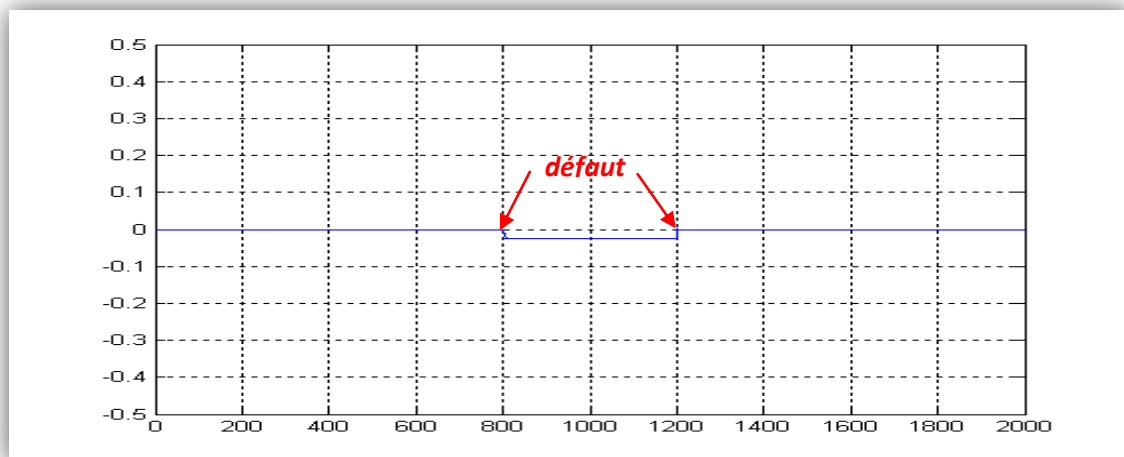


Figure IV.31 : Le résidu.

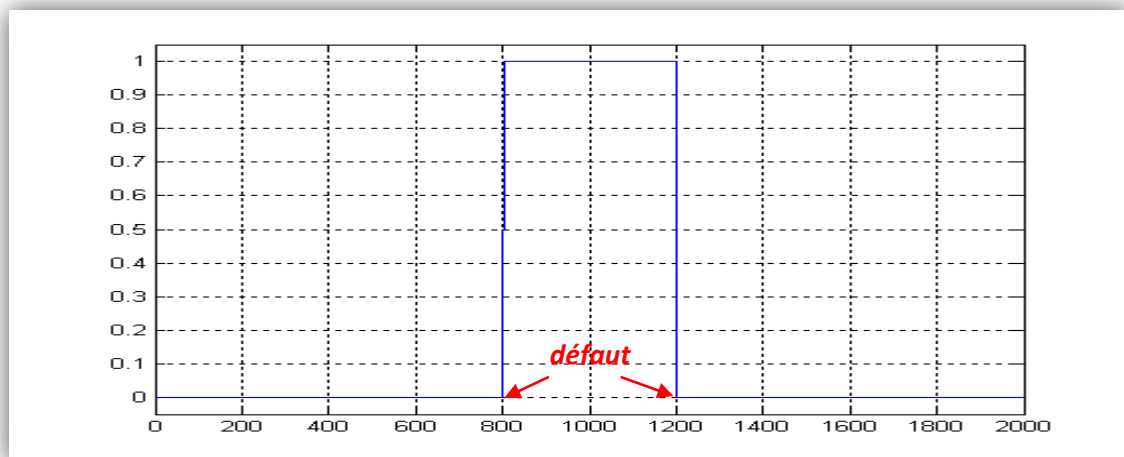


Figure IV.32 : La décision.

On voit d'après la figure (IV.31) que le résidu a une valeur non nulle entre les instants $k = 800$ et $k = 1200$. La figure (IV.32) aussi montre que la décision aussi prend la valeur 1 entre ces instants. Cet essai montre que le système de diagnostic a une bonne décision contre les défauts d'amplitude faible.

Essai 6 :

<i>Emplacement du défaut</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
L'entrée du système	$k = 200$ jusqu'à $k = 400$	10 %
L'entrée du système	$k = 800$ jusqu'à $k = 1600$	Dérive de [0,90] %

Maintenant, on va simuler deux défauts sur l'entrée du système. Le premier défaut est une perte d'amplitude d'environ de 10 % de la plage de variation entre les instants $k = 200$ et $k = 400$. Le deuxième défaut est une dérive avec une perte de 90 %. Ce défaut est simulé entre les instants $k = 800$ et $k = 1600$.

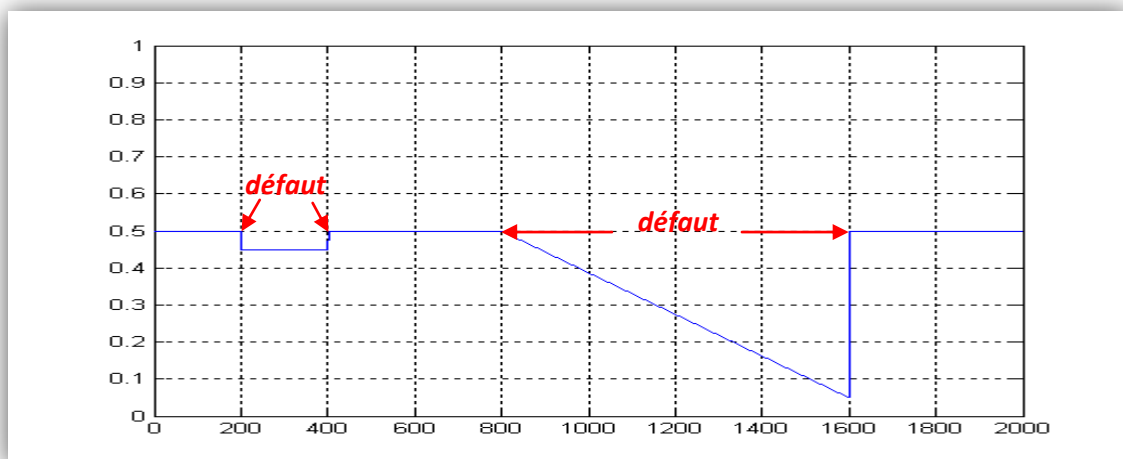


Figure IV.33 : L'entrée du système.

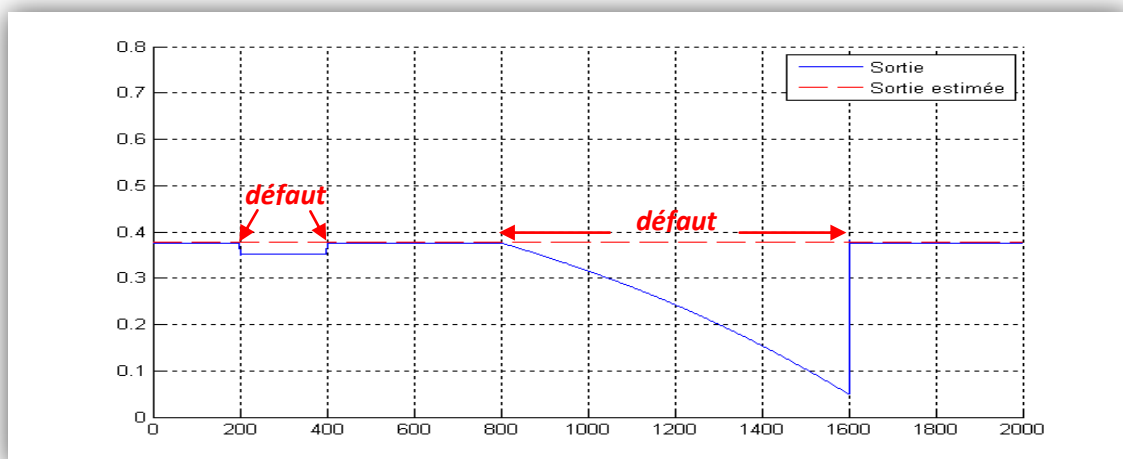


Figure IV.34 : La sortie et la sortie estimée.

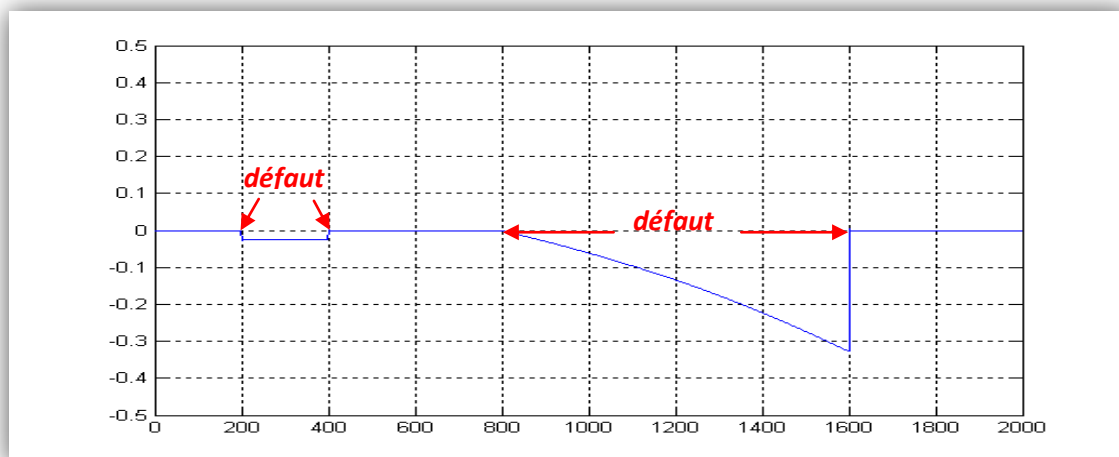


Figure IV.35 : Le résidu.

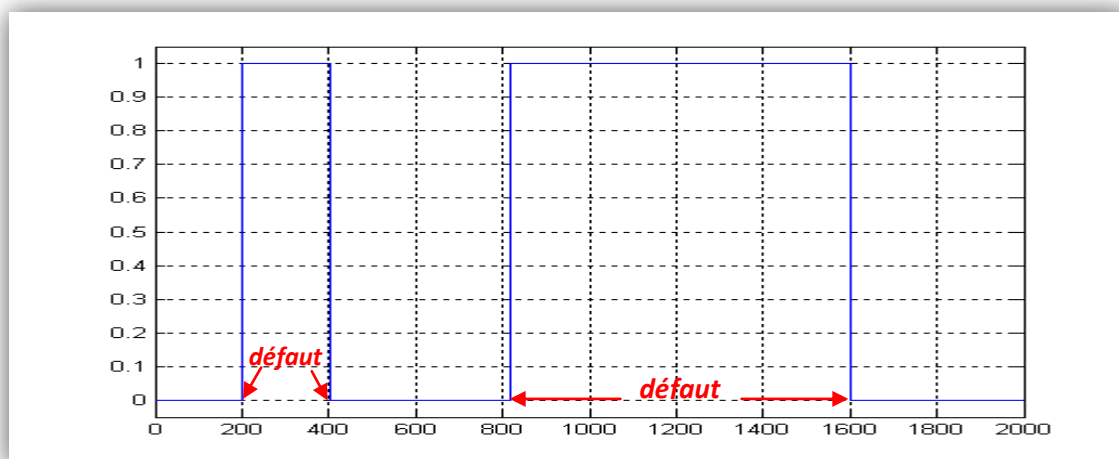


Figure IV.36 : La décision.

D'après la figure (IV.36), on voit que le système de décision marque deux défauts : le premier entre les instants $k = 200$ et $k = 400$. Le deuxième entre les instants $k = 800$ et $k = 1600$.

Donc, ce système de diagnostic donne des meilleurs résultats quelque soit le type et l'amplitude du défaut.

Essai 7 :

<i>Emplacement du défaut</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
L'entrée du système	$k = 1000$ jusqu'à $k = 1400$	15 %
L'entrée du système	$k = 1400$ jusqu'à $k = 1600$	75 %

Pour simuler le cas de défauts successifs, deux défauts d'amplitude 15 % et 75 % sont simulés sur l'entrée du système entre les instants $k = 1000$ et $k = 1400$ pour le premier et entre les instants $k = 1400$ et $k = 1600$ pour le deuxième.

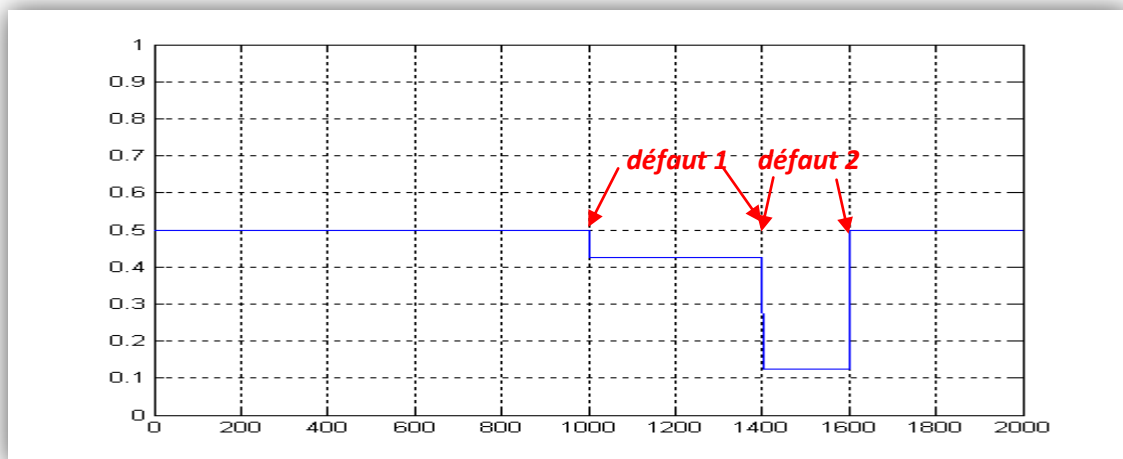


Figure IV.37 : L'entrée du système.

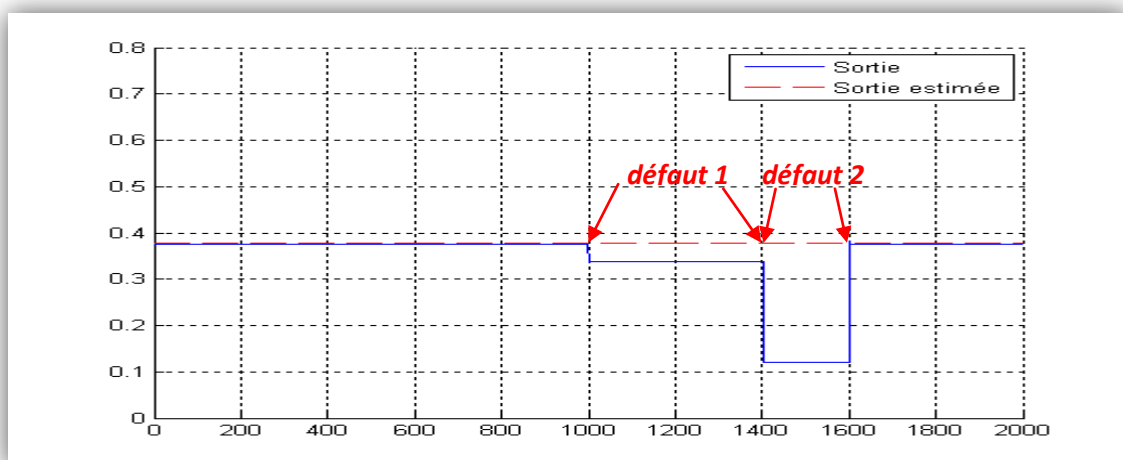


Figure IV.38 : La sortie et la sortie estimée.

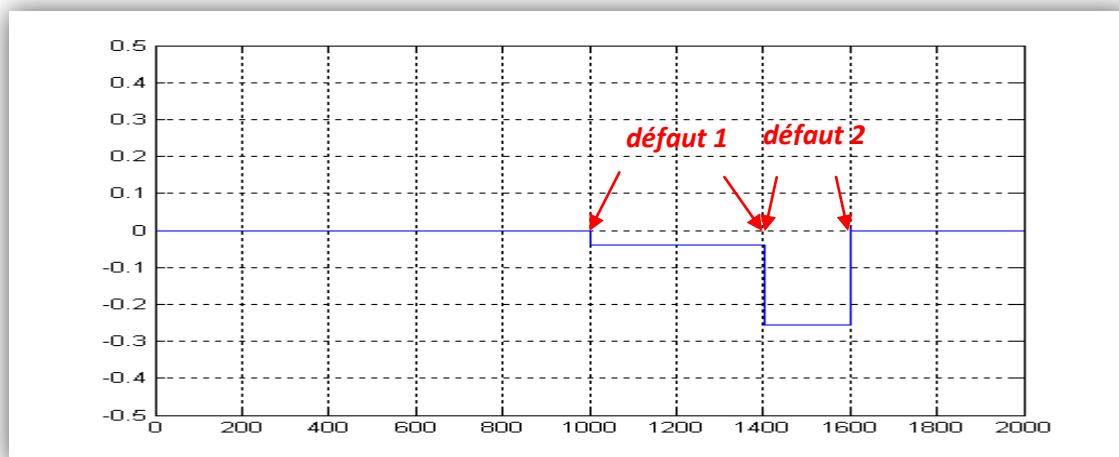


Figure IV.39 : Le résidu.

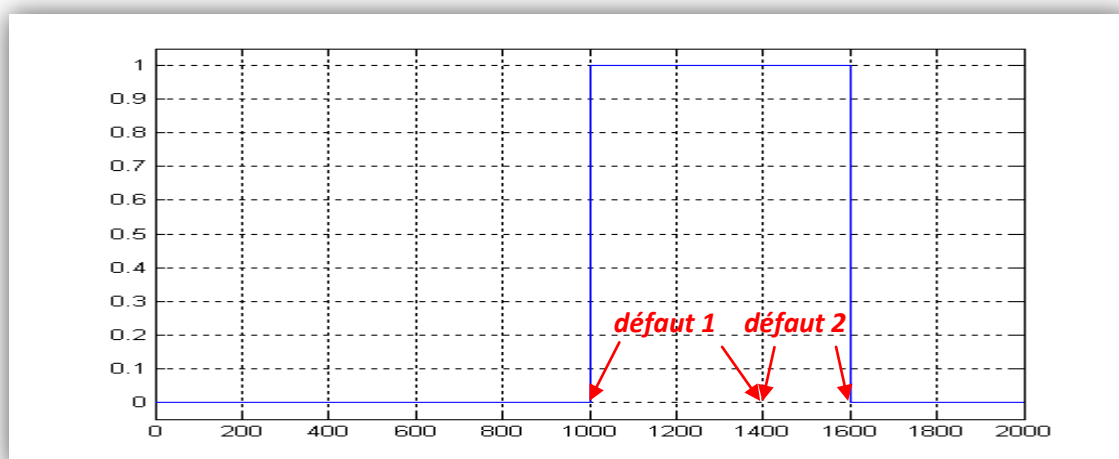


Figure IV.40 : La décision.

D'après la figure (IV.40), on voit que le système de décision marque une décision positive entre les instants $k = 1000$ et $k = 1600$.

Donc, ce système de diagnostic donne des meilleurs résultats dans le cas où des défauts provoquent l'entrée du système d'une façon successive.

Essai 8 :

<i>Emplacement du défaut</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
L'entrée du système	$k = 400$	60 %
L'entrée du système	$k = 800$ jusqu'à $k = 900$	50 %
L'entrée du système	$k = 1200$ jusqu'à $k = 1600$	Dérive de [0,40] %

Pour mieux vérifier notre système de diagnostic, nous simulons trois défauts de types différents. Le premier à l'instant $k = 400$ d'amplitude 60 %, le deuxième d'amplitude 50 % dans l'intervalle $k = [800,900]$ et le troisième est une dérive d'amplitude 40 % dans l'intervalle $k = [1200,1600]$.

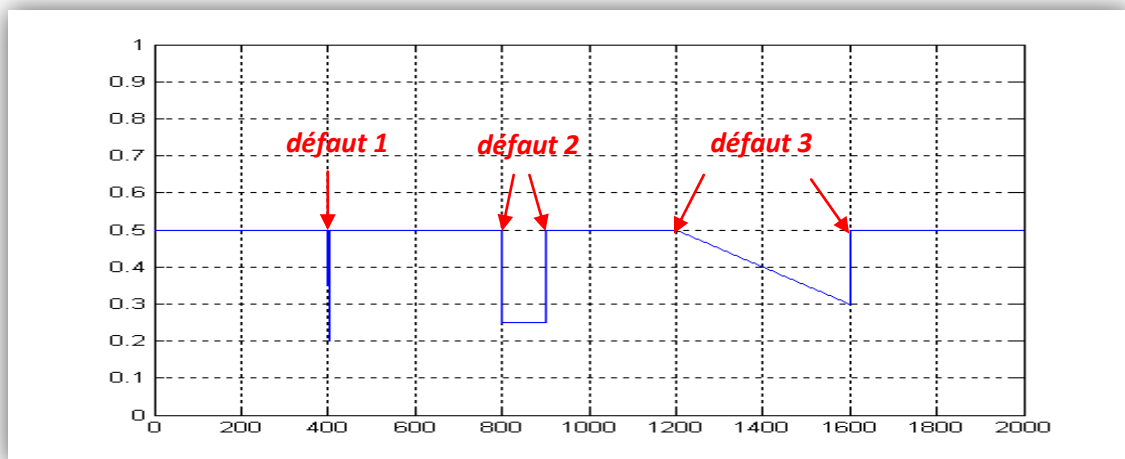


Figure IV.41 : L'entrée du système.

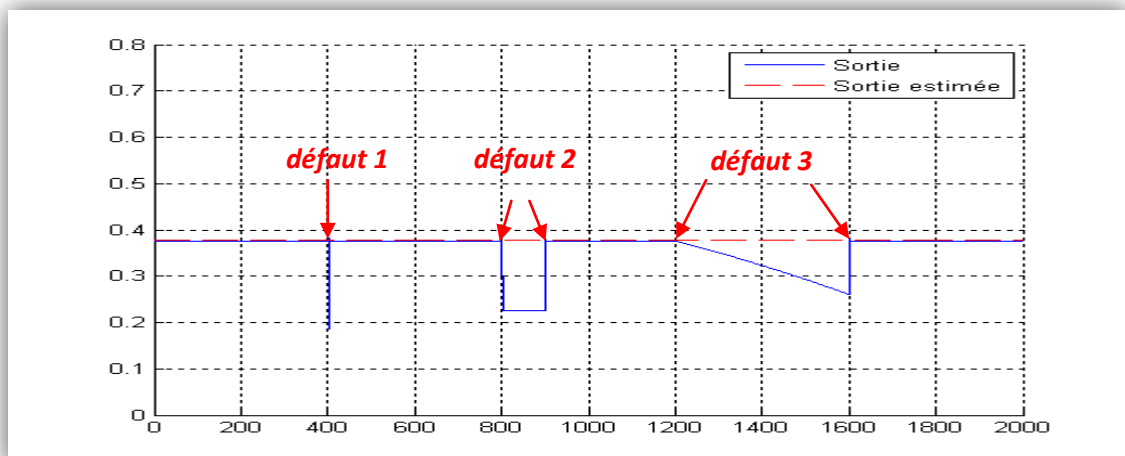


Figure IV.42 : La sortie et la sortie estimée.

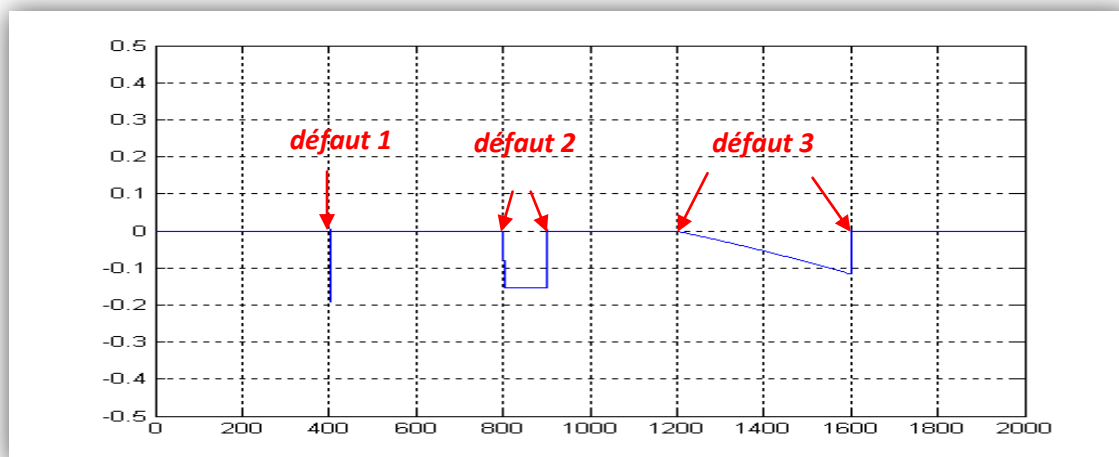


Figure IV.43 : Le résidu.

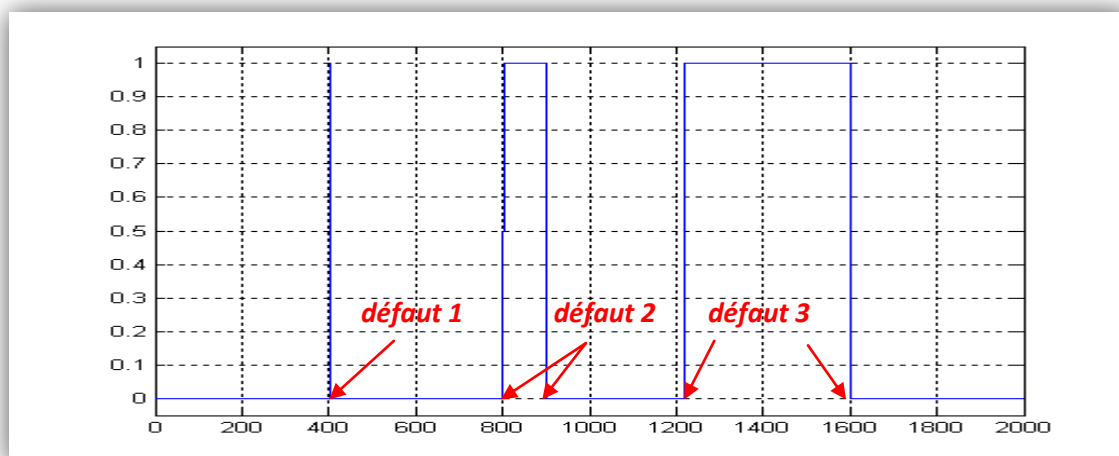


Figure IV.44 : La décision.

