

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Automatique

Spécialité Automatique & systèmes

Présenté par

Badji Imad Eddine & Mezli Mohamed

Étude et conception d'une maquette de TP autour d'un système automatique à trois cuves pour tester les différentes méthodes de diagnostic des systèmes automatiques

✚ Proposé par : Dr Amrouche Badia

✚ Examineur : B.kazed

✚ Président : A.Ferdjouni

Année Universitaire 2019-2020

Remerciements

Nous remercions d'abord ALLAH le tout puissant qui nous a guidés et donnés la force et la volonté de réaliser ce mémoire. Nôtres pensées vont vers nos parents, qui ont toujours cru en nous. C'est grâce à leurs soutient et prières que nous avons accomplis ce travail.

Nous remercions notre promotrice Mme Amrouche Badia de nous 'avoir pris en charge et aidé tout au long du projet. Et de nous avoir orientés avec ces précieux conseils et son soutien moral vu la situation sanitaire liée à la COVID-19.

Nos remerciements les plus sincères à toutes les personnes qui ont de près ou de loin contribué à l'évaluation de ce mémoire. Enfin, nous tenons aussi à remercier les membres du jury pour avoir accepté de juger notre travail.

Dédicaces

Je m'incline devant dieu tout puissant qui m'a ouvert la porte de savoir et m'a aidé à la franchir.

Je dédier ce modeste travail :

A ma chère et tendre mère, source d'affection de courage et d'inspiration qui autant sacrifier pour me voire atteindre ce jour.

A mon père source de respect, en témoignage de ma profonde reconnaissance pour tout l'effort et le soutien incessant qui m'a toujours apporté.

A mon frère : Nour El Islam.

A toutes mes sœurs en particulier ma petite chère Ichrak.

A mes tantes, oncles, cousines et cousins et A toute ma famille et nos proches.

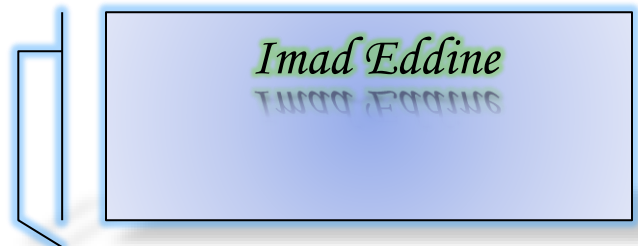
Je suis reconnaissant à mon binôme du projet de fin d'étude Mezli Mohamed pour les moments de découverte et de compréhension communs et pour notre sincère amitié. J'adresse à l'occasion mes salutations à toute la famille Mezli et leurs proches.

Tout le respect et gratitude à tous mes enseignants, généralement du département d'Electronique de l'université de Blida, spécialement ma promotrice Mme B. Amrouche.

A mes amis : Dj. Yaakoub, Ch. Chrif, k. Nassim, G. Omar, B. Moussa, B. Abd El Hamid, L. Ali, S. Ziraoui.

A toute la promotion 2020 sans exception.

Je remercie enfin tous ceux qui m'ont aidé de près ou de loin dans L'accomplissement de ce travail.



Imad Eddine

Dédicaces

Je m'incline devant dieu tout puissant qui m'a ouvert la porte de savoir et m'a aidé à la franchir.

Je dédier ce modeste travail :

A ma chère et tendre mère, source d'affection de courage et d'inspiration qui autant sacrifier pour me voire atteindre ce jour.

A mon père source de respect, en témoignage de ma profonde reconnaissance pour tout l'effort et le soutien incessant qui m'a