D'après la figure (IV, 44), on voit clairement que à l'instant $k = 400$ le système de décision donne un résultat positif dans le cas de présence des défauts de petite durée. Le système de décision signale deux défauts, un défaut entre les instants $k = 800$ et $k = 900$ et l'autre entre les instants $k = 1200$ et $k = 1600$. Cet essai montre aussi que le système de décision prend de bons résultats quelque soit le type, l'amplitude et la durée des défauts.

IV.4. COMMANDE TOLERANTE AUX DEFAUTS

Le diagnostic des défauts est nécessaire mais ne peut pas être suffisant. Donc, Il serait judicieux de tenir compte de la présence des défauts dès l'étape de conception de la loi de commande du système. Dans ce cadre, nous avons vu l'importance du diagnostic et de la tolérance aux défauts dans la conduite des systèmes automatisés dans l'objectif :

- ✓ d'améliorer la sécurité des personnes et des systèmes.
- ✓ de mieux gérer les procédures de maintenance.
- ✓ d'améliorer la qualité et le rendement de la production.

Depuis plusieurs années, le problème de la tolérance aux défauts a été abordé sous plusieurs points de vue. La tolérance aux défauts devient d'une importance capitale afin d'éviter des conséquences catastrophiques et de préserver la sécurité des personnes et des systèmes. L'objectif principal des méthodes de tolérance aux défauts consiste à préserver au mieux les performances nominales du système : la stabilité, la précision et la rapidité.

Généralement, la commande tolérante aux défauts peut être classée suivant deux approches distinctes : une approche de commande passive et une approche active. L'approche passive est caractérisée par une loi de commande robuste vis à vis des défaillances et une même commande en absence ou en présence de défauts et les régulateurs sont synthétisés de façon être robuste à certains défauts. Cette approche ne nécessite aucun schéma de détection de défauts ni aucune reconfiguration de loi de commande, mais sa capacité de tolérance reste restreinte à quelques défauts spécifiques. En revanche, l'approche active est caractérisée par un schéma de détection de défauts et une stratégie de changement de la loi de commande tout en préservant la stabilité et les performances du système.

Deux stratégies sont présentées pour l'approche active qui sont l'accommodation des défauts et la reconfiguration du système. L'accommodation est possible quand l'étape FDI permet aussi d'estimer le défaut (nécessité d'un modèle de fonctionnement défaillant). Si le défaut ne peut être estimé, on parle de la reconfiguration du système (utilisation d'un sous-ensemble de composants, remplacement d'éléments défaillants).

L'Accommodation de défaut prend la forme d'une reconfiguration de commande, une fois que le diagnostic observé est activé après la déclaration de défaut. La reconfiguration de commande prend la forme d'un signal additif au contrôleur de modèle du fonctionnement nominale.

La figure (IV.45) représente un schéma général de diagnostic et d'accommodation de défauts.

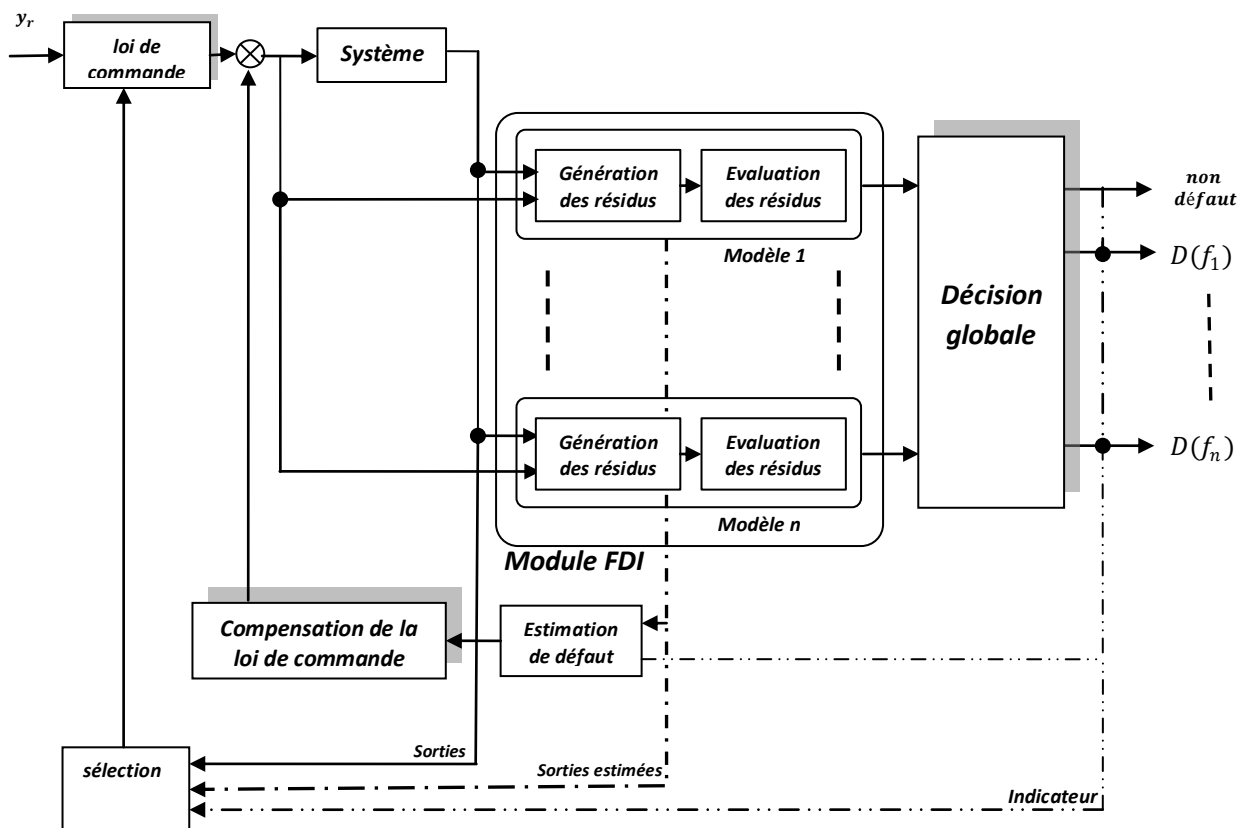


Figure IV.45 : Schéma général de diagnostic et d'accommodation de défauts.

Une approche de commande tolérante effectuée dans le cadre de l'accommodation de défauts en ligne basée sur un module de diagnostic de défauts. Ce module utilise un banc de contrôleurs permet de générer des signaux de compensation de la loi de commande.

Les relations entrées/sorties entre la loi de commande et le système à contrôler restent inchangées lorsque l'on parle d'accommodation de défauts. Donc, il s'agit de sélectionner la loi de commande adéquate à partir d'un ensemble de régulateurs pré-calculés hors ligne. Essentiellement, ces régulateurs peuvent avoir n'importe quelle structure, soit neuronale, floue ou une structure neuro-floue.

Selon la phase de processus identifiée par l'étape de diagnostic, un sélecteur de mode flou sert à sélectionner le mode de commande approprié, qui peut comprendre à la fois l'adaptation des ensembles des paramètres de commande sans changements structurels, ou la commutation de différentes situations spécifiques des structures de contrôleurs. La commutation entre les contrôleurs est plus utile si les exigences de performance doivent être modifiées.

La figure (IV.46) montre un schéma général d'une commande tolérante aux défauts utilisant les réseaux de neurones et la logique floue.

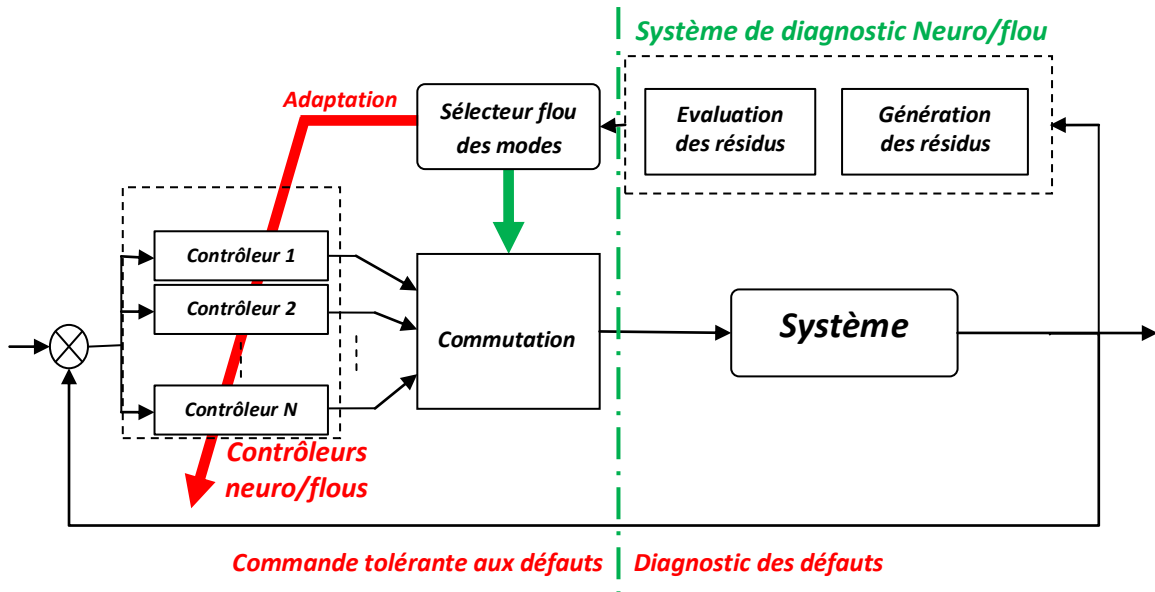


Figure IV.46 : Schéma général d'une commande tolérante aux défauts utilisant les réseaux de neurones et la logique floue.

IV.5. CONCLUSION

Dans ce chapitre, nous avons présenté, dans un premier temps, les différents aspects liés à la représentation des systèmes dynamiques non linéaires et les différentes structures de représentations permettant leurs implantations ont également été présentés.

Aussi nous avons donné les différentes méthodes de génération et d'évaluations des résidus utilisées pour la détection et l'isolation des défauts qui peuvent apparaître dans un procédé industriel. Ces méthodes sont basées sur l'utilisation de réseaux de neurones et de la logique floue. Ensuite, nous validons ces méthodes par des essais sur un exemple de diagnostic.

Enfin, nous avons montré que l'accommodation des défauts se fait en sélectionnant des lois de commande neuro/floues pré-calculées, ceci nous a amené à introduire une commande tolérante aux défauts capable de compenser l'effet des défauts par l'ajout d'un terme de compensation. Selon la considération d'un modèle dynamique non linéaire, ces techniques seront validées en simulation dans le chapitre suivant.

Chapitre V

Application de la commande tolérante aux défauts

- V.1. Introduction*
 - V.2. Commande tolérante aux défauts*
 - V.3. Description et modélisation*
 - V.4. Modélisation du système*
 - V.5. Résultats de simulation*
 - V.6. Conclusion*
-

V.1. INTRODUCTION

En vue de tester l'efficacité de commande tolérante aux défauts présentée au quatrième chapitre, une simulation de cette dernière a été faite sur un processus hydraulique/thermique constitué de trois cuves.

L'objectif de ce chapitre est de synthétiser une loi de commande active tolérante aux défauts actionneurs pour des systèmes représentés par des réseaux de neurones et des systèmes d'inférence floue. La synthèse d'une commande tolérante aux défauts active que nous proposons de développer ici, consiste à s'appuyer sur un module de diagnostic qui pourvoit à l'estimation du ou des défaut(s) actionneur(s) considéré(s). Cette estimation du défaut actionneur doit permettre l'annulation et le masquage des défauts qui apparaissent ce qui représente un des principes de la commande tolérante aux défauts active.

Dans un premier temps une modélisation du système est présentée et ensuite les résultats de simulation seront exposés et discutés. Les résultats numériques qu'on va présenter dans cette section sont issus d'une simulation réalisée en Matlab/Simulink.

V.2. DESCRIPTION ET MODELISATION

1. Description du système

Dans ce qui suit, nous présentons un procédé hydraulique/thermique, comme il est représenté par la figure (V.1). Ce système est composé de trois cuves cylindriques de section S identiques. Les coefficients de viscosité μ_1, μ_2 et les sections des tuyaux S_1, S_2 sont aussi identiques. La troisième cuve sert à mélanger les liquides. Deux pompes, constituées par des moteurs à courant continu, sont utilisées pour commander les débits Q_1, Q_2 affectant les niveaux sur chaque cuve.

Le système est composé de deux actionneurs et quatre capteurs. Les mesures sont la température T_3 de troisième cuve et les niveaux de liquide (h_1, h_2, h_3) dans chaque cuve. Les cuves sont supposés parfaitement calorifugées et leurs capacités thermiques sont négligeables. L'objectif est de conserver un volume de liquide constant à une température désirée dans la troisième cuve.

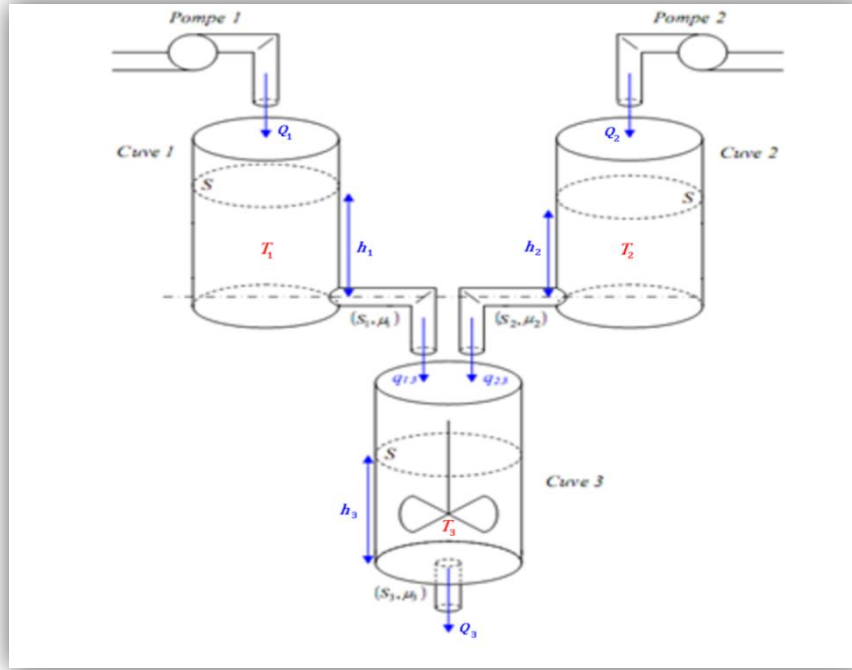


Figure V.1 : Diagramme du système de trois cuves.

2. Modélisation du système

En utilisant les équations de la conservation du volume pour obtenir le modèle mathématique du système. Ce système peut être représenté de la façon suivante :

$$\begin{cases} S \frac{dh_1}{dt} = Q_1 - Q_{13} \\ S \frac{dh_2}{dt} = Q_2 - Q_{23} \\ S \frac{dh_3}{dt} = Q_{13} + Q_{23} - Q_3 \end{cases} \quad \text{avec} \quad \begin{cases} Q_{13} = \alpha_1 \cdot \sqrt{h_1} \\ Q_{23} = \alpha_2 \cdot \sqrt{h_2} \\ Q_3 = \alpha_3 \cdot \sqrt{h_3} \end{cases} \quad (\text{V.1})$$

Nous obtenons les trois équations différentielles non linéaires suivantes, tels que les coefficients $\alpha_i = \mu_i S_i \sqrt{2g}$ avec $i = 1,2,3$.

$$\begin{cases} \dot{h}_1 = \frac{1}{S} (Q_1 - \alpha_1 \cdot \sqrt{h_1}) \\ \dot{h}_2 = \frac{1}{S} (Q_2 - \alpha_2 \cdot \sqrt{h_2}) \\ \dot{h}_3 = \frac{1}{S} (\alpha_1 \cdot \sqrt{h_1} + \alpha_2 \cdot \sqrt{h_2} - \alpha_3 \cdot \sqrt{h_3}) \end{cases} \quad (\text{V.2})$$

Si l'on utilise l'équation de bilan calorimétrique nous obtenons :

$$\frac{d(S \cdot h_3 \cdot T_3)}{dt} = Q_{13} \cdot T_1 + Q_{23} \cdot T_2 - Q_3 \cdot T_3 \quad (\text{V.3})$$

On a aussi :

$$\frac{d(S \cdot h_3 \cdot T_3)}{dt} = S \cdot \left[\frac{dh_3}{dt} \cdot T_3 + \frac{dT_3}{dt} \cdot h_3 \right] \quad (\text{V.4})$$

À partir des équations (IV.3) et (IV.4) on obtient :

$$S \cdot \left[\frac{dh_3}{dt} \cdot T_3 + \frac{dT_3}{dt} \cdot h_3 \right] = Q_{13} \cdot T_1 + Q_{23} \cdot T_2 - Q_3 \cdot T_3 \quad (\text{V.5})$$

Remplaçons $\frac{dh_3}{dt}$:

$$S \cdot \left[(Q_{13} + Q_{23}) \cdot T_3 + \frac{dT_3}{dt} \cdot h_3 \right] = Q_{13} \cdot T_1 + Q_{23} \cdot T_2 \quad (\text{V.6})$$

Nous obtenons l'équation différentielle suivante :

$$\dot{T}_3 = - \frac{1}{S \cdot h_3} [Q_{13} \cdot (T_3 - T_1) + Q_{23} \cdot (T_3 - T_2)] \quad (\text{V.7})$$

Finalement, ce système peut être représenté par les quatre équations non linéaires suivantes :

$$\begin{cases} \dot{h}_1 = \frac{1}{S} (Q_1 - \alpha_1 \cdot \sqrt{h_1}) & = f(Q_1, h_1) \\ \dot{h}_2 = \frac{1}{S} (Q_2 - \alpha_2 \cdot \sqrt{h_2}) & = f(Q_2, h_2) \\ \dot{h}_3 = \frac{1}{S} (\alpha_1 \cdot \sqrt{h_1} + \alpha_2 \cdot \sqrt{h_2} - \alpha_3 \cdot \sqrt{h_3}) & = f(h_1, h_2, h_3) \\ \dot{T}_3 = - \frac{1}{S \cdot h_3} [Q_{13} \cdot (T_3 - T_1) + Q_{23} \cdot (T_3 - T_2)] & = f(h_1, h_2, h_3, T_3) \end{cases} \quad (\text{V.8})$$

V.3. MODELISATION DU SYSTEME

Pour identifier notre modèle non linéaire, on a opté pour un modèle basé sur la structure NARX. Le choix de cette structure est venu après une comparaison entre les différentes structures non linéaires (NOE, NARMAX, NFIR).

On a remarqué aussi que la modélisation des systèmes non linéaires à l'aide de la technique neuro-floue donne des meilleurs résultats par rapport aux techniques floues ou les techniques basées sur les réseaux de neurones. Donc, on a opté dans la suite de ce travail sur l'architecture basée sur la structure NARX et la technique neuro-floue, car la valeur de l'erreur quadratique est la plus petite.

Suivant le fonctionnement et l'architecture du système, nous avons choisi un modèle composé de quatre réseaux ANFIS, dont les quatre équations suivantes représentent les différents entrées et sorties :

$$h_1(k) = F_1(Q_1(k-1), h_1(k-1)) \quad (\text{V.9})$$

$$h_2(k) = F_2(Q_2(k-1), h_2(k-1)) \quad (\text{V.10})$$

$$h_3(k) = F_3(h_1(k-1), h_2(k-1), h_3(k-1)) \quad (\text{V.11})$$

$$T_3(k) = F_4(h_1(k-1), h_2(k-1), h_3(k-1), T_3(k-1)) \quad (\text{V.12})$$

On a fait plusieurs essais pour obtenir l'architecture qui simule le fonctionnement du système. Ces expériences prennent beaucoup de temps, tel que le choix du nombre des ensembles flous et le type des fonctions d'appartenance. Les sorties estimées convergent vers les sorties réelles, au bout de trois itérations seulement. L'architecture de ces modèles est représentée comme suit :

- ✓ Le premier réseau du modèle neuro-flou simule la sortie h_1 . Il prend deux entrées et une sortie : les valeurs précédentes de l'entrée Q_1 et de la sortie h_1 comme entrées et les valeurs actuelles de h_1 comme sortie. Chaque entrée est représentée par quatre ensembles flous dont la forme des fonctions d'appartenance est gaussienne.
 - ✓ Le deuxième réseau du modèle neuro-flou simule la sortie h_2 . Il prend deux entrées et une sortie : les valeurs précédentes de l'entrée Q_2 et de la sortie h_2 comme entrées et les valeurs actuelles de h_2 comme sortie. Chaque entrée est représentée par quatre ensembles flous dont la forme des fonctions d'appartenance est gaussienne.
 - ✓ Le troisième réseau du modèle neuro-flou simule la sortie h_3 . Il prend trois entrées et une sortie : les valeurs précédentes des sorties h_1, h_2 et h_3 comme entrées et la valeur actuelle de h_3 comme sortie. Chaque entrée est représentée par quatre ensembles flous dont la forme des fonctions d'appartenance est gaussienne.
 - ✓ Le quatrième réseau du modèle neuro-flou simule la sortie T_3 . Il prend quatre entrées et une sortie : les valeurs précédentes des sorties h_1, h_2, h_3 et T_3 comme entrées et les valeurs actuelles de T_3 comme sortie. Chaque entrée est représentée par trois ensembles flous dont la forme des fonctions d'appartenance est gaussienne.
-

V.4. RESULTATS DE SIMULATION

En vue de voir l'efficacité de commande tolérante aux défauts présenté au quatrième chapitre, un certain nombre de tests à été fait en simulant certains cas de défauts. Pour le système hydraulique/thermique, les défauts simulés sont des défauts qui affectent les deux pompes. Donc, on ne considérera ici que les défauts actionneurs. La simulation des défauts actionneurs n'affecte pas la commandabilité ni l'observabilité du système [34].

Dans un premier temps, nous avons réalisé une expérimentation (en l'absence de défaut) pour tester la fiabilité du système : nous avons choisi deux fonctions de commandes qui peuvent représenter toutes les cas possibles des entrées. Nous pouvons voir les résultats sur les figures (V.2) jusqu'à (V.7). Ces figures montrent l'évolution des sorties et les erreurs de modélisations correspondants en respect des changements des commandes u_1, u_2 .

N.B: La simulation du système a été réalisée avec un pas de simulation $T_e = 10 \text{ msec}$. Donc, toutes les figures sont en fonction du temps discret $k = t / T_e$.

Essai 1 :

- La première fonction d'entrée du système est donnée par :

$$u_{k,1} = 8 \cdot 10^{-6} \cdot [\sin(k/320) + 0,5 \cdot \sin(k/160) + \dots + 0,5 \cdot \sin(k/640) + 0,3 \cdot \sin(k/60)] + 1,5 \cdot 10^{-5} \quad (\text{V. 13})$$

- La deuxième fonction du système est donnée par :

$$u_{k,2} = 8 \cdot 10^{-6} \cdot [\cos(k/320) + 0,5 \cdot \sin(k/160) + \dots + 0,5 \cdot \cos(k/640) + 0,3 \cdot \sin(k/60)] + 1,35 \cdot 10^{-5} \quad (\text{V. 14})$$

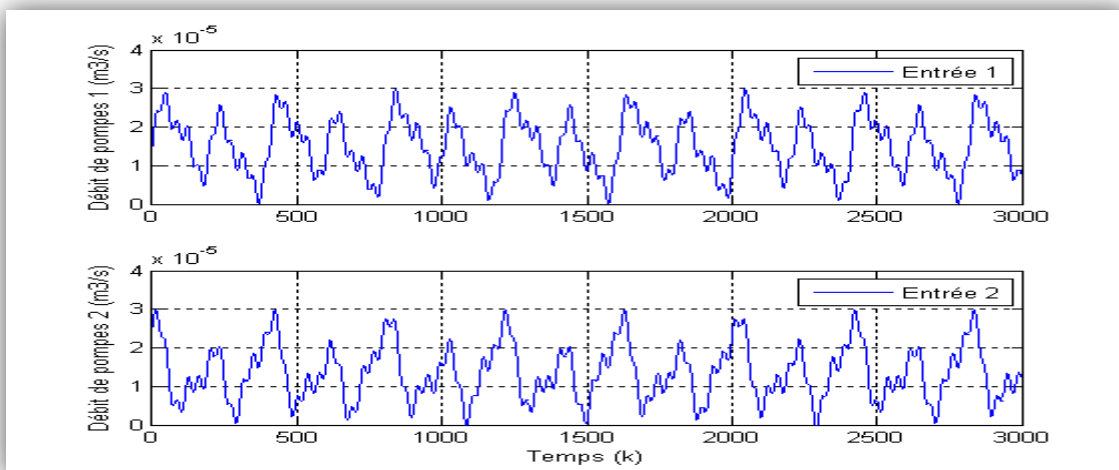


Figure V.2 : Les entrées du système.

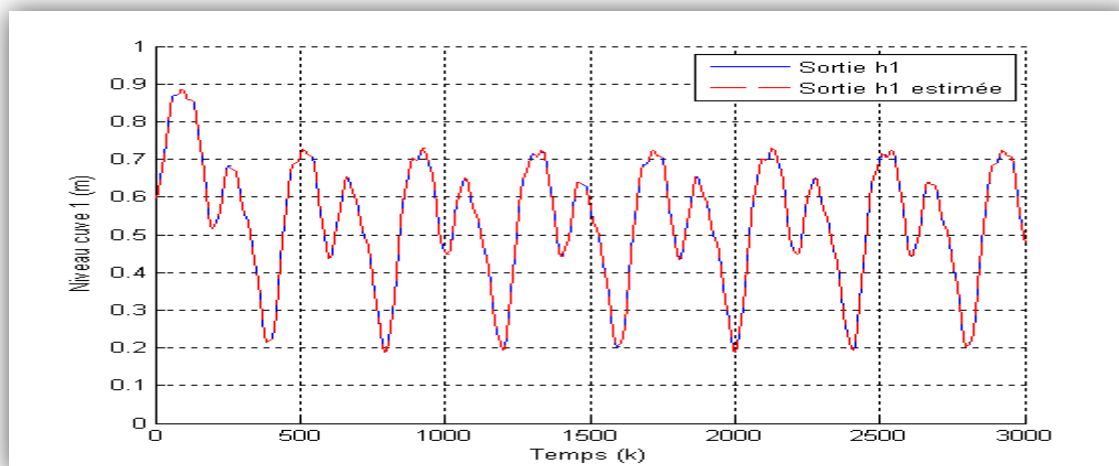


Figure V.3 : La sortie h_1 et la sortie h_1 estimée.

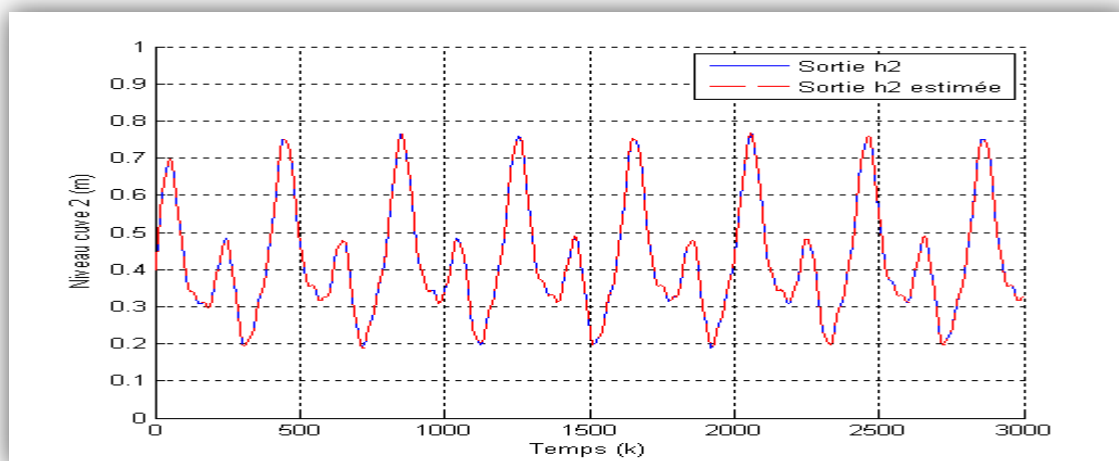


Figure V.4 : La sortie h_2 et la sortie h_2 estimée.

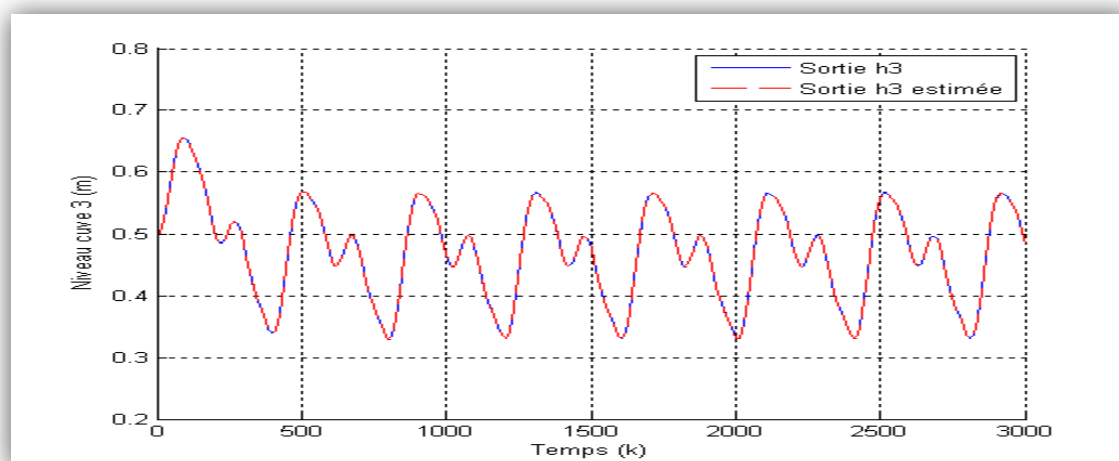


Figure V.5 : La sortie h_3 et la sortie h_3 estimée.

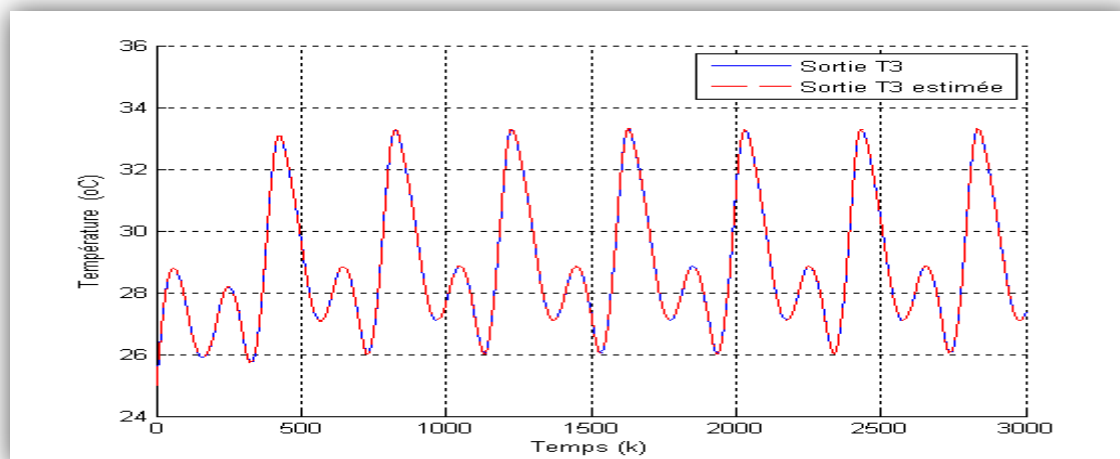


Figure V.6 : La sortie T_3 et la sortie T_3 estimée.

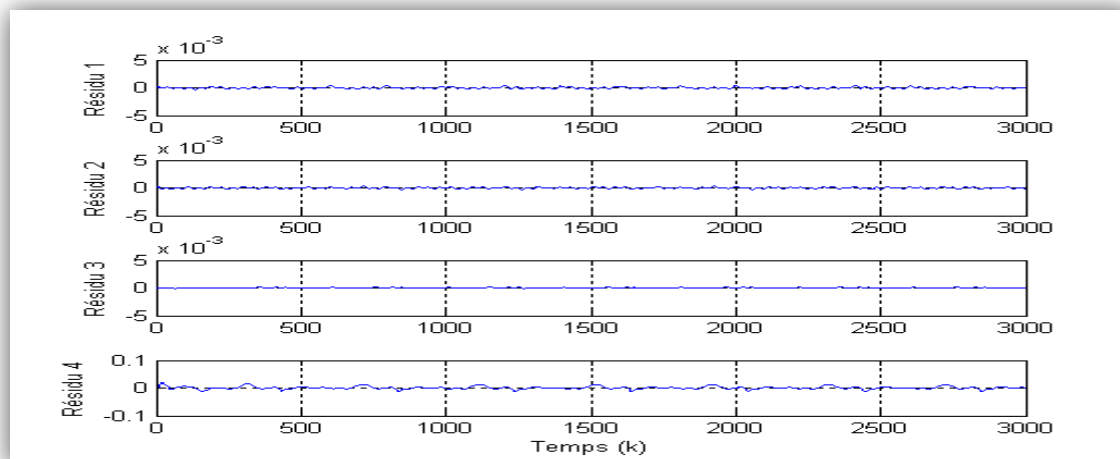


Figure V.7 : Erreurs de modélisation (résidus dans le cas sans défaut).

Après l'étape de modélisation c'est-à-dire après l'étape de la génération des résidus ; il faut passer à la deuxième étape qui représente l'étape de l'évaluation des résidus. Ces derniers sont utilisés pour diagnostiquer les défauts lesquels existent dans le système et pour donner des informations pertinentes pour fait le diagnostic.

Nous avons proposé un modèle de raisonnement flou, pour classer ces défauts à partir de ces informations. On a choisi pour chaque résidu trois fonctions d'appartenance : deux fonctions de type trapézoïdale et une de type triangulaire. Pour le choix de leurs paramètres, on a effectué de nombreux essais en présence de plusieurs défauts différents. La sortie de l'un de ces modèles

de décision flou prend soit la valeur 0 dans le cas d'absence de défaut ou la valeur 1 dans le cas de présence de défaut.

Afin de montrer l'efficacité du modèle neuro-flou proposé, nous avons étudié deux situations : Dans la première situation nous montrons tout d'abord l'efficacité du système de diagnostic (sans reconfiguration), ensuite nous présentons la stratégie de commande tolérante aux défauts proposée.

Les résultats de simulation sont donnés par des expérimentations effectuées sur le procédé hydraulique/thermique et qui sont montrées sur les figures ci-dessous. Notons que ces simulations sont effectuées en supposant des débits des pompes constantes. Dans ce cas, l'objectif est de conserver un volume de liquide de la troisième cuve constant à une température désirée.

Les débits des pompes :

$$[Q_1, Q_2]^T = [1,5.10^{-5}, 1,5.10^{-5}]^T m^3/s \quad (V.15)$$

La température désirée :

$$T_3 = 30 C^o \quad (V.16)$$

L'hauteur de liquide dans la troisième cuve :

$$h_3 = 0.5 m. \quad (V.17)$$

Les fonctions d'appartenance de résidus 1 et 2 sont données comme suit :

▪ **Résidu 1 :**

$$\begin{aligned} N_1 &= [-1 \quad -1 \quad -1,01.10^{-4} \quad -9,9.10^{-5}] \\ Z_1 &= [-10^{-4}, 10^{-4}] \\ P_1 &= [9,9.10^{-5} \quad 1,01.10^{-4} \quad 1 \quad 1] \end{aligned} \quad (V.18)$$

▪ **Résidu 2 :**

$$\begin{aligned} N_2 &= [-1 \quad -1 \quad -1,01.10^{-4} \quad -9,9.10^{-5}] \\ Z_2 &= [-10^{-4}, 10^{-4}] \\ P_2 &= [9,9.10^{-5} \quad 1,01.10^{-4} \quad 1 \quad 1] \end{aligned} \quad (V.19)$$

1. La 1^{ère} situation : Diagnostic aux défauts.

Premièrement, nous allons tester et vérifier l'efficacité et la fiabilité de système de diagnostic. Pour cela, nous allons simuler des différents types de défauts (biais, valeur intermittente, dérive) dans différents temps (défauts simultanés, défauts successifs) au niveau des actionneurs.

Essai 2 : En absence de défauts.

Tout d'abord dans cet essai nous allons présenter les entrées et les sorties du système nominal, les sorties du modèle, les résidus et les décisions dans le cas d'absence de défaut. Les résultats sont illustrés sur les figures (V.8) jusqu'à (V.14) respectivement.

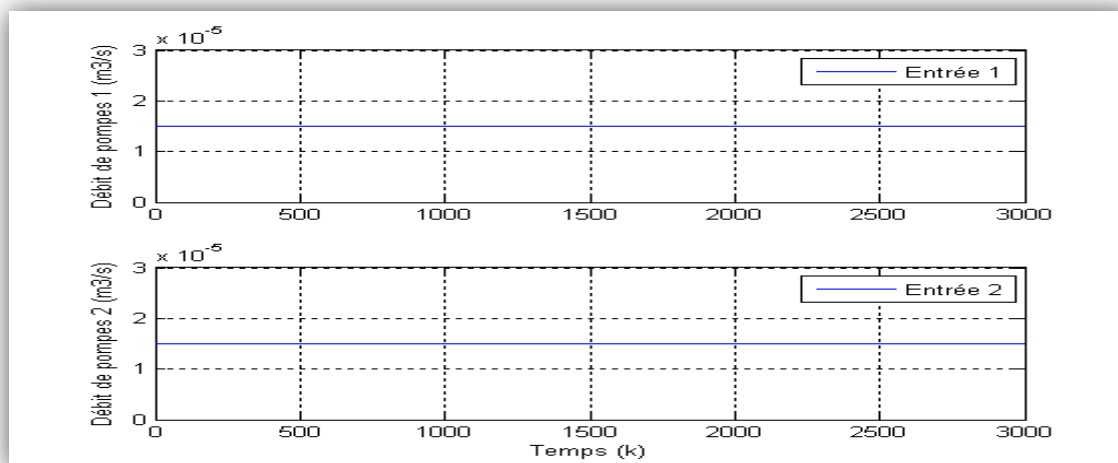


Figure V.8 : Les entrées du système.

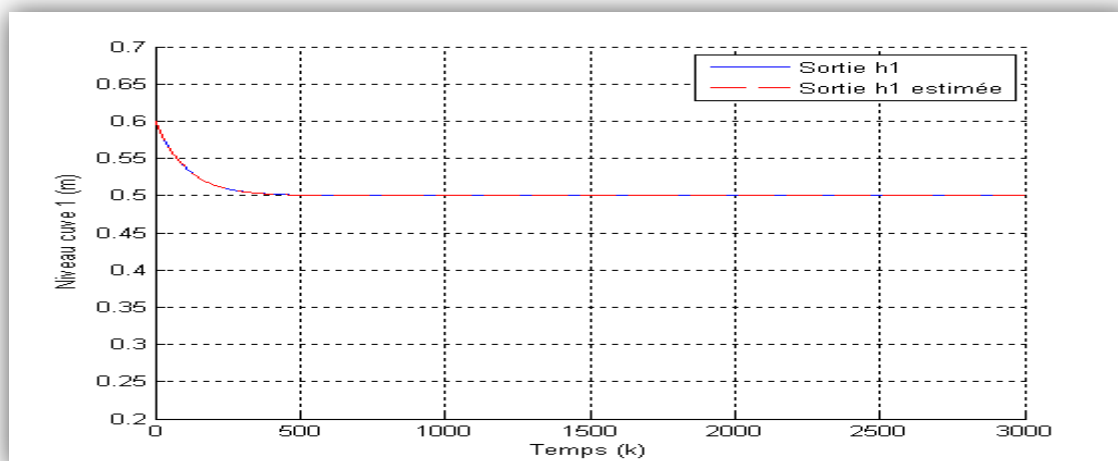


Figure V.9 : La sortie h_1 et la sortie h_1 estimée.

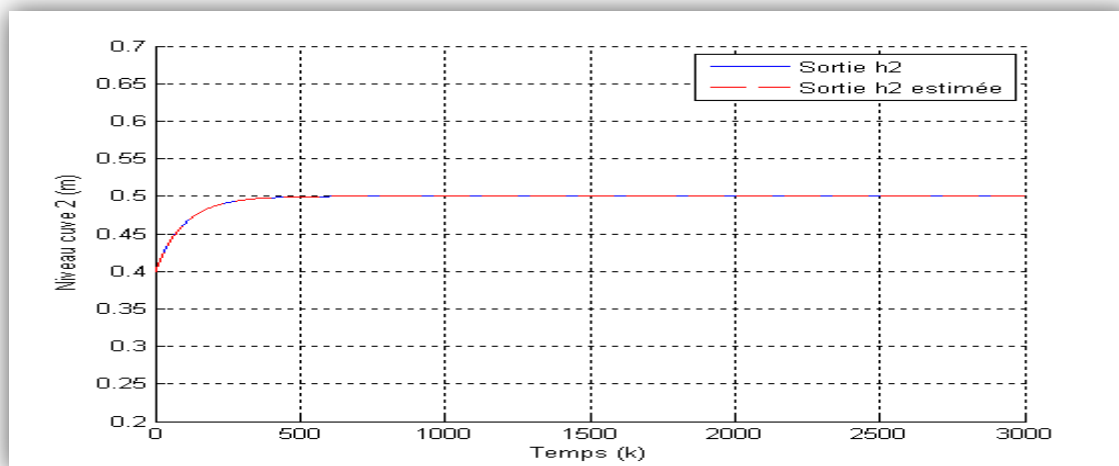


Figure V.10 : La sortie h_2 et la sortie h_2 estimée.

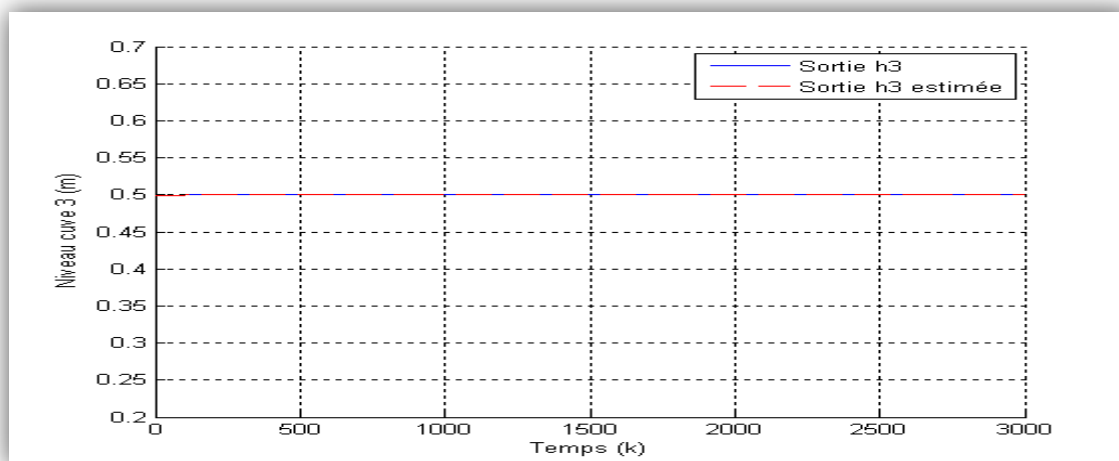


Figure V.11 : La sortie h_3 et la sortie h_3 estimée.

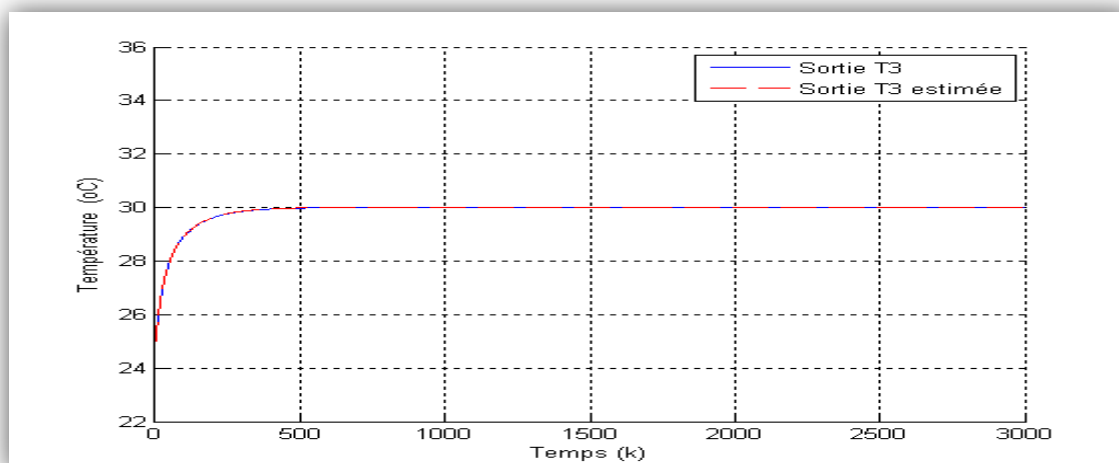


Figure V.12 : La sortie T_3 et la sortie T_3 estimée.

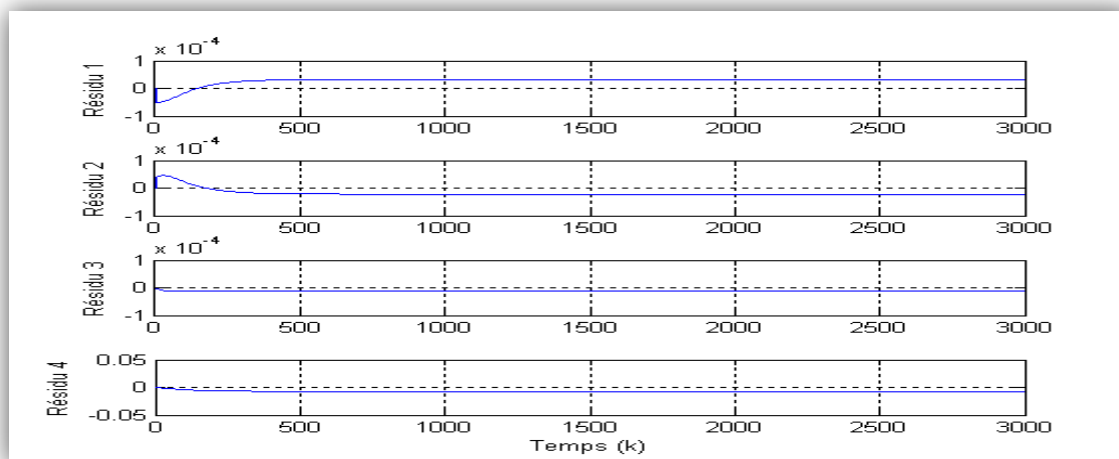


Figure V.13 : Les résidus.

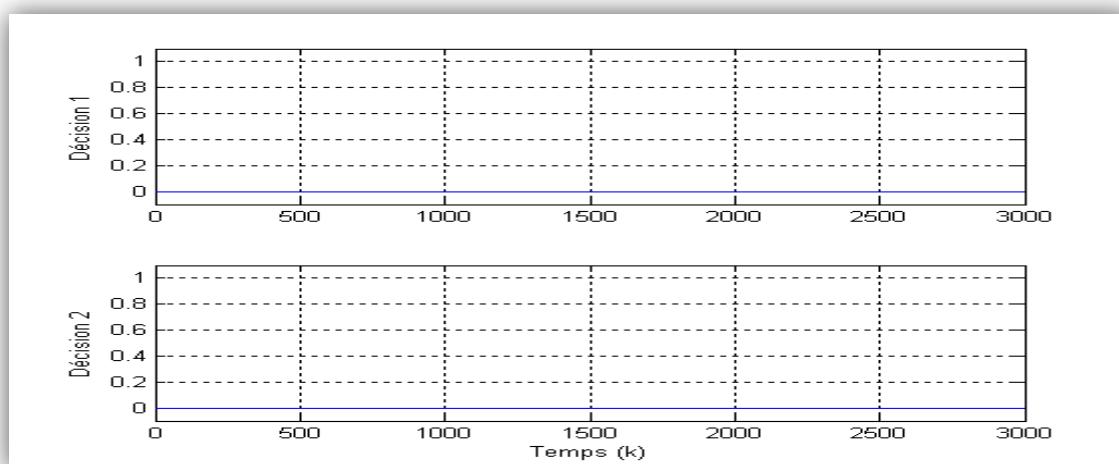


Figure V.14 : Les décisions.

On voit d'après les figures précédentes que l'objectif de cette simulation est vérifié pendant toute la durée de la simulation : L'hauteur de liquide de la troisième cuve est égale 0.5 m à une température désirée égale 30 C°. Il est clair d'après la figure (V.14), que les sorties de décision restent nulles entre les deux instants.

Essai 3: en présence de défauts.

Actionneur affecté	Instant de défaut	Perte de l'efficacité
Actionneur N° 1	$k = 1000$ jusqu'à $k = 2000$	50%

Une perte d'efficacité de 50 % est simulée avec un défaut sur l'entrée de premier actionneur à l'instant $k = 1000$ jusqu'à $k = 2000$, ensuite nous constatons les résultats sur les sorties du système, les résidus associés et les sorties des modèles de décision.

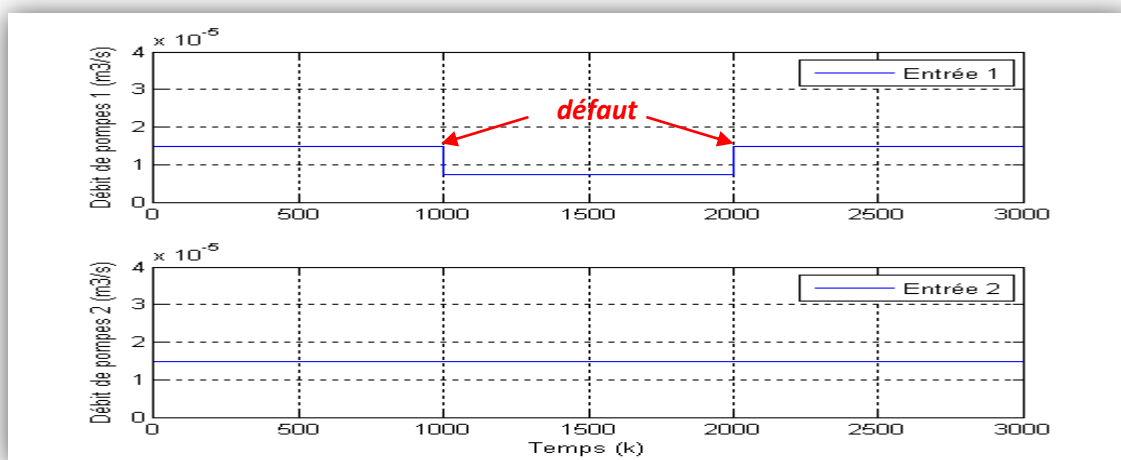


Figure V.15 : Les entrées du système.

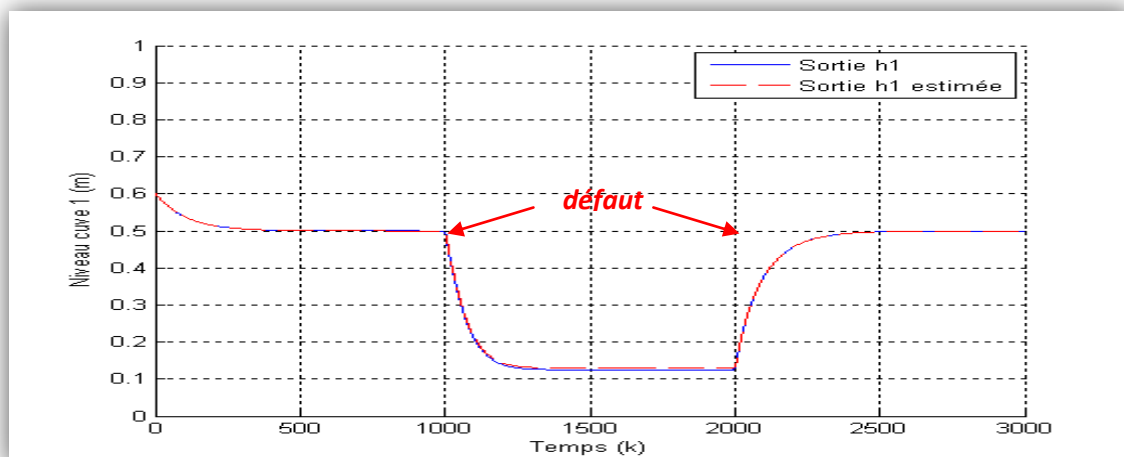


Figure V.16 : La sortie h_1 et la sortie h_1 estimée.

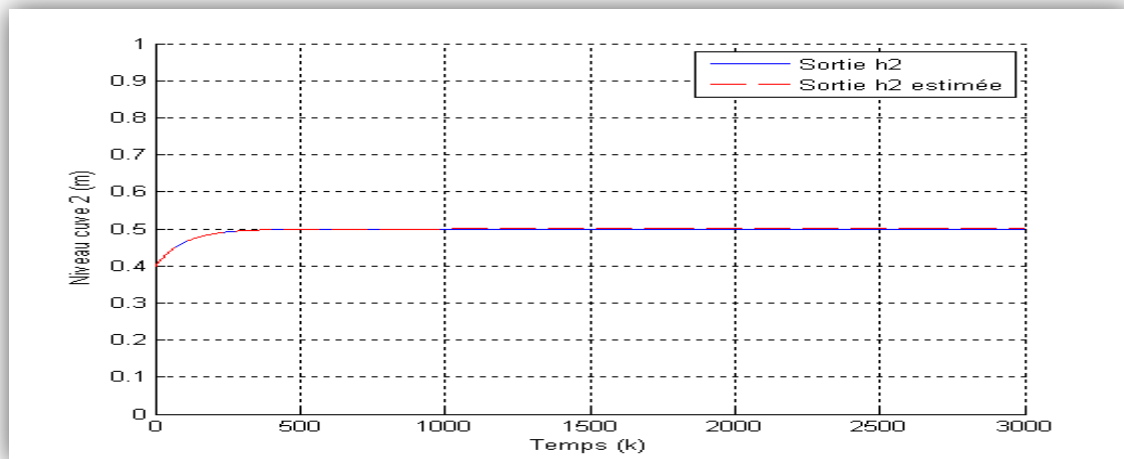


Figure V.17 : La sortie h_2 et la sortie h_2 estimée.

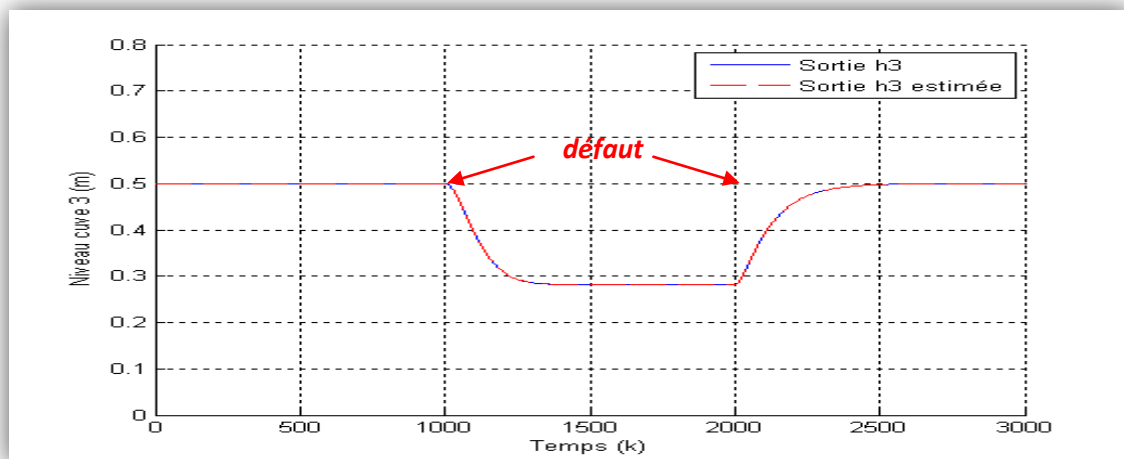


Figure V.18 : La sortie h_3 et la sortie h_3 estimée.

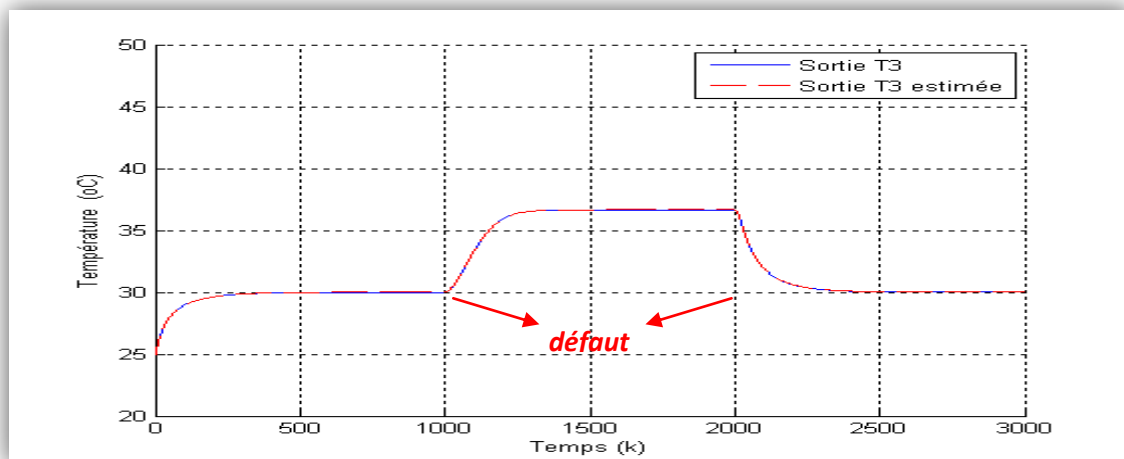


Figure V.19 : La sortie T_3 et la sortie T_3 estimée.

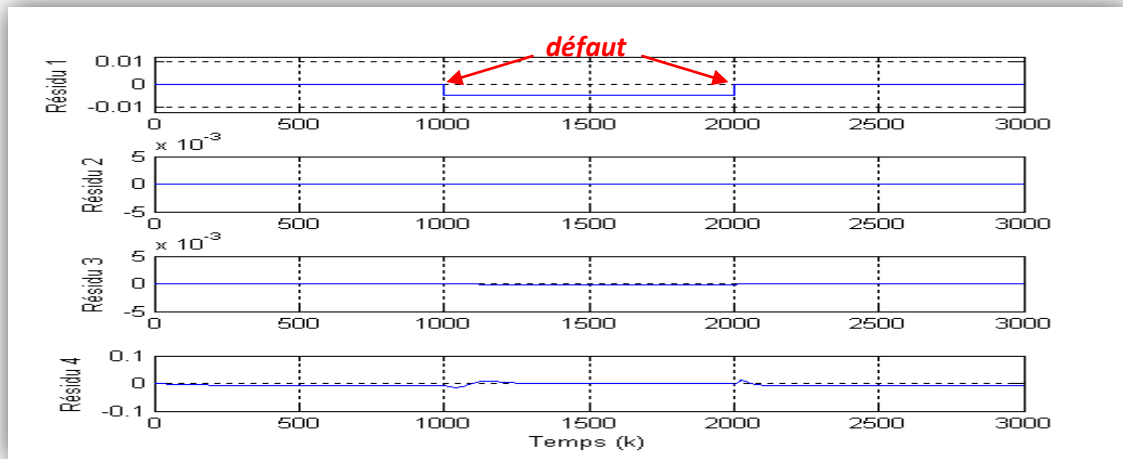


Figure V.20 : Les résidus.

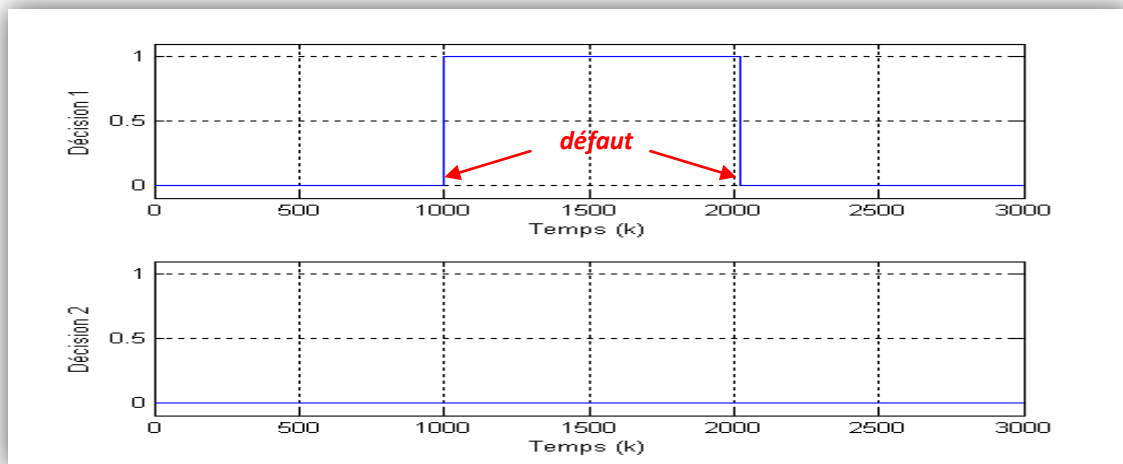


Figure V.21 : Les décisions.

La figure (V.15) montre la diminution de la valeur de l'entrée 1 entre l'instant $k = 1000$ et l'instant $k = 2000$. Sur les figures (V.16), (V.18), (V.19) et (V.20), on peut noter que les trois sorties du système h_1, h_3 et T_3 ont des valeurs constantes et la sortie de premier résidu reste à zéro jusqu'au temps $k = 1000$. Après cet instant, la valeur de la sortie change vers une autre valeur constante à cause du défaut qui est appliqué sur l'entrée 1 du système jusqu'à l'instant $k = 2000$. Le premier résidu prend une valeur assez petite pendant cette période de temps et revient à zéro. Cette période permet de déterminer le temps d'isolation du défaut.

Sur la figure (V.17), la sortie h_2 ne change pas et la sortie de deuxième résidu reste à zéro pendant toute la durée de la simulation, parce que le défaut n'affecte pas cette sortie. Il est clair d'après la figure (V.21), que la sortie de décision égale à 1 entre les deux instants.

Essai 4: en présence de défauts.

<i>Actionneur affecté</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
Actionneur N° 2	$k = [1200,1800]$	Dérive de [0,40] %

Un autre défaut d'une perte d'efficacité est simulé sur l'entrée de deuxième actionneur. Ce défaut est une dérive avec une amplitude d'environ 40% de la plage de variation de cette entrée entre les instants $k = 1200$ et $k = 1800$. Les résultats fournis sont illustrés sur les figures (V.22) jusqu'à (V.28).

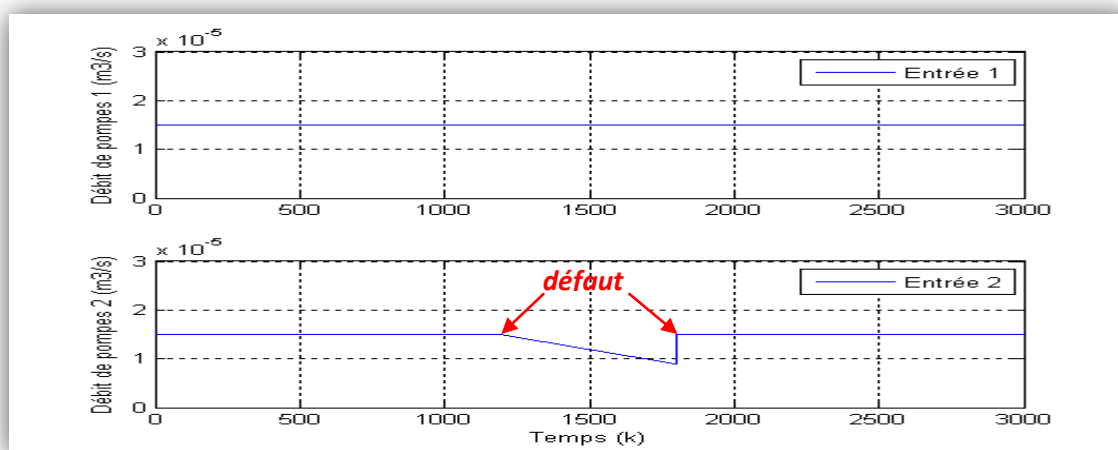


Figure V.22 : Les entrées du système.

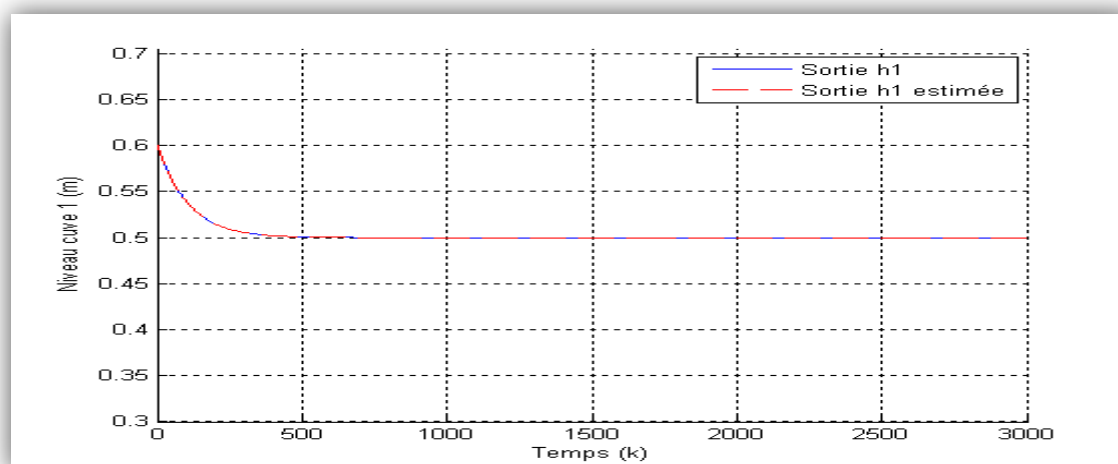


Figure V.23 : La sortie h_1 et la sortie h_1 estimée.

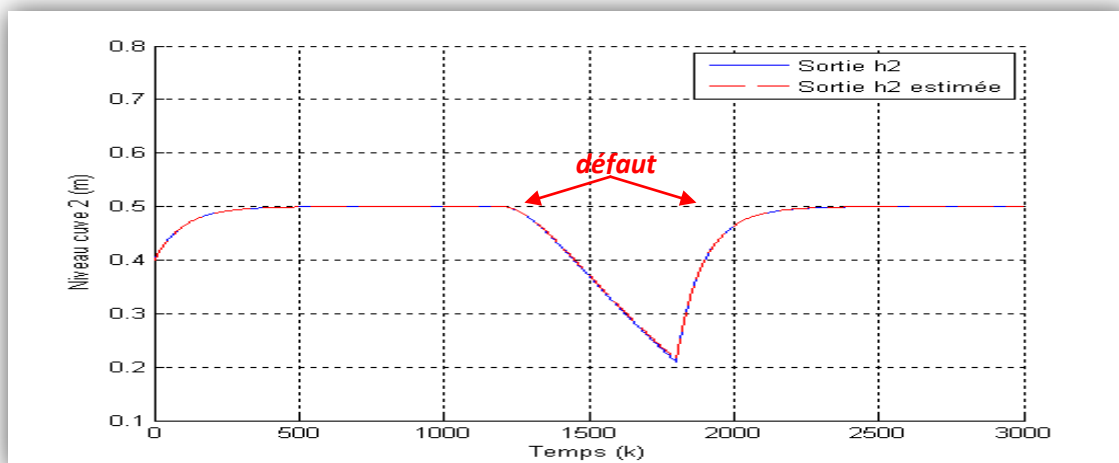


Figure V.24 : La sortie h_2 et la sortie h_2 estimée.

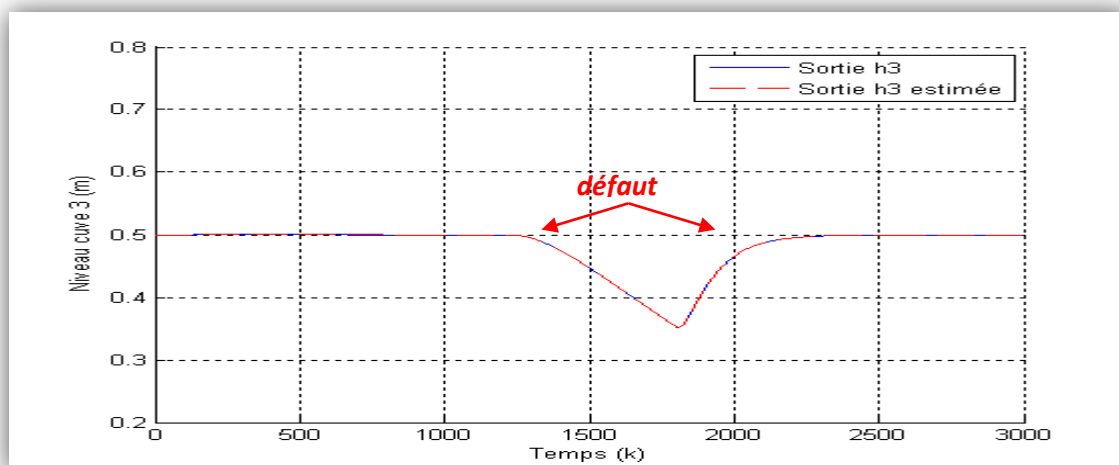


Figure V.25 : La sortie h_3 et la sortie h_3 estimée.

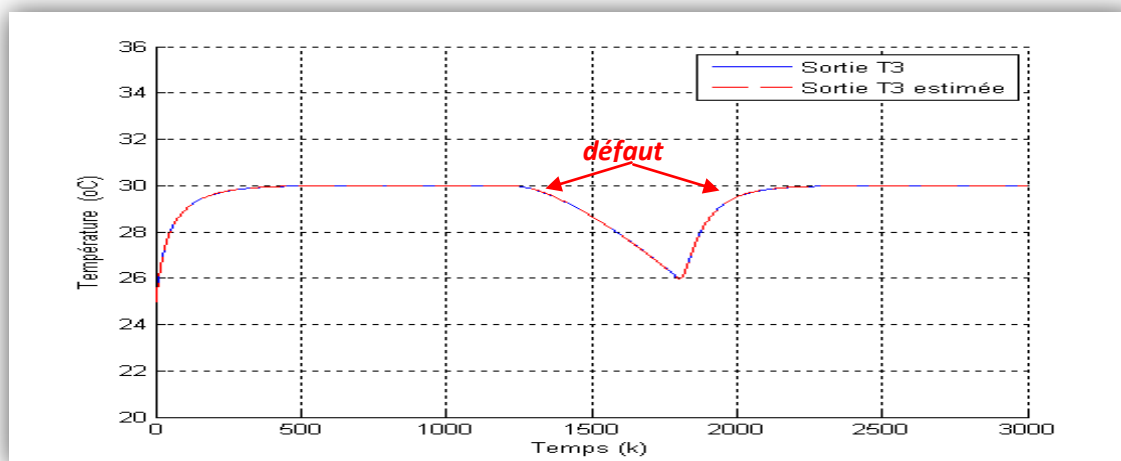


Figure V.26 : La sortie T_3 et la sortie T_3 estimée.

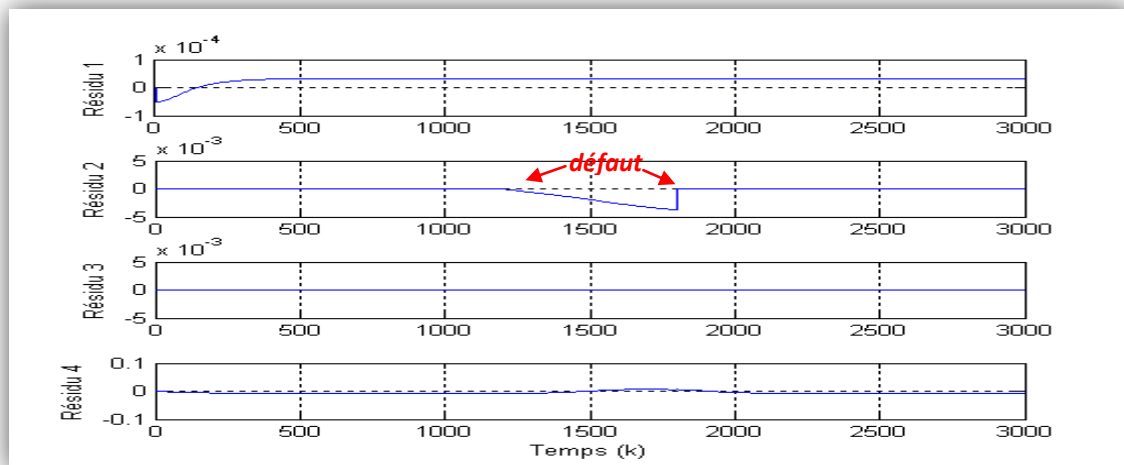


Figure V.27 : Les résidus.

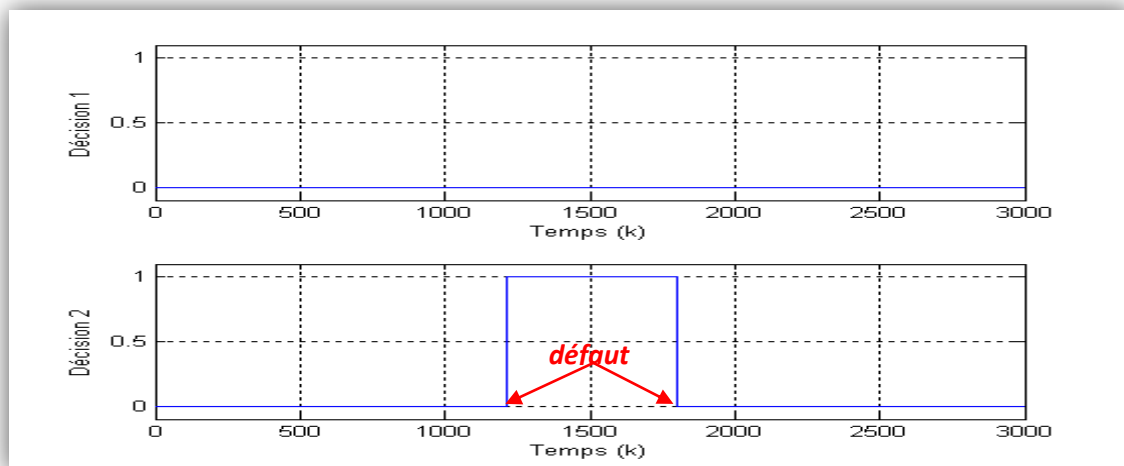


Figure V.28 : Les décisions.

A partir de figures (V.24), (V.25) et (V.26) il est clair que les trois sorties du système h_2 , h_3 et T_3 ont des valeurs constantes et la sortie de deuxième résidu reste à zéro jusqu'au temps $k = 1200$. Après cet instant, chaque valeur de ces sorties change vers une autre valeur à cause du défaut qui est appliqué sur l'entrée 2 du système jusqu'à l'instant $k = 1800$. On note aussi que le deuxième résidu prend une valeur assez petite pendant cette période de temps et revient à zéro. Sur la figure (V.23), la sortie h_1 ne change pas et la sortie de deuxième résidu reste à zéro pendant toute la durée de la simulation, parce que le défaut n'affecte pas cette sortie. Il est clair d'après la figure (V.28), que la sortie de décision égale à 1 entre les deux instants.

Essai 5 :

<i>Actionneur affecté</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
Actionneur N° 1	$k = [800,1200]$	Dérive de $[0,80]$ %
Actionneur N° 2	$k = [1500,2500]$	2%

Maintenant, on va simuler deux défauts sur les deux entrées du système. Le défaut simulé sur la première entrée est une dérive avec une perte d'amplitude d'environ 80% de la plage de variation de cette entrée entre les instant $k = 800$ et $k = 1200$. Un autre défaut est simulé sur la deuxième entrée est un biais avec une perte d'amplitude de 2% de la plage de variation de cette entrée entre les instant $k = 1500$ et $k = 2500$. Les résultats sont illustrés sur les figures (V.29) jusqu'à (V.35).

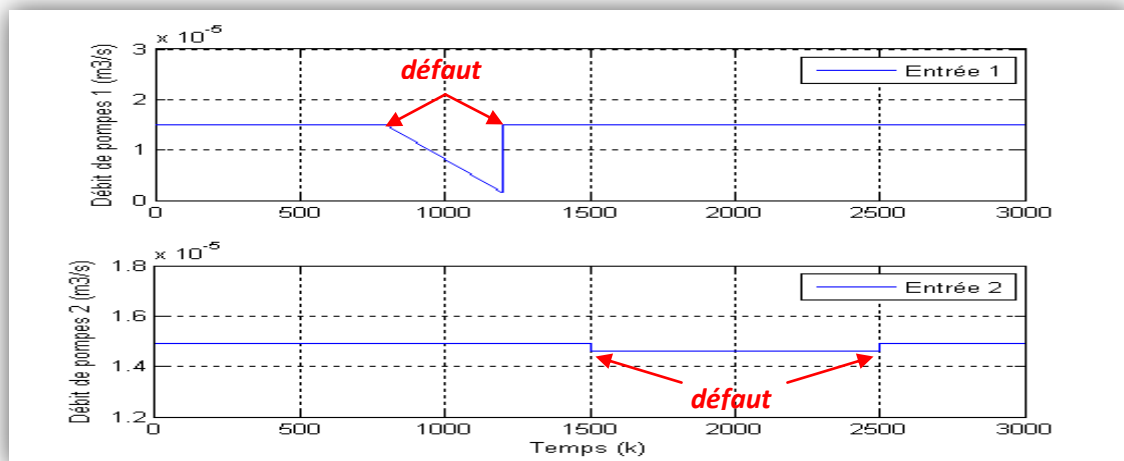


Figure V.29 : Les entrées du système.

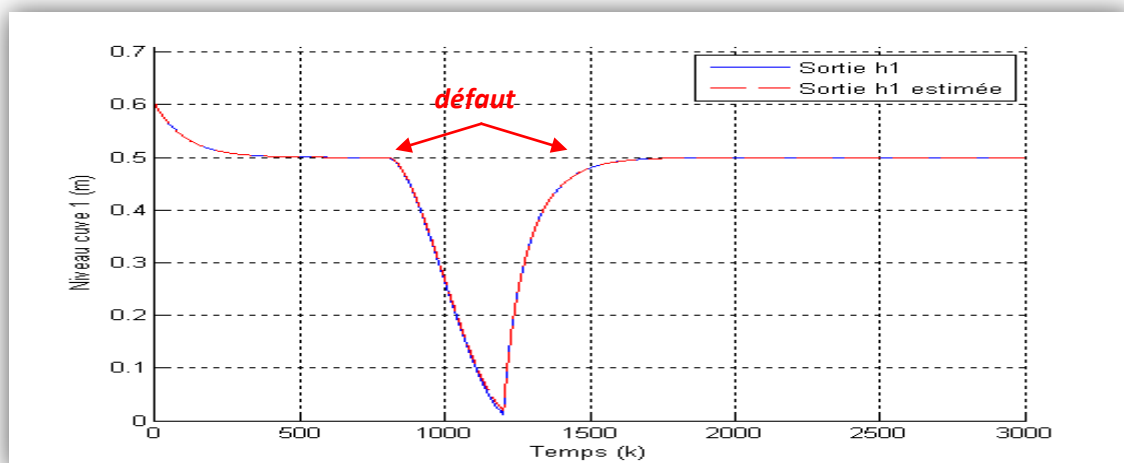


Figure V.30 : La sortie h_1 et la sortie h_1 estimée.

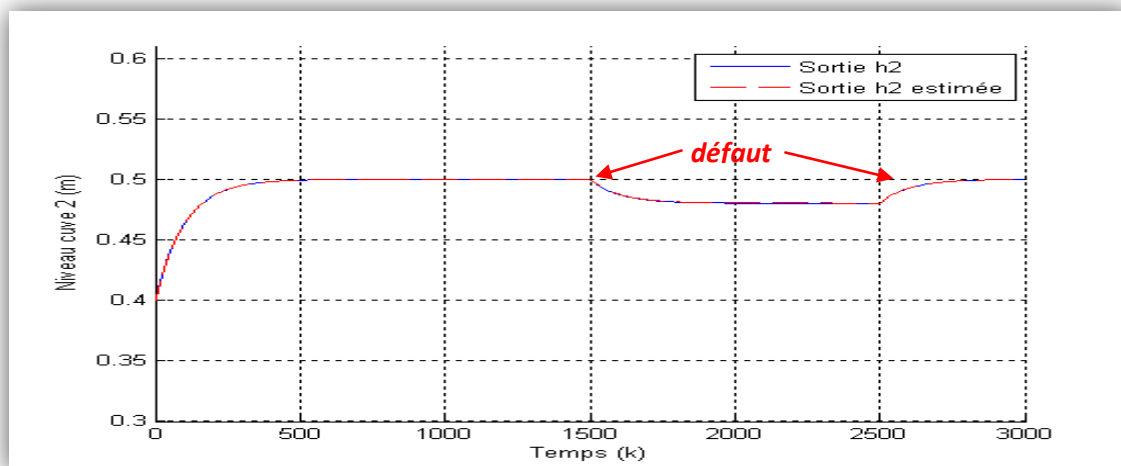


Figure V.31 : La sortie h_2 et la sortie h_2 estimée.

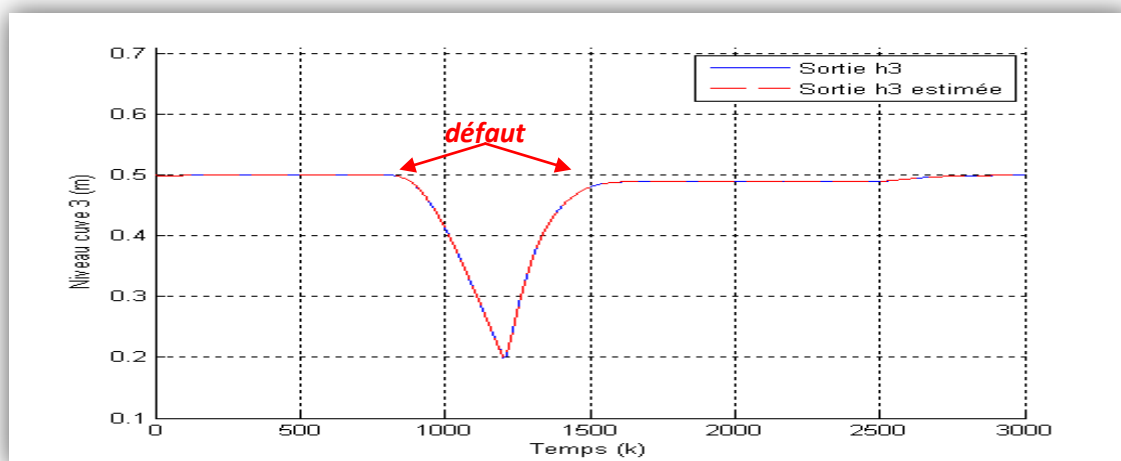


Figure V.32 : La sortie h_3 et la sortie h_3 estimée.

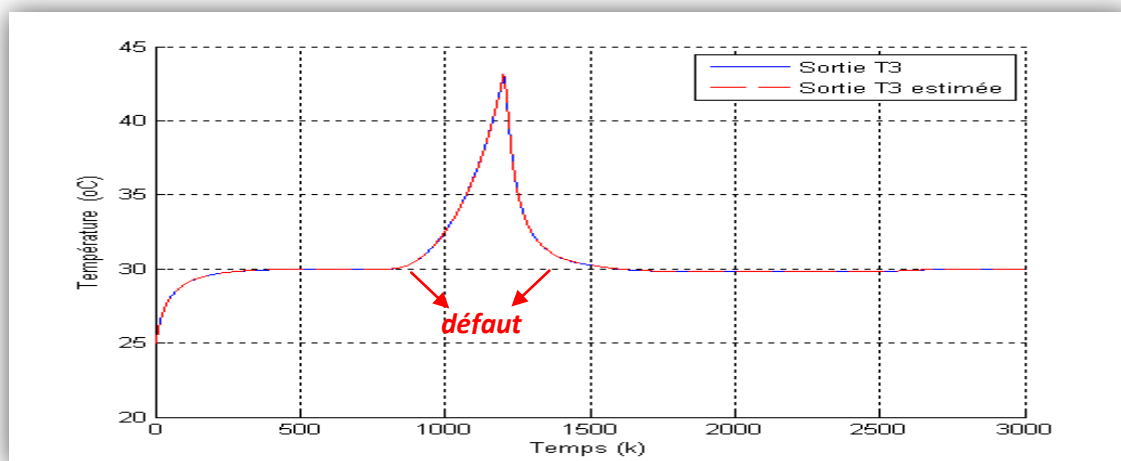


Figure V.33 : La sortie T_3 et la sortie T_3 estimée.

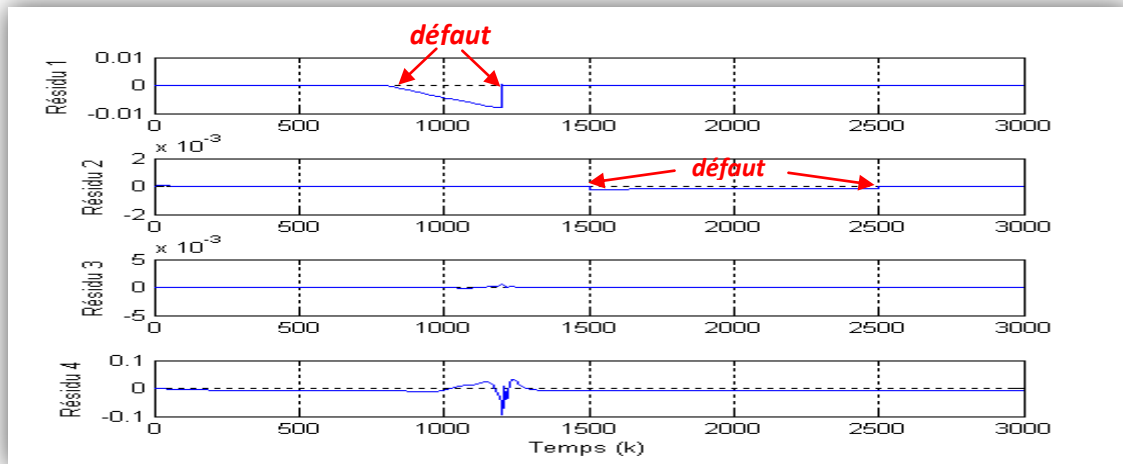


Figure V.34 : Les résidus.

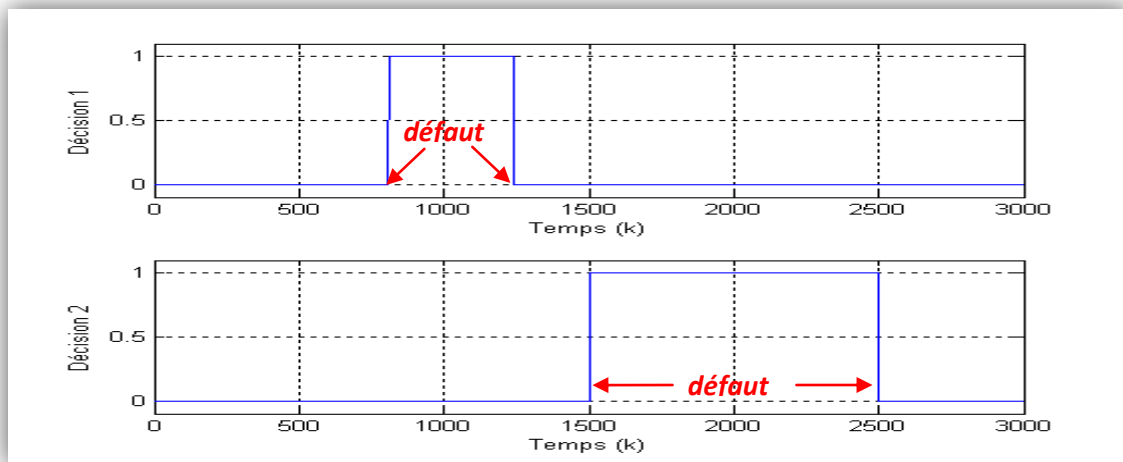


Figure V.35 : Les décisions.

On voit clairement d'après les figures (V.30), (V.32) et (V.33) que chaque sortie de ces sorties du système h_1 , h_3 et T_3 a une valeur constante jusqu'au temps $k = 800$. Après cet instant, chaque valeur de ces sorties change vers une autre valeur à cause du défaut qui est appliqué sur l'entrée 1 du système jusqu'à l'instant $k = 1200$.

Il est clair aussi d'après les figures (V.31), (V.32) et (V.33) que les sorties h_2 , h_3 et T_3 du système changent vers d'autres valeurs entre l'instant $k = 1500$ et l'instant $k = 2500$ à cause du défaut appliqué sur l'entrée 2 du système.

On note aussi que le système de diagnostic prend une décision positive malgré la valeur du défaut est très petite (2%).

Essai 6 :

<i>Actionneur affecté</i>	<i>Instant de défaut</i>	<i>Perte de l'efficacité</i>
Actionneur N° 1	$k = [1500,2000]$	90 %
Actionneur N° 2	$k = [1500,2100]$	10 %

Maintenant, on va tester notre système de diagnostic en cas de défauts simultanés sur les deux actionneurs. Le défaut simulé sur la première entrée est un biais avec une perte d'amplitude d'environ 90% de la plage de variation de cette entrée entre les instants $k = 1500$ et $k = 2000$. Un autre défaut est simulé sur la deuxième entrée est un biais avec une perte d'amplitude de 10% de la plage de variation de cette entrée entre les instants $k = 1500$ et $k = 2100$. Les résultats sont illustrés sur les figures (V.36) jusqu'à (V.42).

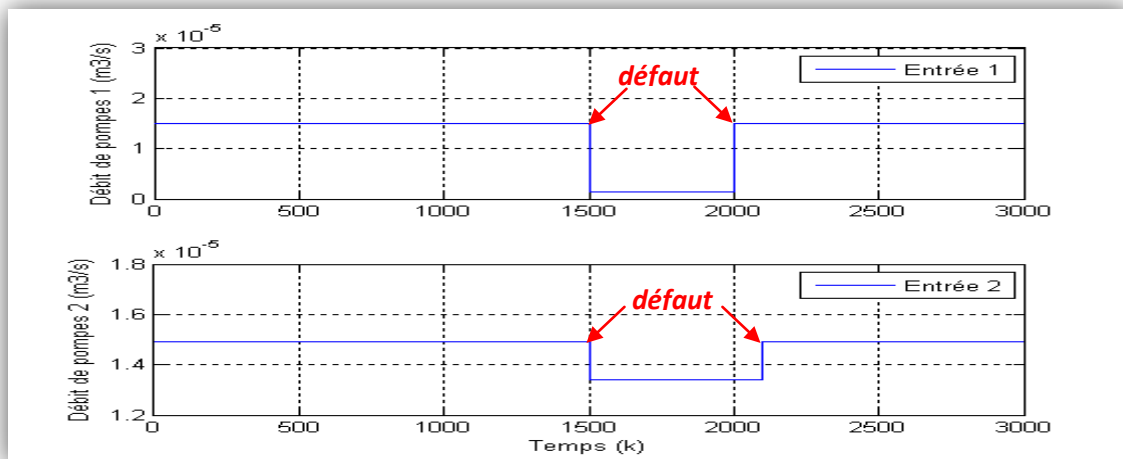


Figure V.36 : Les entrées du système.

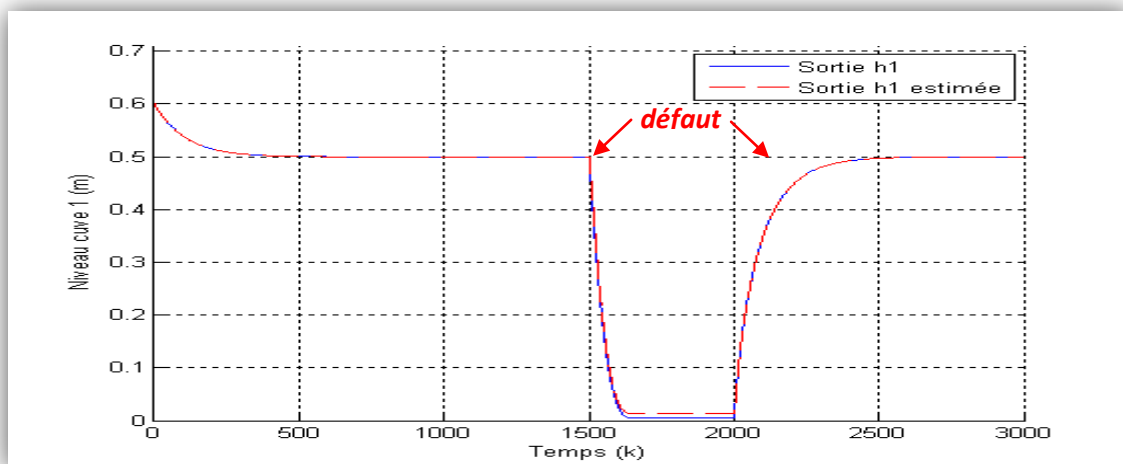


Figure V.37 : La sortie h_1 et la sortie h_1 estimée.

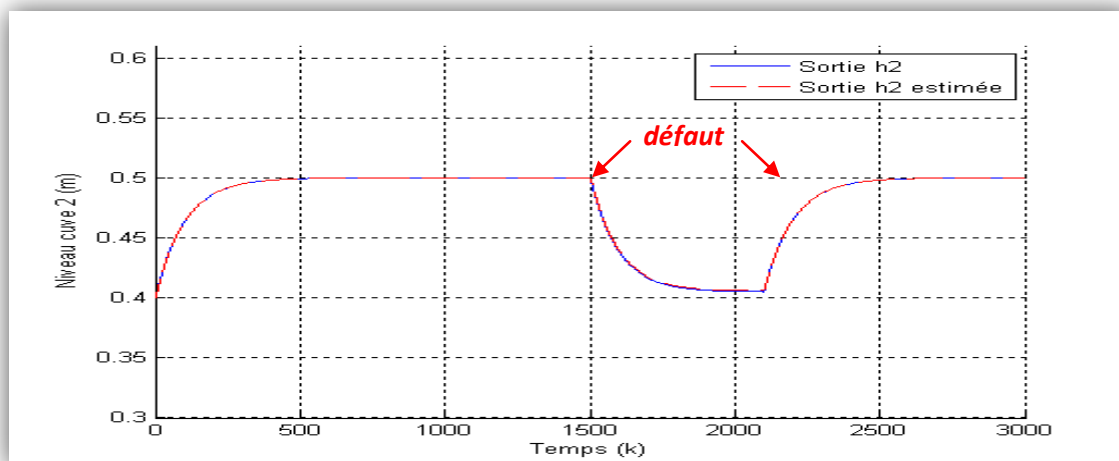


Figure V.38 : La sortie h_2 et la sortie h_2 estimée.

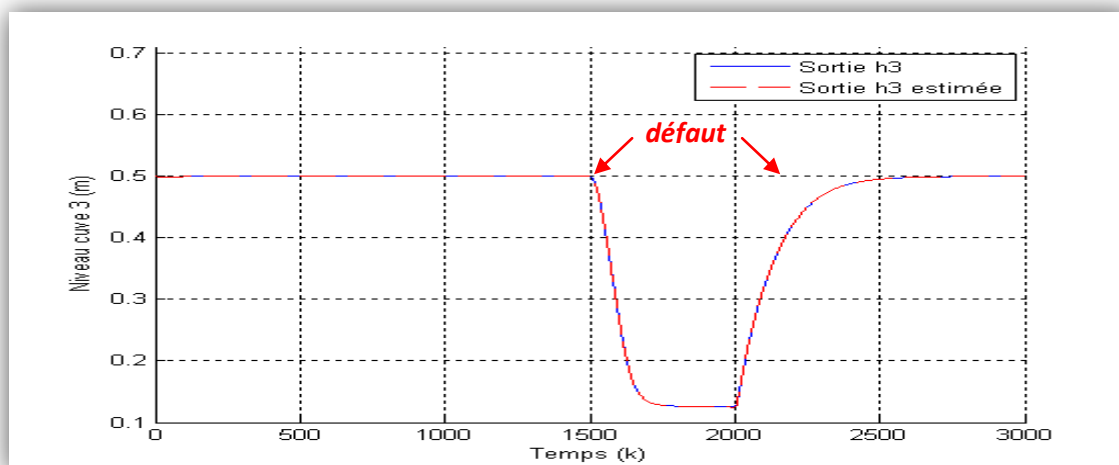


Figure V.39 : La sortie h_3 et la sortie h_3 estimée.

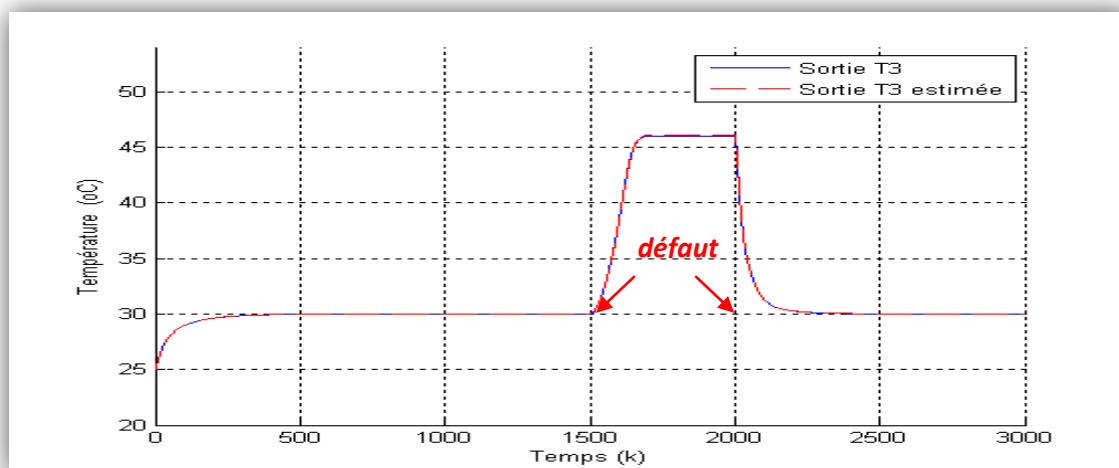


Figure V.40 : La sortie T_3 et la sortie T_3 estimée.

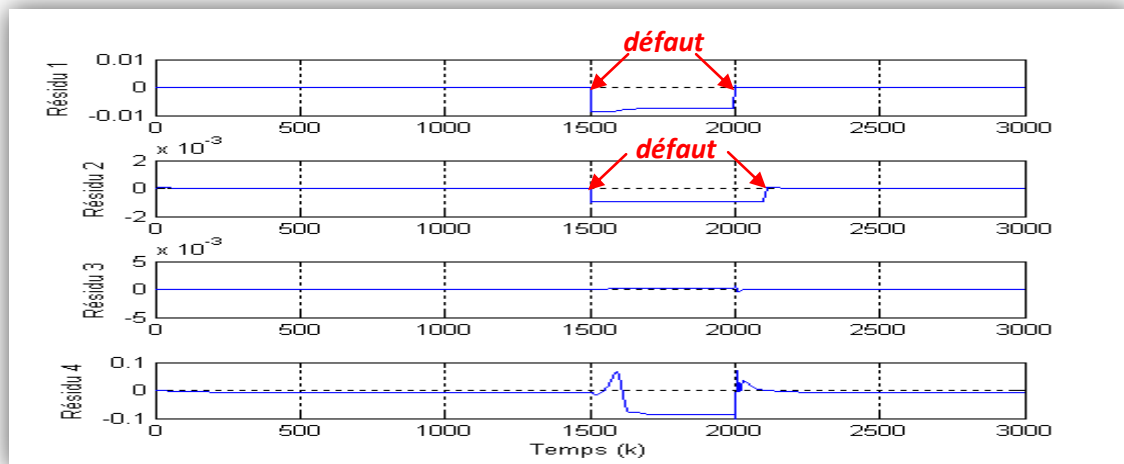


Figure V.41 : Les résidus.

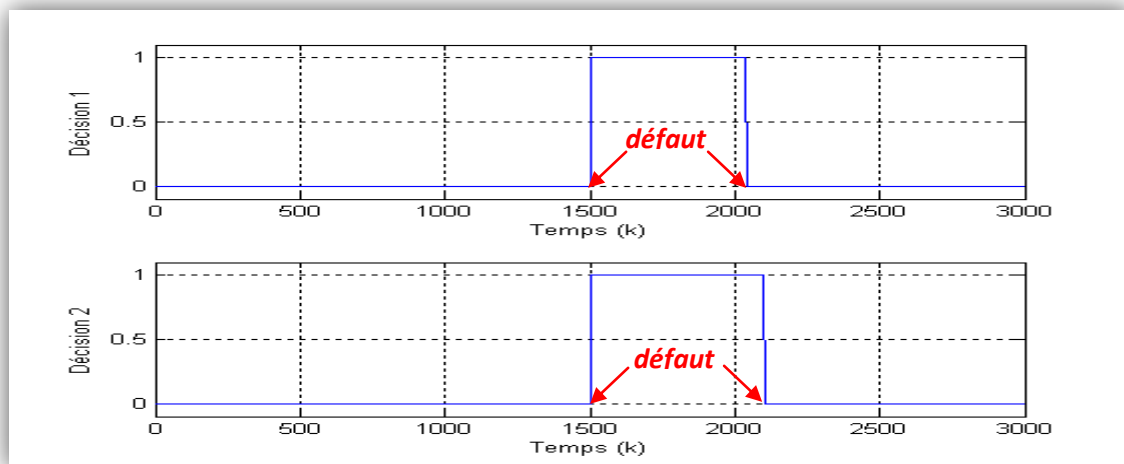


Figure V.42 : Les décisions.

On voit clairement d'après ces figures que chaque sortie du système a une valeur constante jusqu'au temps $k = 1500$. Après cet instant, chaque valeur de ces sorties change vers une autre valeur à cause du défaut qui est appliqué sur l'entrée 1 et 2 du système jusqu'à l'instant $k = 2000$ et $k = 2500$ respectivement. La décision 1 prend la valeur 1 entre les instants $k = 1500$ et $k = 2000$, et la décision 2 prend la valeur 1 entre les instants $k = 1500$ et $k = 2100$.

On note que le système de diagnostic prend des décisions positives dans le cas où les défauts provoquent les entrées du système d'une façon simultanée.

Essai 7 :

Actionneur affecté	Instant de défaut	Perte de l'efficacité
Actionneur N° 1	$k = [2000, 2200]$	20 %
Actionneur N° 1	$k = [2200, 2400]$	80 %
Actionneur N° 2	$k = 1000$	20 %

Pour simuler le cas de défauts successifs, un défaut d'amplitude 20% est injecté sur le premier actionneur entre les instants $k = 2000$ et $k = 2200$. Aussi, un autre défaut d'amplitude 80% est injecté sur le deuxième actionneur entre les instants $k = 2200$ et $k = 2400$. Enfin un défaut d'amplitude 20% est injecté sur le troisième actionneur à l'instant $k = 1000$. Les résultats sont illustrés sur les figures (V.43) jusqu'à (V.49).

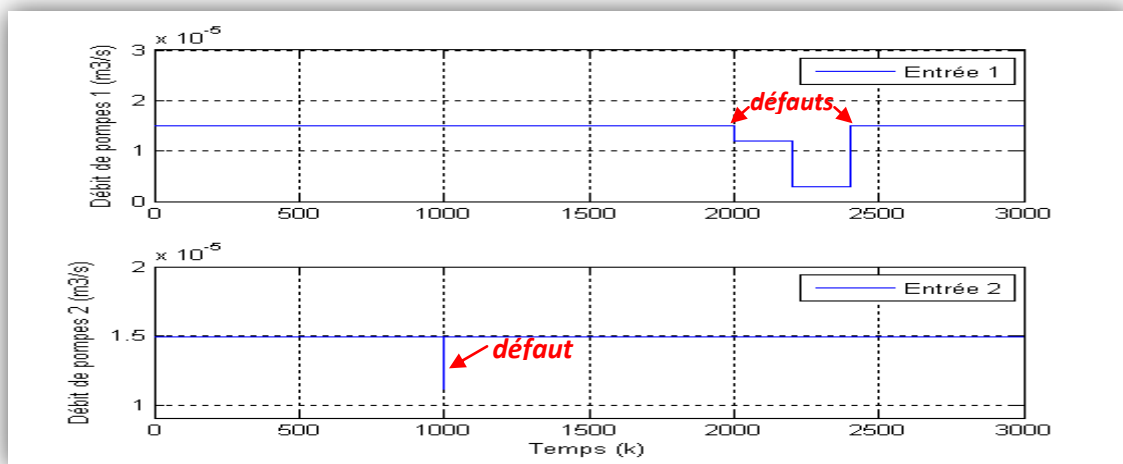


Figure V.43 : Les entrées du système.

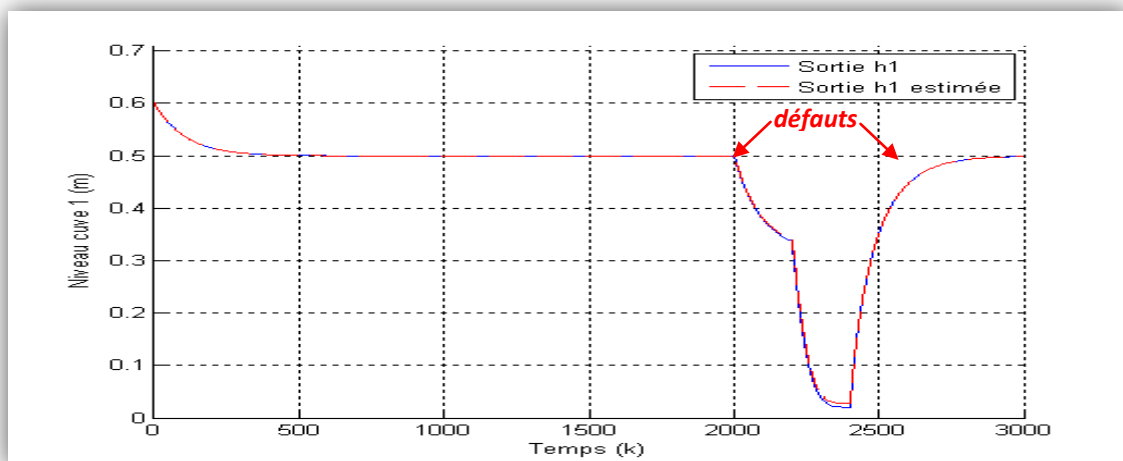


Figure V.44 : La sortie h_1 et la sortie h_1 estimée.

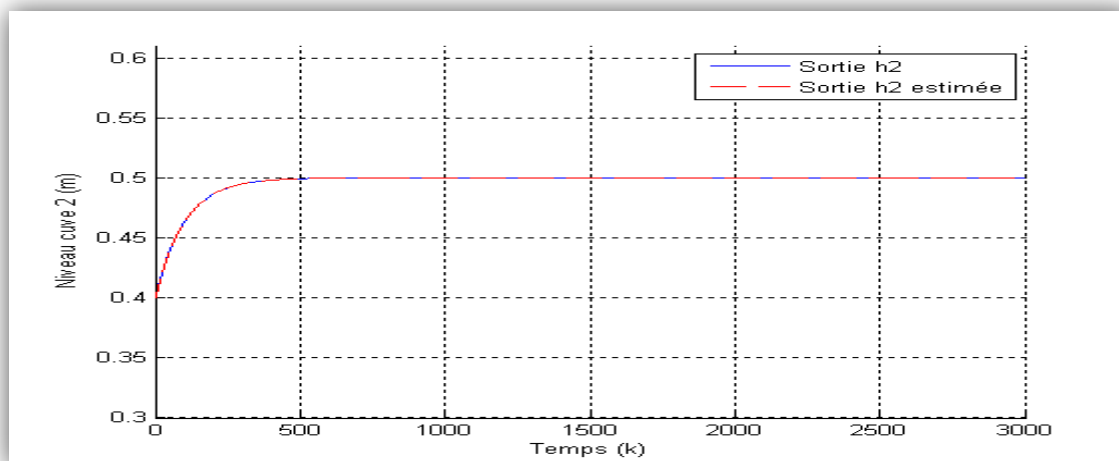


Figure V.45 : La sortie h_2 et la sortie h_2 estimée.

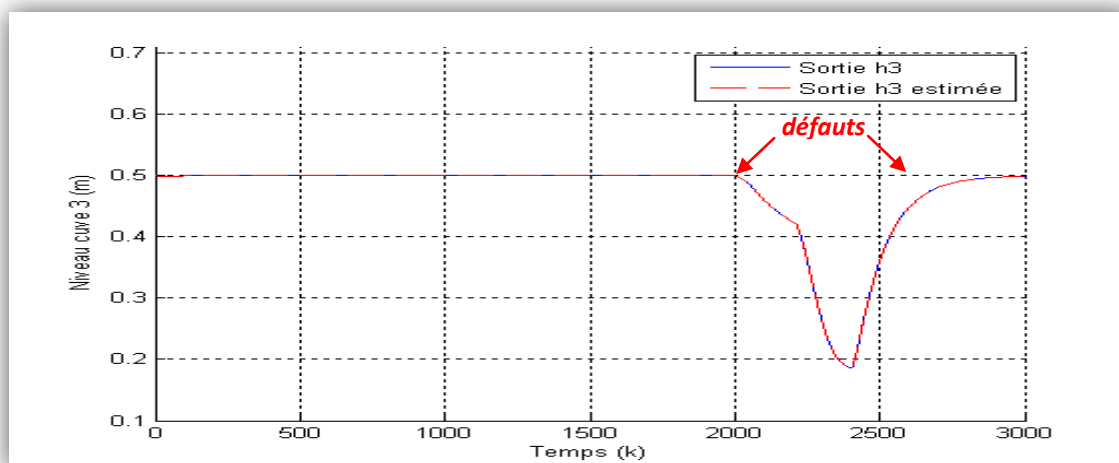


Figure V.46 : La sortie h_3 et la sortie h_3 estimée.

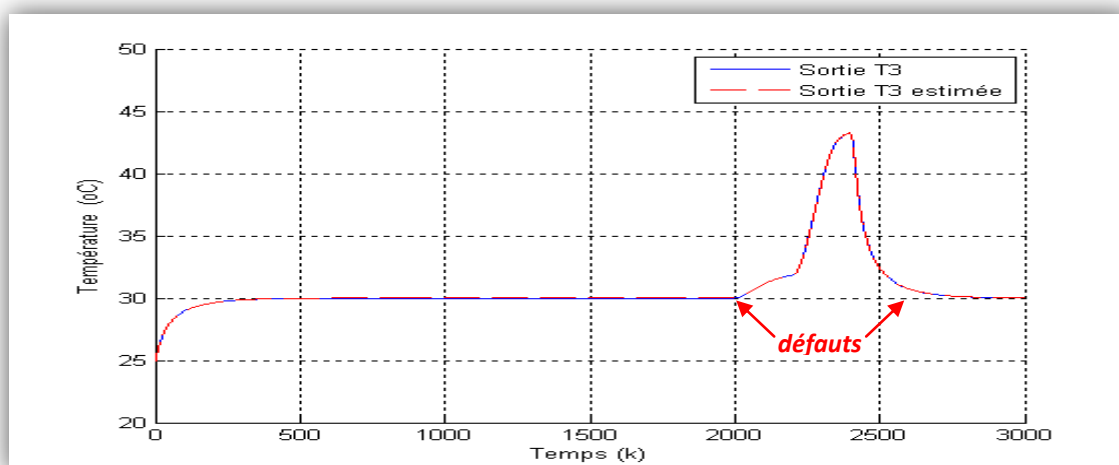


Figure V.47 : La sortie T_3 et la sortie T_3 estimée.

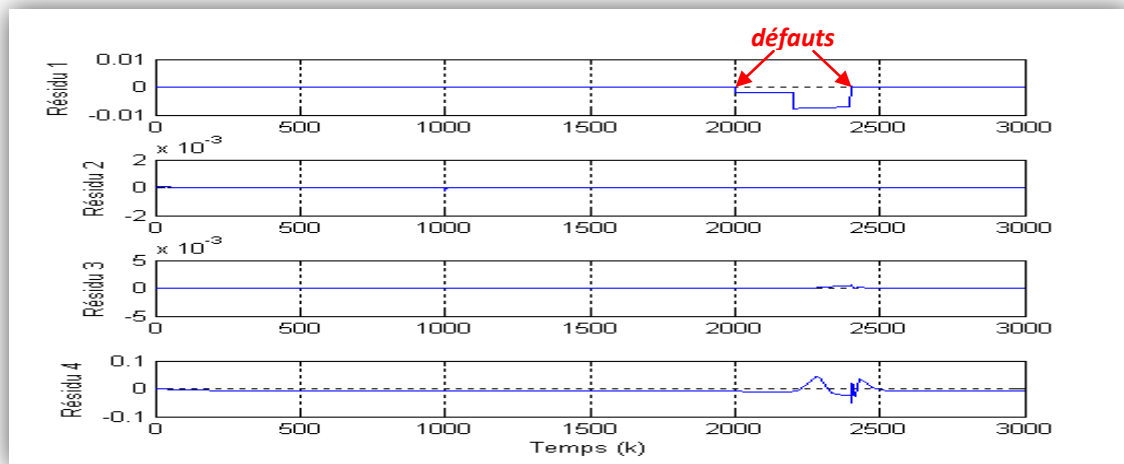


Figure V.48 : Les résidus.

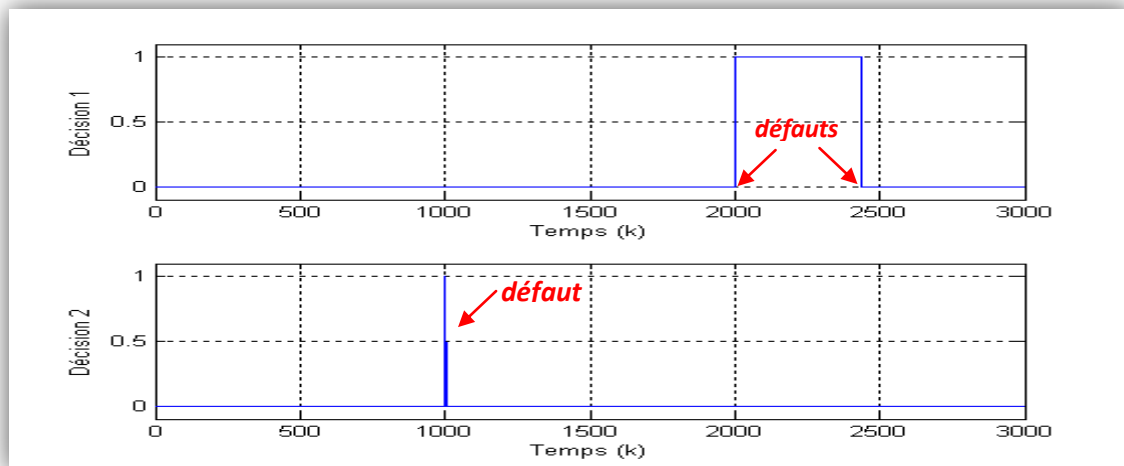


Figure V.49 : Les décisions.

On voit clairement d'après les figures (V.44), (V.46) et (V.47) que chaque sortie de ces sorties du système h_1 , h_3 et T_3 a une valeur constante jusqu'au temps $k = 2000$. Après cet instant, chaque valeur de ces sorties change vers une autre valeur à cause du défaut qui est appliqué sur l'entrée 1 du système jusqu'à l'instant $k = 2400$. Le système de diagnostic prend une décision positive entre ces deux instants dans le cas où deux défauts provoquent l'entrée du système d'une façon successive.

Pour la sortie h_2 le système de décision marque un défaut à l'instant $k = 1000$ malgré ce défaut ne cause aucun effet sur les sorties du système. Donc le système de décision a la possibilité de détecter les défauts de petite durée.

2. La 2^{ème} situation : Commande tolérante aux défauts.

Afin de vérifier et illustrer l'efficacité et la fiabilité de système de diagnostic, nous présentons maintenant la stratégie de commande tolérante aux défauts. Dans cette section nous allons présenter les résultats de certains nombre de tests simples pour vérifier l'efficacité et la fiabilité de commande tolérante aux défauts présenté au chapitre IV.

Essai 8 :

Actionneur affecté	Instant de défaut	Perte de l'efficacité
Actionneur N° 1	$k = [1000,2000]$	50 %

Au début, un défaut d'une perte d'efficacité est simulé sur l'entrée de premier actionneur. Ce défaut est représenté avec une amplitude d'environ 50 % de la plage de variation de cette entrée entre les instants $k = 1000$ et $k = 2000$. On montre les entrées du système, le défaut affecté, l'entrée additionnée, les sorties du système, les sorties du modèle, les résidus et les décisions sur les figures (V.50) jusqu'à (V.58) respectivement.

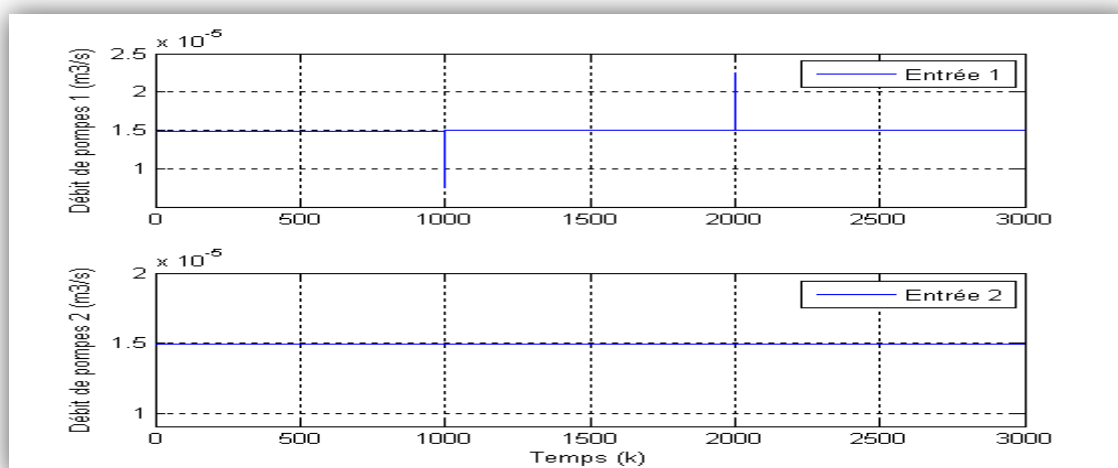


Figure V.50 : Les entrées du système.

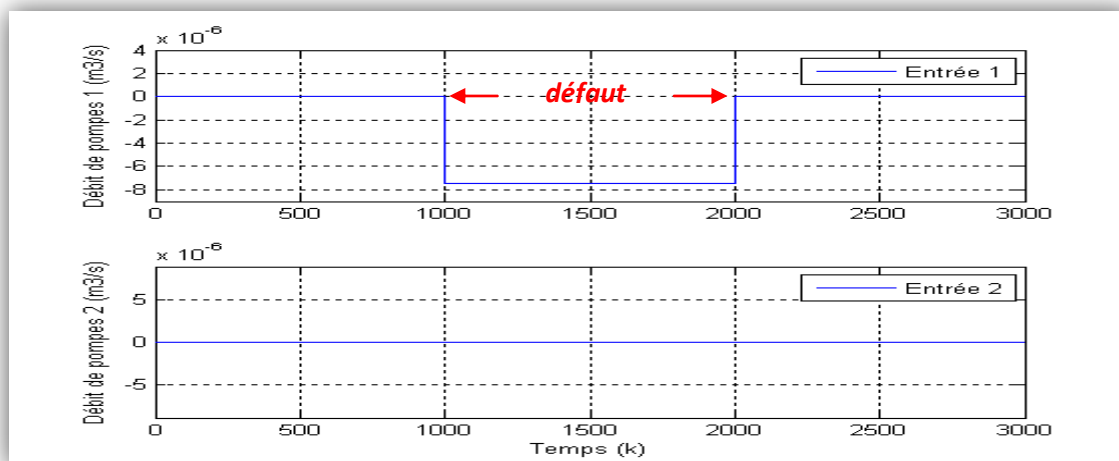


Figure V.51 : Le défaut affecté l'entrée 1.

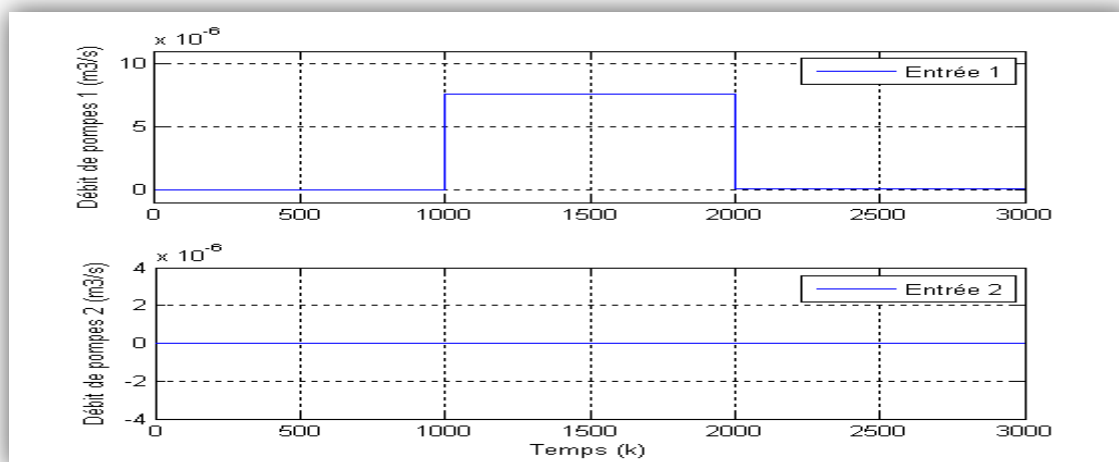


Figure V.52 : L'entrée additionnée à l'entrée 1.

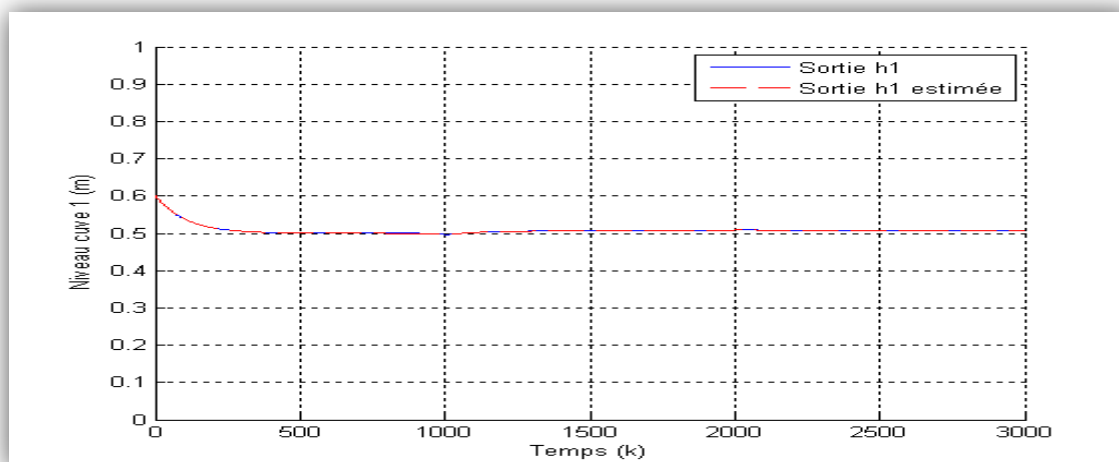


Figure V.53 : La sortie h_1 et la sortie h_1 estimée.

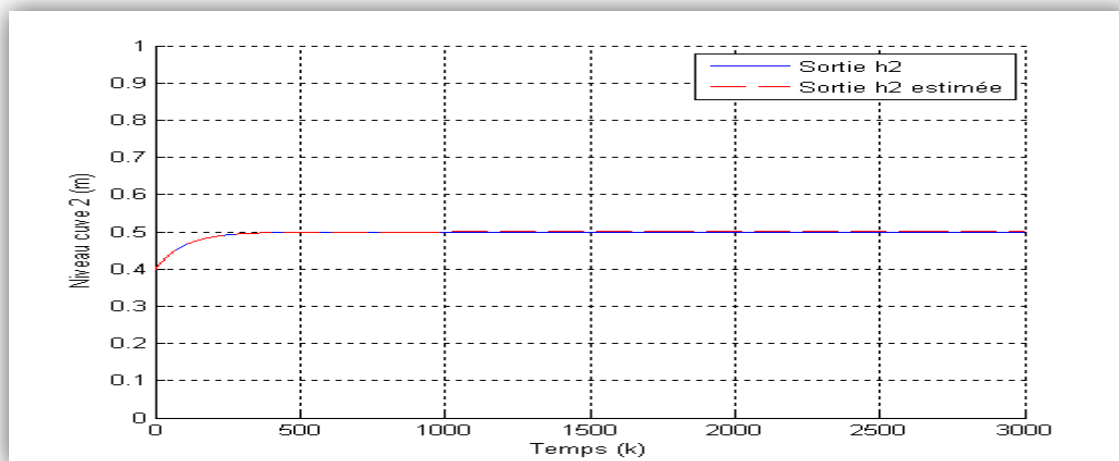


Figure V.54 : La sortie h_2 et la sortie h_2 estimée.

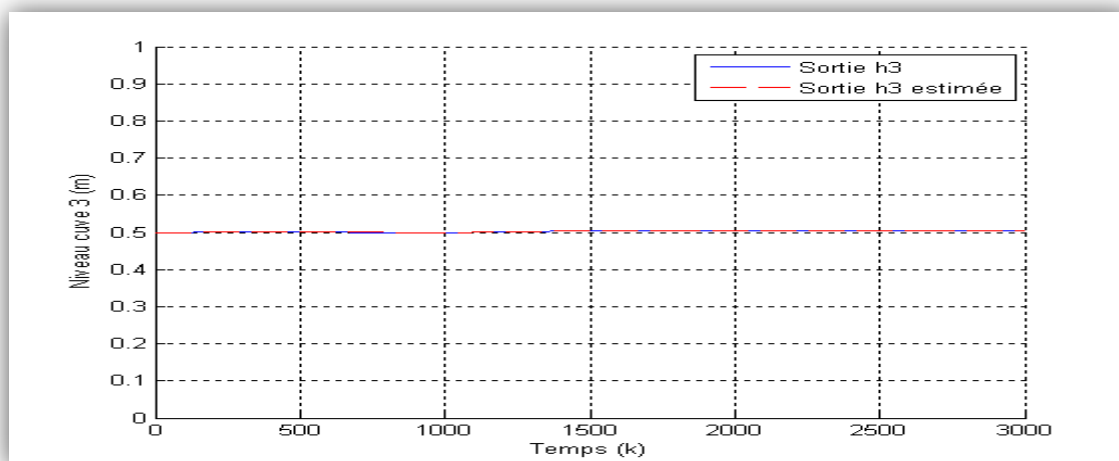


Figure V.55 : La sortie h_3 et la sortie h_3 estimée.

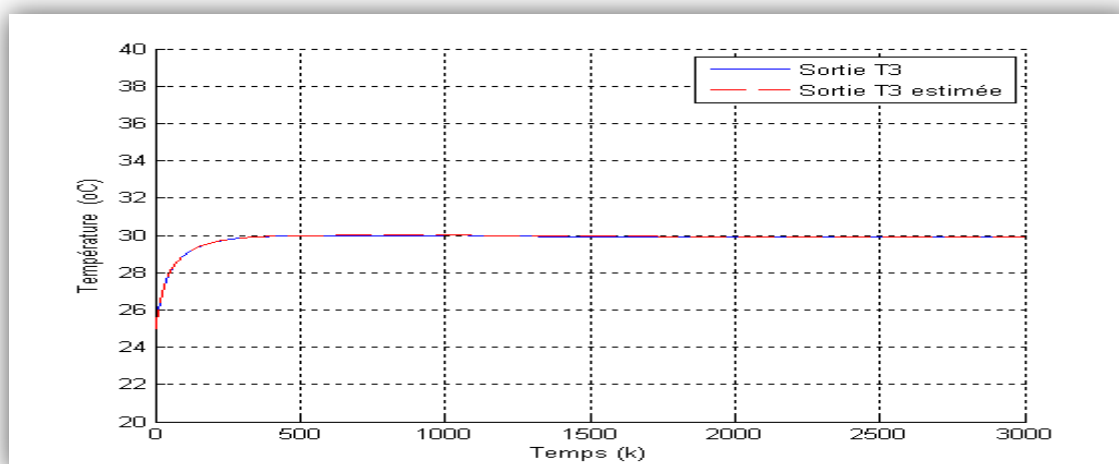


Figure V.56 : La sortie T_3 et la sortie T_3 estimée.

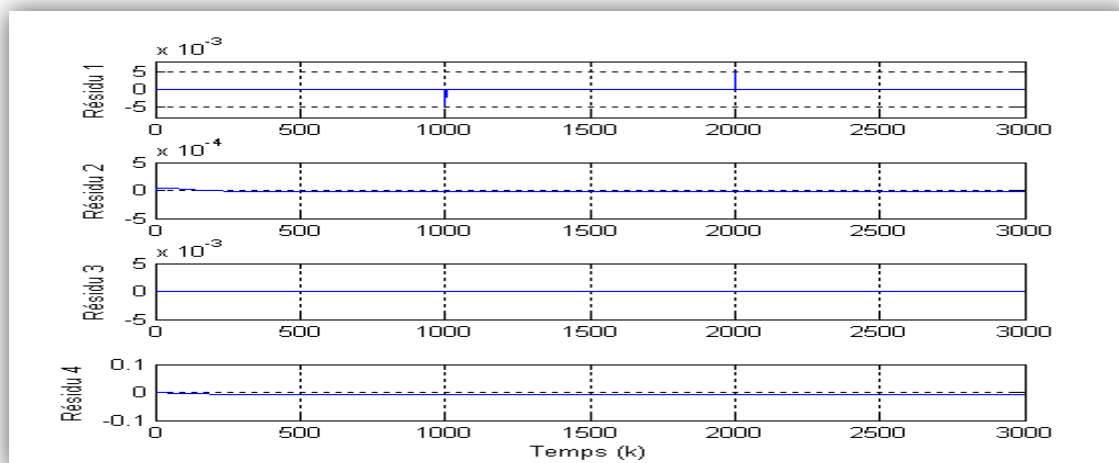


Figure V.57 : Les résidus.

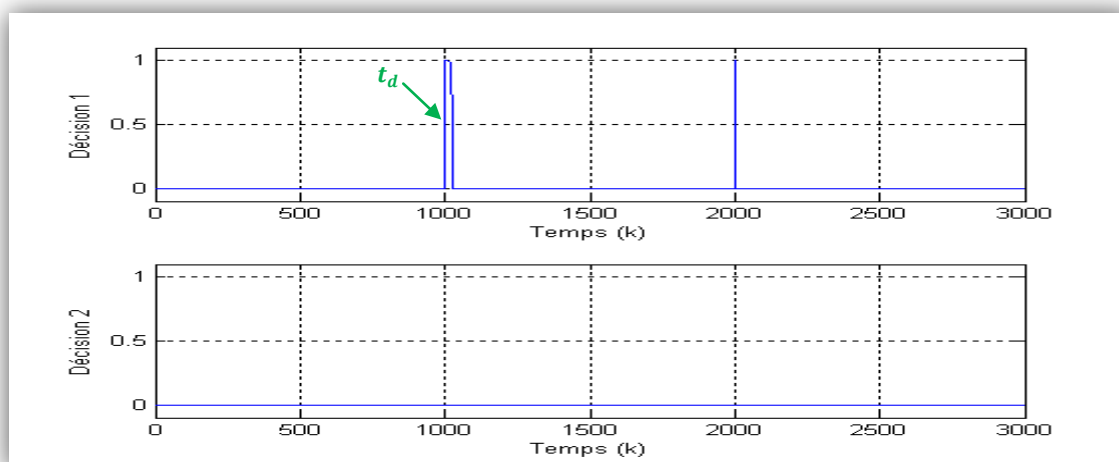


Figure V.58 : Les décisions.

On voit clairement que la commande utilisée donne de meilleurs résultats ce qui permet de donner des sorties du système pas du tout affectées à cause du défaut dans l'entrée 1. Le système de décision marque une petite saute et revient à zéro. Ce petit temps de retard est appelé le temps de détection : $t_d = 1.T_e = 10 \text{ msec}$. Donc la reconfiguration de la commande tolérante est activée après $1k = 10 \text{ msec}$ de l'apparition du défaut, ceci est le temps à peu près pour que le système atteigne son régime permanent.

Essai 9 :

Actionneur affecté	Instant de défaut	Perte de l'efficacité
Actionneur N° 1	$k = [1000,1500]$	60 %
Actionneur N° 2	$k = [2000,2500]$	40 %

Maintenant, on va simuler deux défauts sur les deux entrées du système. Le défaut simulé sur la première entrée est un biais avec une perte d'amplitude d'environ 60% de la plage de variation de cette entrée entre les instants $k = 1000$ et $k = 1500$. Un autre défaut est simulé sur la deuxième entrée est un biais avec une perte d'amplitude de 40% de la plage de variation de cette entrée entre les instants $k = 2000$ et $k = 2500$. Les résultats sont illustrés sur les figures (V.59) jusqu'à (V.67).

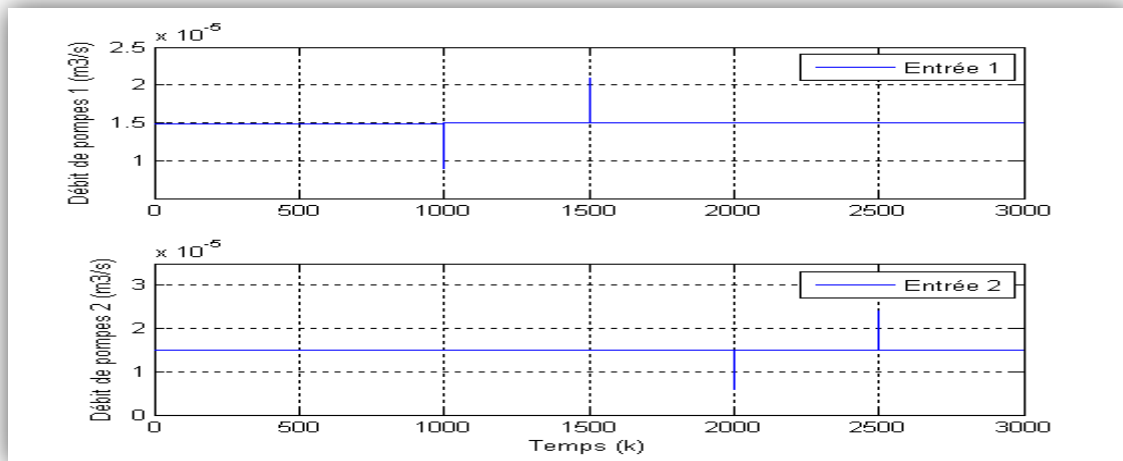


Figure V.59 : Les entrées du système.

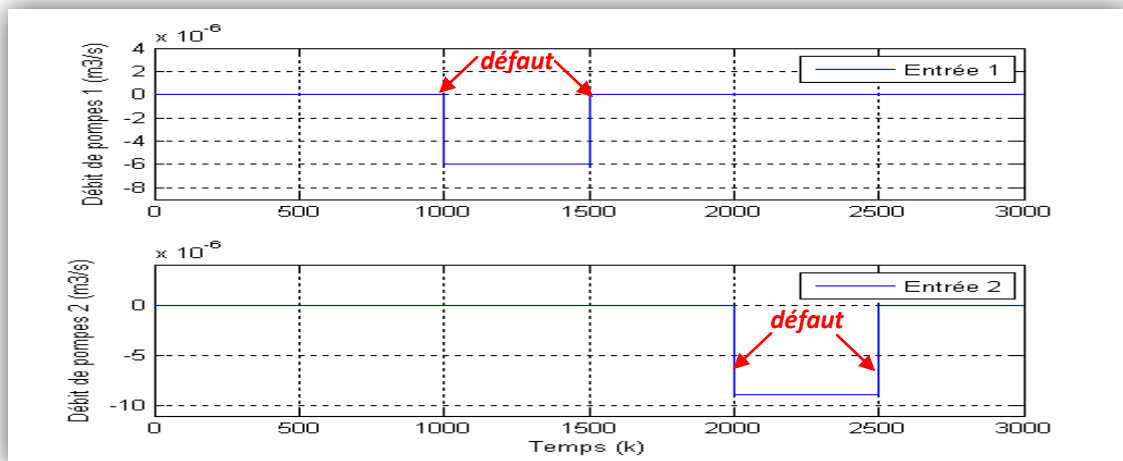


Figure V.60 : Les défauts affectés les entrées 1 et 2.

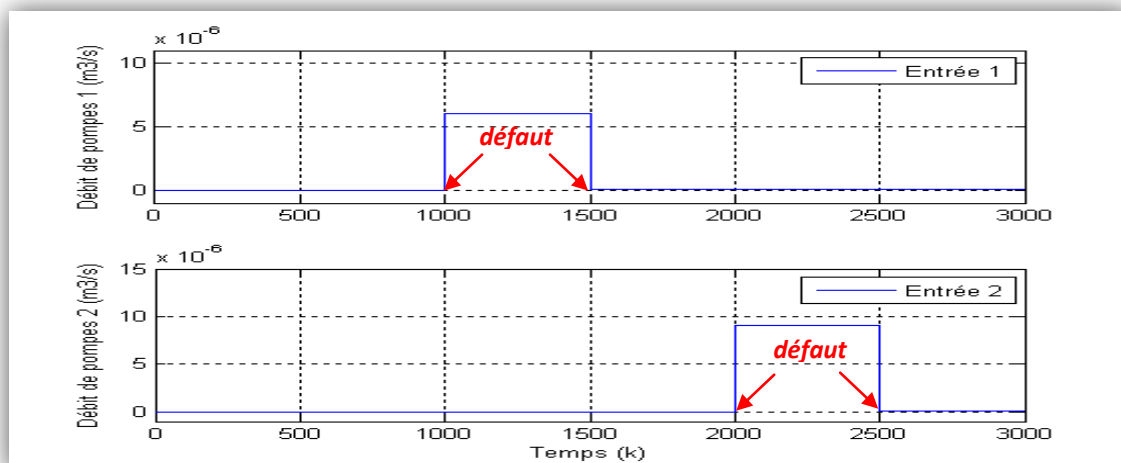


Figure V.61 : Les entrées additionnés aux entrées 1 et 2.

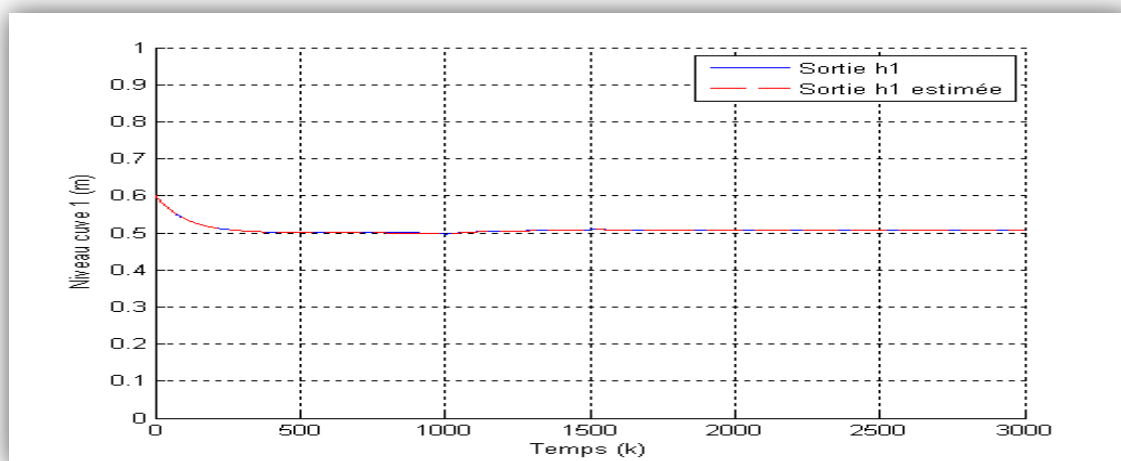


Figure V.62 : La sortie h_1 et la sortie h_1 estimée.

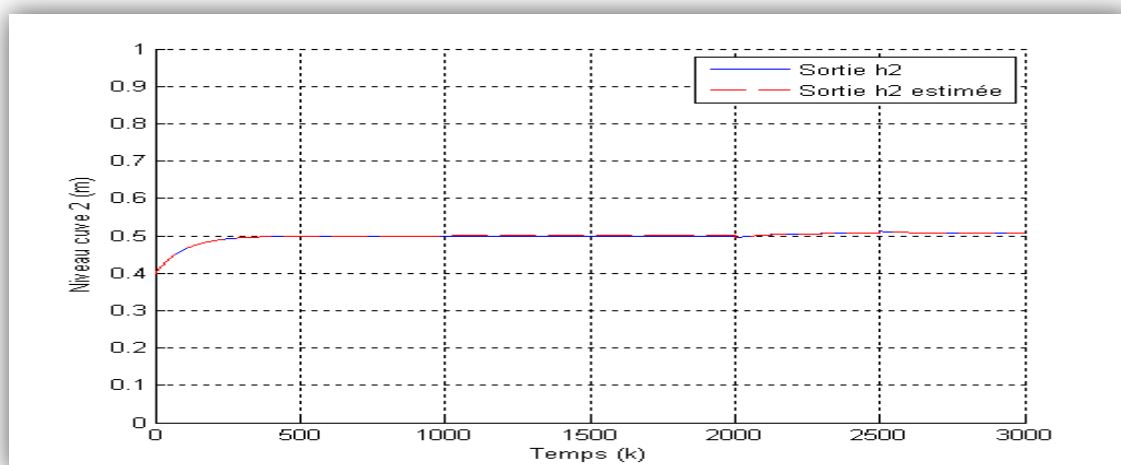


Figure V.63 : La sortie h_2 et la sortie h_2 estimée.

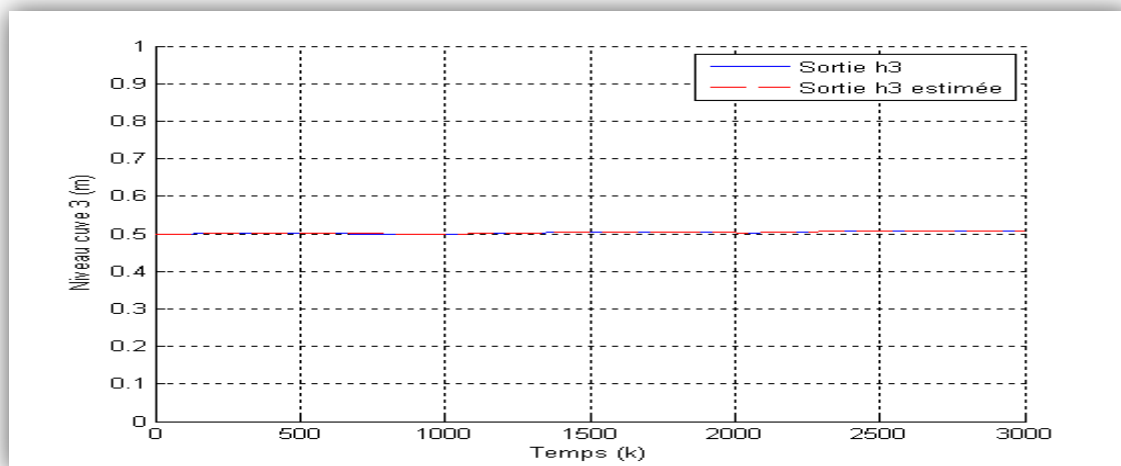


Figure V.64 : La sortie h_3 et la sortie h_3 estimée.

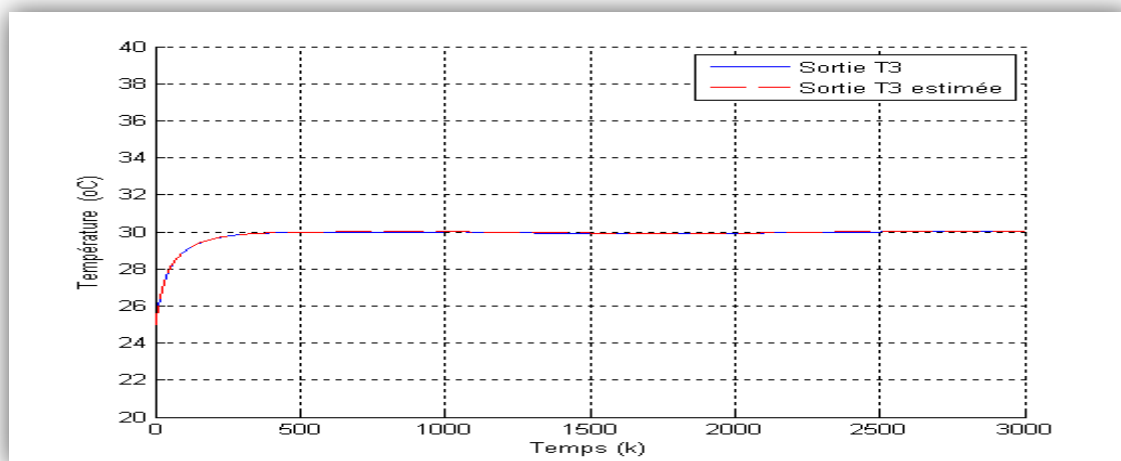


Figure V.65 : La sortie T_3 et la sortie T_3 estimée.

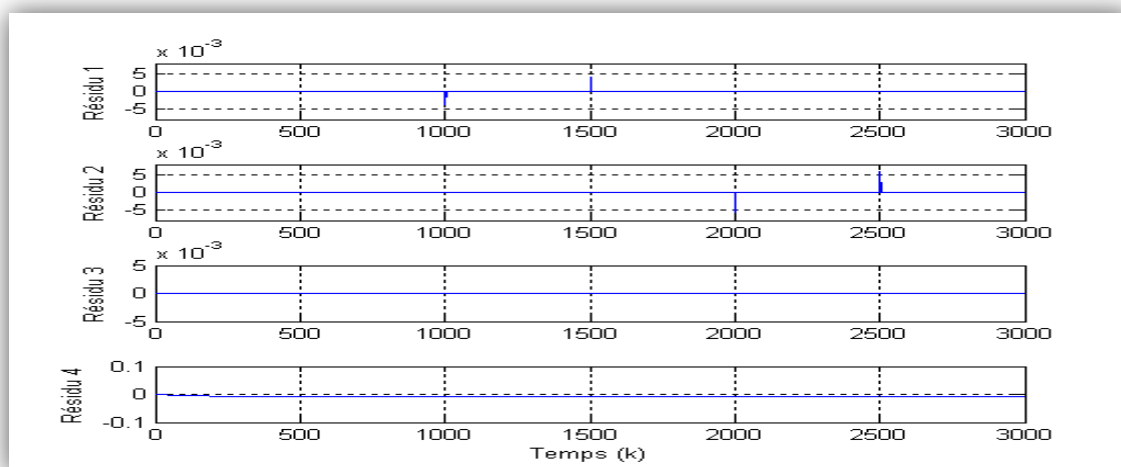


Figure V.66 : Les résidus.

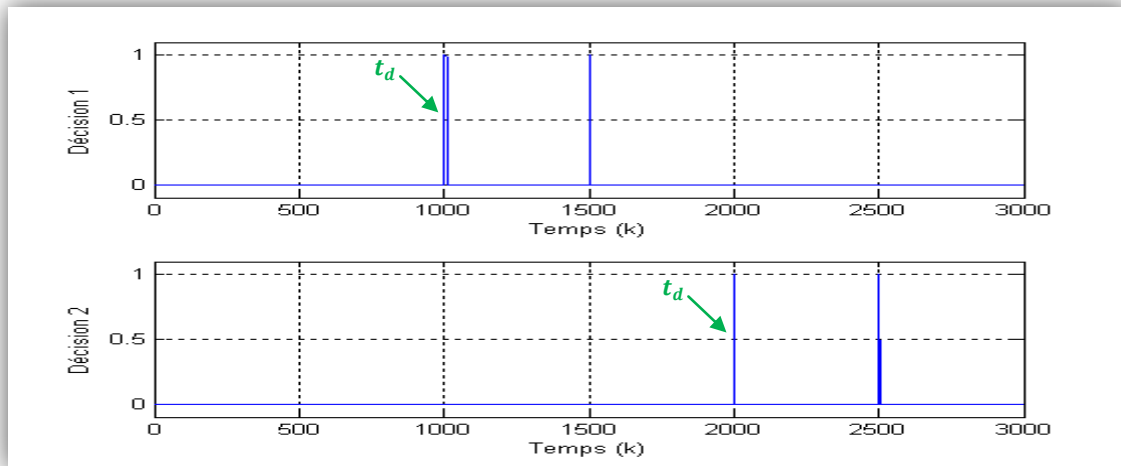


Figure V.67 : Les décisions.

On constate d'après ces figures obtenus que la reconfiguration de la commande est nécessaire car on voit clairement que la commande utilisée donne des meilleurs résultats et les sorties du système pas du tout affecté à cause des défauts dans l'entrée 1 et 2. On voit que à l'instant d'introduction du défaut, la commande (l'entrée additionnée) s'accroît considérablement et même chose concernant les sorties ce qui a rendu le fonctionnement normal tout de suite de suite après.

Le système de décision marque une petite saute et revient à zéro. Donc la reconfiguration de la commande tolérante est activée après $1k = 10 \text{ msec}$ de l'apparition du défaut.

Essai 10 :

Actionneur affecté	Instant de défaut	Perte de l'efficacité
Actionneur N° 1	$k = [1000,3000]$	50 %

Un autre défaut d'une perte d'efficacité est simulé sur l'entrée du premier actionneur. Ce défaut est un biais avec une amplitude d'environ 50% de la plage de variation de cette entrée entre les instants $k = 1000$ et $k = 3000$. Les figures (V.68) jusqu'à (V.72) illustrent les entrées et les sorties du système avec et sans reconfiguration.

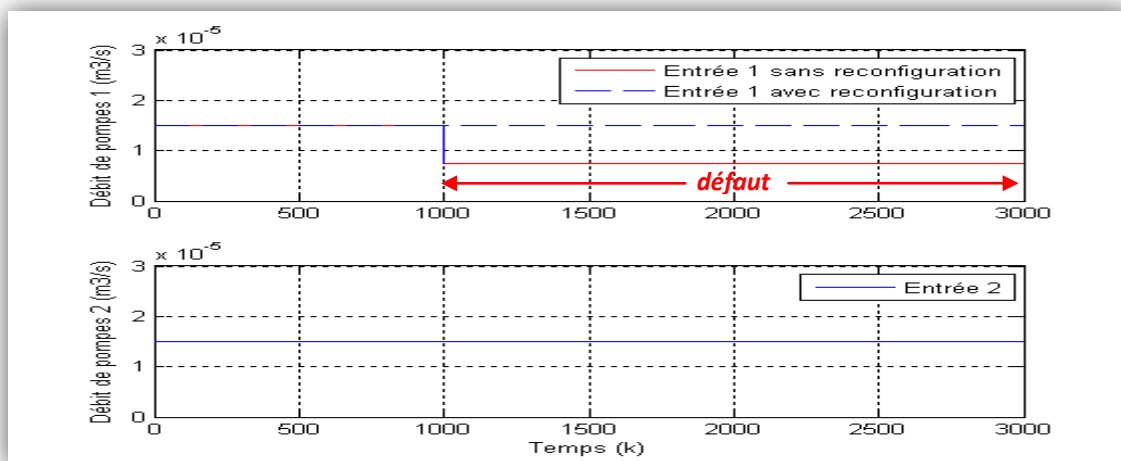


Figure V.68 : Les entrées du système.

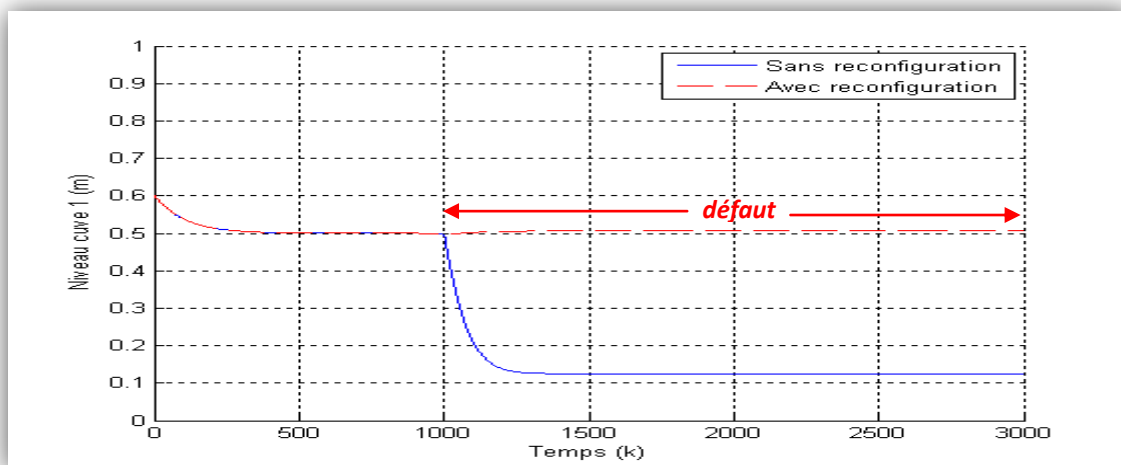


Figure V.69 : La sortie h_1 et la sortie h_1 estimée.

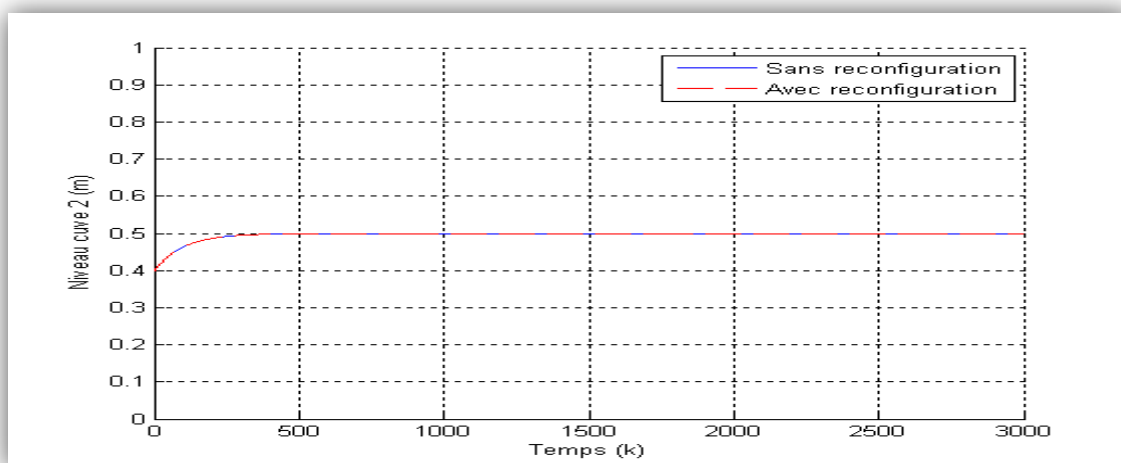


Figure V.70 : La sortie h_2 et la sortie h_2 estimée.

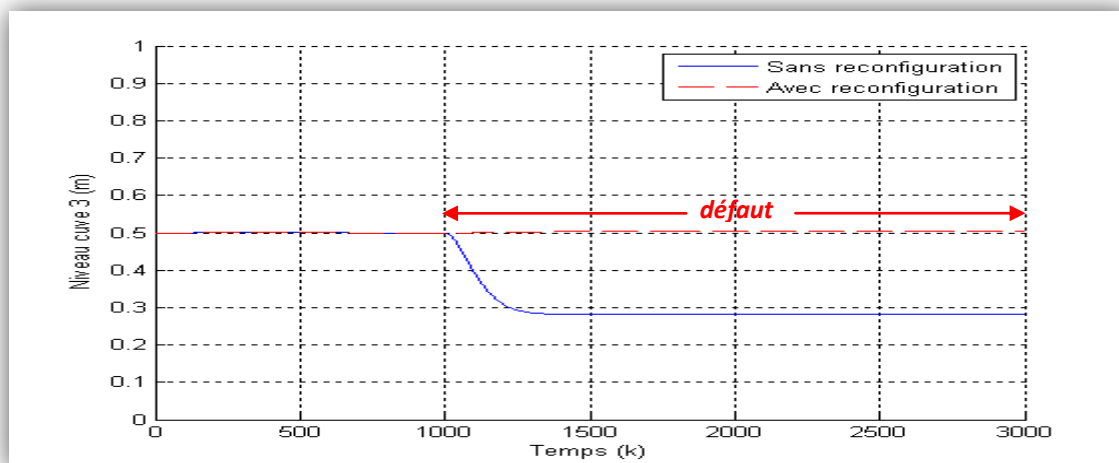


Figure V.71 : La sortie h_3 et la sortie h_3 estimée.

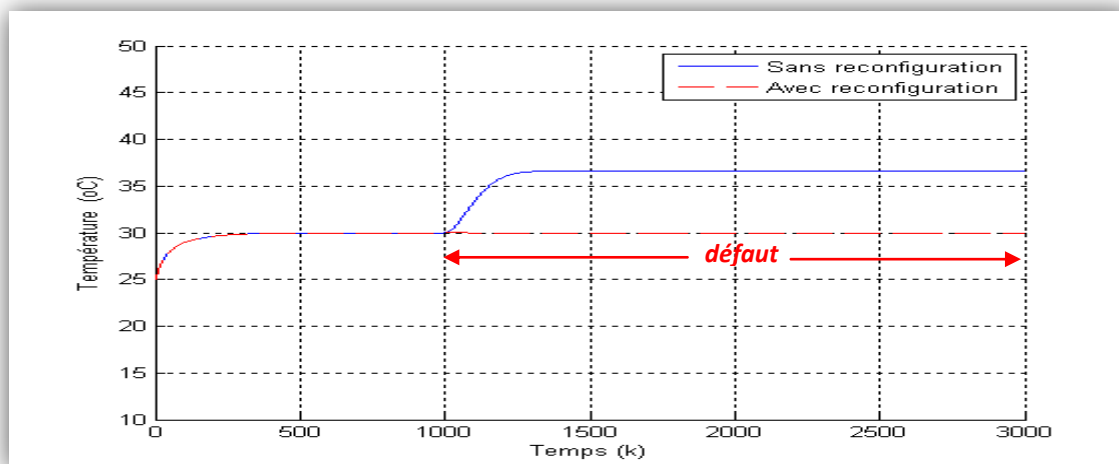


Figure V.72 : La sortie T_3 et la sortie T_3 estimée.

Les figures de cet essai illustrent une comparaison entre les deux cas, le cas de reconfiguration de commande et le cas sans reconfiguration. Dans le cas sans reconfiguration de commande les sorties h_1, h_3, T_3 s'éloignent progressivement par rapport aux sorties nominales, mais la sortie h_2 ne change pas parce que le défaut n'affecte que la première entrée. Dans le cas de commande reconfigurée, toutes les sorties restent inchangées. Donc ces résultats présentent l'importance de la commande tolérante utilisée.

V.5. CONCLUSION :

Dans ce chapitre, la méthode de commande tolérante aux défauts développée dans le chapitre précédent a été mise en œuvre en simulation sur un processus hydraulique/thermique.

Cette simulation a été faite afin d'évaluer l'effet de compensation des éventuels défauts d'actionneur qui peuvent corrompre le fonctionnement normal du système. Les résultats de la simulation ont été montrés les propriétés désirées relativement à l'annulation des effets fatals liés à l'appariation d'un ou de plusieurs défauts sur le système.

Enfin on peut conclure que l'objectif d'intégration d'un outil de diagnostic basé sur l'estimation des défauts dans une loi de commande est atteint.

Conclusion générale

Conclusion générale

La commande tolérante aux défauts (FTC) a pour but de s'accommoder automatiquement de l'effet des défauts tout en étant capable de maintenir la stabilité et au mieux les performances nominaux du système. La conséquence est d'éviter l'arrêt immédiat du système et de permettre son fonctionnement en mode dégradé.

On distingue deux approches de commandes tolérantes aux défauts : passive et active. Dans l'approche passive on utilise les techniques de commande robuste de manière à ce que le système en boucle fermée reste insensible à un ensemble connu de défauts. A la différence, l'approche active réagit 'activement' sur les défauts en reconfigurant la loi de commande de façon de maintenir la stabilité et les performances du système. Cette approche permet de traiter des défauts imprévus mais nécessite une commande reconfigurable et un module de diagnostic permettant la détection, l'isolation et l'estimation de l'amplitude des défauts.

Le travail effectué dans ce mémoire concerne la mise en place d'une démarche méthodologique pour le développement d'une stratégie de commande tolérante aux défauts. Il porte aussi sur l'étude d'un système d'aide au domaine du diagnostic des défauts avec des techniques de l'intelligence artificielle.

Les techniques développées sont applicables aux systèmes multivariables et modélisables sous forme des modèles basées sur les réseaux de neurones et les systèmes d'inférence floue. Nous avons utilisé la modélisation neuro-floue pour caractériser et prendre en compte une très large classe d'incertitudes de modèle et également les défauts dont l'impact sur le système se traduit par des variations paramétriques anormales.

La méthode de diagnostic que nous avons présentée dans ce travail est basée sur l'estimation en ligne des défauts, ce qui permet d'établir un diagnostic direct (isolation et identification du défaut détecté), c'est-à-dire ils permettent une isolation de tous les défauts (même simultanés), et de donner une information directe sur la nature du défaut ainsi que son degré de sévérité (amplitude). En revanche, la stratégie de commande proposée est fondée sur un schéma coopératif qui fédère au sein de la même structure de système de commande nominale, le module de diagnostic et les régulateurs chargés d'assurer la tolérance aux défauts.

La loi de commande tolérante proposée repose sur les trois étapes essentielles suivantes : une stratégie de modélisation des systèmes non linéaires est représenté avec un système neuro-flou. Cette première étape permet de générer les résidus qui seront utilisés dans l'étape d'évaluation des résidus. La deuxième étape est représentée avec un modèle de raisonnement flou dans le but de classer les défauts détectés. La troisième étape sert à sélectionner des lois de commande neuro/floues pré-calculées, ceci nous a amené à introduire une commande tolérante aux défauts capable de compenser l'effet des défauts par l'ajout d'un terme de compensation. Les lois de commande tolérante proposées reposent sur une estimation parfaite des défauts actionneurs.

Les résultats obtenus dans ce mémoire offrent des perspectives intéressantes de développements. En particulier, les points suivants peuvent faire l'objet de travaux futurs :

- Concernant le module de diagnostic, déterminer un seuil adaptatif de détection de défauts pourrait être une solution pour évaluer les résidus générés. En effet, afin d'accroître les performances à la détection de défaut entre les fausses alarmes et la non détection, il serait intéressant de réaliser un seuil de détection adaptatif en fonction des incertitudes de modèle.
- La validation des méthodes de simulation développées sur le même procédé hydraulique/thermique mais en utilisant des données réelles est un point très important qui va nous permettre d'exhiber la pertinence et la potentialité de notre approche sur une application pratique.

Annexe

Annexe

Les valeurs nominales de différents paramètres du système sont résumées dans le tableau suivant :

Symbole	Paramètre	Valeur
$q_{1,max}$	Débit d'entrée maximal de la cuve 1.	$3.10^{-5} m^3/s$
$q_{2,max}$	Débit d'entrée maximal de la cuve 2.	$3.10^{-5} m^3/s$
h_i	Hauteur du fluide dans la cuve i . $i = 1,2,3$.	$[0 - 1] m$
h_{10}	Hauteur initiale du fluide dans la cuve 1.	$0.6 m$
h_{20}	Hauteur initiale du fluide dans la cuve 2.	$0.4 m$
h_{30}	Hauteur initiale du fluide dans la cuve 3.	$0.5 m$
T_1	Température initiale du fluide dans la cuve 1	$10 C^o$
T_2	Température initiale du fluide dans la cuve 2	$50 C^o$
T_3	Température initiale du fluide dans la cuve 3	$25 C^o$
S	Section des cuves 1, 2 et 3.	$0.0154 m^2$
α_1	Coefficient de viscosité du fluide dans la cuve 1.	$2.1082.10^{-5}$
α_2	Coefficient de viscosité du fluide dans la cuve 2.	$2.1082.10^{-5}$
α_3	Coefficient de viscosité du fluide dans la cuve 3.	$4.2164.10^{-5}$

Tableau 1: Les valeurs numériques de différents paramètres du système.

Le tableau ci-dessous résume les caractéristiques des modèles ANFIS utilisés :

	ANFIS1	ANFIS2	ANFIS 3	ANFIS 4
Univers de discours de Q_1	$[0-3.10^{-5}]$	-	-	-
Univers de discours de Q_2	-	$[0-3.10^{-5}]$	-	-
Univers de discours de h_1	$[0-1]$	-	$[0-1]$	$[0-1]$
Univers de discours de h_2	-	$[0-1]$	$[0-1]$	$[0-1]$
Univers de discours de h_3	-	-	$[0-1]$	$[0-1]$
Univers de discours de T_3	-	-	-	$[10-50]$

Tableau 2 : Les caractéristiques des modèles ANFIS utilisées.

Bibliographie

- [1] Aberkane S., 2006. Systèmes Tolérants aux Défauts: Analyse et Synthèse Stochastiques. Thèse de doctorat, Université Henri Poincaré, Nancy1.
 - [2] Abraham A., 2001. Neuro Fuzzy Systems: State-of-the-art Modeling Techniques. School of Computing & Information Technology Press, Australia.
 - [3] Abraham, A., 2005. Adaptation of Fuzzy Inference System Using Neural Learning, Fuzzy System Engineering: Theory and Practice, Nadia Nedjah et al. (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag Germany, ISBN 3-540-25322-X, Chapter 3, pages 53-83.
 - [4] AFNOR. 1994. Maintenance - Concepts et définitions des activités de maintenance. Norme NF X 60-010, Association Française de Normalisation, 28 pages.
 - [5] Ammar M.Y., 2007. Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition batch/continu. Thèse de doctorat. Institut national polytechnique de Toulouse.
 - [6] Andry A.N., Shapiro E.Y., Chung J.C., 1983. Eigen structure assignment for linear systems. IEEE Transactions on Aerospace Electronic Systems.
 - [7] Athamena B., Abbassi H. A., 2003. Fault Detection and Isolation Using Hybrid Parameter Estimation and Fuzzy Logic Residual Evaluation. Informatica. Volume 27. Pages 29–37.
 - [8] Babuska 1998. Fuzzy Modeling for Control, Kluwer Academic Publishers.
 - [9] Benloucif M.L., 2009. Neuro-Fuzzy sensor fault diagnosis of an induction motor. International conference on communication, computer and power (icccp'09). Muscat.
 - [10] Benloucif M.L., Mehennaoui L., Debbache N., 2006. Neuro fuzzy methods for fault diagnosis of nonlinear systems. Asia J. Inform. Technol., volume 5, pages 750-760.
 - [11] Blanke M., Kinnaert M., Lunze J., Staroswiecki M., 2006. Diagnosis and Fault-Tolerant Control. 2^{ème} édition, Springer-Verlag, Berlin.
 - [12] Bodson M., Groszkiewicz J., 2002. Multivariable adaptive control algorithms for reconfigurable flight control. IEEE Transactions on Control Systems Technology. Volume 5. N 2.
 - [13] Boskovic J.D., Li S., et Mehra R. K., 2000. Reconfigurable flight control design using multiple switching controllers and on-line estimation of damage related parameters. IEEE, Conference on Control Application.
 - [14] Boyd S., Ghaoui L., Feron E., 1994. Linear Matrix Inequalities in Systems and Control Theory. SIAM, Philadelphia.
 - [15] Chafaa K., 2006. Structure d'identification et de commande des systèmes non linéaire basées sur les techniques floues. Thèse de doctorat, université de Batna.
 - [16] Ciubotaru B., Staroswiecki M., Christophe C., 2007. Fault tolerant control of the Boeing 747 short-period mode using the admissible model matching technique. 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'06), P.R. China, Beijing.
-

- [17] Claudia V.I.N., 2007. Diagnostic par techniques d'apprentissage floues : conception D'une méthode de validation et d'optimisation des partitions. Thèse doctorat. Institut national des sciences appliquées de Toulouse.
 - [18] Combacau M., 1991. Commande et Surveillance des Systèmes a Évènements Discrets Complexes : Application aux Ateliers Flexibles. Thèse de doctorat, Université Paul Sabatier, Toulouse.
 - [19] Dreyfus G., Samuelides M., Martinez J-M., Gordon M. B., Badran F., Thiria S., Hérault L., 2004. Réseaux de neurones : Méthodologies et applications. 2ème édition, 2-212-11464-8, Édition E.
 - [20] Dubuisson B., 2001. Diagnostic, intelligence artificielle et reconnaissance des formes. Paris, Hermès sciences.- ISBN 2-7462-0249-2.
 - [21] Dunn J.C., 1973. A fuzzy Relative of the ISODATA process and its use in detecting compact well-separated clusters", Journal of cybernetics, volume 3, pages 32-57.
 - [22] Elom A.D., 2006. Diagnostic des Systèmes à Changement de Régime de Fonctionnement. Thèse de Doctorat de l'Institut National Polytechnique de Lorraine, 2006.
 - [23] Eterno J.S., Looze D. P., Weiss J. L., Willsky A.S., 1985. Design issues for fault-tolerant restructurable aircraft control. In: Proc. of the 24th IEEE Conference on Decision and Control, Fort Lauderdale,USA, pages 900-905.
 - [24] Fellouah R., 2007. Contribution au Diagnostic de Pannes pour les Systèmes Différentiellement Plats". Thèse de doctorat de l'Université de Toulouse.
 - [25] Frank P.M., 1990. Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge Based Redundancy- A survey and New Results. Automatica, volume 26, pages 459-474.
 - [26] Gao Z., Antsaklis P.J., 1991. Stability of the pseudo-inverse method for reconfigurable control. International Journal of Control. Volume 53 N 3, pages 717-729.
 - [27] Gao Z., Antsaklis P.J., 1992. Reconfigurable control system design via perfect model following. International Journal of Control. Volume 56 N 4, pages 783-798.
 - [28] Handelman D.A., Stengel R.F., 1989. Combining expert system and analytical redundancy concepts for fault-tolerant flight control". Journal of Guidance, Dynamics and Control. Volume 12. Pages 39-45.
 - [29] Haykin S., 1998. Neural Networks: A Comprehensive Foundation. 2^{ème} édition, New York: Macmillan College Publishing
 - [30] Huzmezan M., 1997. Reconfigurable flight control methods and related issues: a survey. Defense Evaluation and Research Agency (DERA) under research agreement no. ASF/3455.
 - [31] Isermann R., Ballé P., 1996. Trends in the application of model based fault detection and diagnosis of technical process", Proc. of the IFAC, 13th Triennial World Congress, San Francisco, USA. Volume 7, pages 1-12.
 - [32] Jamouli H., 2003. Génération de résidus directionnels pour le diagnostic des systèmes linéaires stochastiques et le commande tolérante aux défauts. Thèse de Doctorat, Spécialité Automatique, Université Henri Poincaré, Nancy1.
 - [33] Jerome C., 2007. Analyse et synthèse d'une architecture coopérative pour la commande tolérante aux défauts. Thèse de doctorat, Spécialité : automatique, productique, signal et image. Université bordeaux I.
-

- [34] Kerrigan E., Maciejowski J., 1999. Fault-tolerant control of a ship propulsion system using model predictive control. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany.
 - [35] Konstantopoulos I., Antsaklis P., 1996. An eigenstructure assignment approach to control reconfiguration. In: Proceedings of 4th IEEE Mediterranean Symposium on Control and Automation. Greece.
 - [36] Linkens M. D. A., 2001. A systematic neuro-fuzzy modeling framework with application to material property prediction. IEEE Trans, Cybern, volume 31, pages 781-790.
 - [37] Lootsma, T. F., 2001. Observer-based Fault Detection and Isolation for Nonlinear Systems. Thèse de PhD. Department of Control Engineering, Aalborg University, Denmark.
 - [38] Maciejowski J.M., 1997. Reconfigurable control using constrained optimization. European control conference, Belgium, pages 107–130.
 - [39] Maciejowski J.M., Jones C., 2003. MPC fault-tolerant flight control case study: Flight 1862". In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pages 121-126.
 - [40] Mahmoud M.M., Jiang J., Zhang Y.M., 2003. Active Fault Tolerant Control Systems: Stochastic Analysis and Synthesis. Springer, Berlin.
 - [41] Marcos A., Balas G. Bokor J., 2005. Integrated FDI and Control for Transport Aircraft. AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco.
 - [42] Marcos A., Balas G., "A robust integrated controller/diagnosis aircraft application". International Journal of Robust and Nonlinear Control 2005, Volume 15. pages. 531-551.
 - [43] Maybecket P.S., Stevens R.D., 1991. Reconfigurable flight control via multiple model adaptive control methods". IEEE Trans. On Aerospace and Electronic Systems, Volume 27.
 - [44] Monnin M., 2004. Diagnostic et pronostic industriel en utilisant les techniques de l'intelligence artificielle. Application a la plate-forme de e-maintenance du LAB, DEA IAP de l'Université de Franche-Comté, Besançon.
 - [45] Nakoula Y, 1997. Apprentissage des modèles linguistique flous, par jeu de règles pondérées. Thèse de doctorat, université de Savoie.
 - [46] Narendra K.S., Parthasarathy K., 1990. Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks. Volume 1. N 1.
 - [47] Nauck D., Kruse R., 1997. What are Neuro-Fuzzy Classifiers? 7th International Fuzzy Systems Association World Congress IFSA'97, Volume 5, pages 228-233, Académie de Prague.
 - [48] Nett C.N., Jacobson C.A., Miller A.T., 1998. An integrated approach to controls and diagnostics: the 4-parameter controller. In Proceedings of Automatic Control Conference, volume 88, pages 824-835.
 - [49] Niemann H., Stoustrup J., 2005. Passive fault tolerant control of a double inverted pendulum - a case study. Control Engineering Practice. Volume 13, pages 1047-1059.
 - [50] Noura H., Sauter D., Hamelin F., et Theilliol D., 2002. Fault tolerant control in dynamic systems: Application to a winding machine. IEEE Control system Magazine. Volume 20 N 1. Pages 33–49.
-

- [51] Palade V., Bocaniala C.D., Jain L., 2006. Computational Intelligence in Fault Diagnosis".Springer-Verlag London.
 - [52] Patan K., 2008. Artificial Neural Networks for the Modeling and Fault Diagnosis of Technical Processes. Berlin, Springer-Verlag. XXII, 206 s. Lecture Notes in Control and Information Sciences, ISSN 0170-8643; 377. ISBN 978-3-540-79871-2.
 - [53] Patrick ripoll, 1999. Conception d'un système de diagnostic flou appliqué au moteur automobile. Thèse de doctorat de l'université de Savoie.
 - [54] Patton R.J., 1997. Fault tolerant control: The 1997 situation (survey). IFAC SAFEPROCESS' 97, Hull, UK, volume 2, pages 1033-1055.
 - [55] Peng Yun, James R.A., 1990. Abductive inference models for diagnostic problem solving. Symbolic Computation. Springer-Verlag. New York.
 - [56] Ploix S., 1998. Diagnostic des systèmes incertains : l'approche bornante. Thèse de l'Institut National Polytechnique de Lorraine
 - [57] Racoceanu D., 2006. Contribution à la surveillance des systèmes de production en utilisant les techniques de l'intelligence artificielle. Habilitation à diriger des recherches, université de Franche Comté de Besançon
 - [58] Reyes C.A.P, 2004. Co-evolutionary fuzzy modeling. In pena reyes, C. A. (ed.). Co-evolutionary fuzzy modeling. LNCS, volume 3204, pages 51-69. Springer, Heidelberg.
 - [59] Rodrigues M., 2005. Diagnostic et commande active tolérante aux défauts appliqués aux systèmes décrits par des multi-modèles linéaires. Thèse de doctorat de l'Université Henri Poincaré, Nancy1.
 - [60] Rodriguez F., Yu W., Moreno-Armendariz M., 2007. Recurrent fuzzy CMAC in hierarchical form for dynamic system identification. In American Control Conference, 9-13 juillet 2007, pages 5706-5711.
 - [61] Ruano A.E., 2005. Intelligent control systems using computational intelligence techniques. The institution of Engineering and Technology, London, united Kingdom.
 - [62] Staroswiecki M., 2003. Fault Tolerant Control. Séminaire Aalborg.
 - [63] Staroswiecki M., 2005. Fault Tolerant Control using an admissible model matching approach. Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, Seville.
 - [64] S-Tehrani, E., Khorasani K., 2009. Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach. Lecture Notes in Control and Information Sciences. Springer.
 - [65] Stoyan K., 2004. Robust Fault-Tolerant Control. Thèse Phd de l'Université Twente, Pays-Bas. ISBN 90-9017903-8.
 - [66] Taleb S., 2007. Génération de symptômes flous par FFT et SWT applications à la détection de vibrations. Institut national polytechnique de Grenoble.
 - [67] Thiaw L., 2008. Identification de systèmes dynamiques non-linéaires par réseaux de neurones et multi-modèles. Thèse de doctorat de l'université Paris XII.
 - [68] Tsui C.C., 1999. A design example with eigenstructure assignment control whose loop transfer function is fully realized. Journal of the Franklin Institute 336, pages 1049-1053.
-

- [69] Tyler M.L., Morari M., 1994. Optimal and robust design in integrated control and diagnosis modules. In proceedings of Automatic control Conference, Baltimore, Maryland, pages. 2060-2064.
 - [70] Uppal F.J., Patton R.J., Palade V., 2002. Neuro-fuzzy based Fault Diagnosis applied to an Electro-Pneumatic Valve. 15^{ème} IFAC World Congress on Automatic Control. Barcelone, Espagne.
 - [71] Vieira J., Morgado D.F., Mota A., 2004. Neuro-Fuzzy Systems: A Survey. 5th WSEAS NNA International Conference on Neural Networks and Applications, Udine, Italia.
 - [72] Weng Z., Patton R., Cui P., 2006. Active fault tolerant control of a double inverted pendulum. 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'06).
 - [73] Yang G.H. & Lum K.Y., 2003. Fault tolerant flight tracking control with stuck faults. American Control Conference, Denver.
 - [74] Yang G.H., Wang J.L., Soh Y.C., 2001. Reliable H_{∞} controller design for linear systems. Automatica, Volume.37, pages 717-725.
 - [75] Yang Z., Izadi-Zamanabadi R., Blanke, M., 2000. On-line multiple-model based adaptive control reconfiguration for a class of nonlinear control systems. IFAC Symposium on fault detection, supervision and safety for technical Processes-Safe process, Budapest.
 - [76] Zadeh L., 1973. Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. on Systems, Man, and Cybernetics, volume1, pages 28-44.
 - [77] Zemouri R., 2003. Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance. Thèse de Doctorat, Université de Franche Comté, Besançon.
 - [78] Zhang Q., 1999. Identification et Surveillance de systèmes Dynamiques. Habilitation à diriger les recherches, Université de Rennes1, Institut de Formation Supérieure en Informatique et en Communication.
 - [79] Zhang Y., Jiang J., 1999. An interacting multiple-model based fault detection, diagnosis and fault tolerant control approach. IEEE Conference on Decision and Control, Phoenix, pages 3593-3598.
 - [80] Zhang Y., Jiang J., 2008. Bibliographical review on reconfigurable fault-tolerant control systems”, Annual Reviews in Control. Volume 32, pages 229-252.
 - [81] Zwingelstein G., 1995. Diagnostic des défaillances - Théorie et pratique pour les systèmes industriels. Traité des Nouvelles Technologies, série Diagnostic et Maintenance, Hermès, Paris.
-

ملخص

التحكم المتحمل للأعطاب يستند على دمج تقنيات التشخيص و التحكم للتكفل الفعال بالأعطاب. تشخيص الأعطاب له أهمية كبرى في الأنظمة الصناعية. الاكتشاف المبكر لتوارد الأعطاب أمر حاسم لتفادي الكوارث البيئية، انخفاض الجودة، انخفاض المردود، وكذلك لضمان سلامة الأشخاص واستمرارية الأجهزة.

إن الهدف من هذا العمل هو طرح إستراتيجية تحكم متحمل للأعطاب تعتمد أساسا على استخدام الشبكات العصبية الاصطناعية وأنظمة الاستدلال الضبابي. هاتان الطريقتان يمكن استعمالهما في الجانب التشخيصي وذلك لتمثيل العمل العادي للظاهرة وتصنيف الأعطاب و في جانب التحكم لاختيار الدور بعد ظهور الأعطاب وانتقاء التحكم المناسب.

قمنا بمحاكاة هذه الإستراتيجية على نظام هيدروليكي/حراري وذلك ل تعويض عن أي أعطاب محتملة من نوع فقدان الفعالية للمنفاذات. وتؤكد النتائج فعالية هذا الأسلوب من التعامل والإلغاء التلقائي للآثار الحتمية المرتبطة بظهور العيوب.

المفاتيح: التحكم المتحمل للأعطاب، الكشف، التشخيص، توليد البواق، النمذجة، الشبكات العصبية الاصطناعية، وأنظمة الاستدلال الغامض.

Résumé

La commande tolérante aux défauts est basée sur l'association de techniques de diagnostic et de commande afin de prendre efficacement en charge les défauts. Le diagnostic des défauts a une grande importance pratique dans les systèmes industriels. La détection précoce de l'occurrence du défaut est critique en évitant les dégâts écologiques, la détérioration de produit, la dégradation de performance et la dégradation importante à l'équipement et la sécurité des personnes, et même la perte de vie.

L'objectif de ce travail est de proposer une stratégie de commande tolérante aux défauts basée sur l'utilisation des réseaux de neurones artificiels et des systèmes d'inférence floue. Ces méthodes peuvent être utilisées dans la partie de diagnostic pour la modélisation du fonctionnement normal du processus et la classification des défauts et dans la partie commande pour le choix des nouveaux objectifs après l'appariation des défauts et la sélection de la commande appropriée.

Une simulation de cette stratégie sur un système hydraulique/thermique a été faite afin de compenser les éventuels défauts de type pertes d'efficacité d'actionneurs. Les résultats obtenus confirment l'efficacité de cette méthode de s'accommoder automatiquement et d'annuler les effets fatals liés à l'appariation des défauts.

Mots clés : Commande tolérante aux défauts, modélisation, génération des résidus, détection, diagnostic, réseaux de neurones artificiels, systèmes d'inférence floue.
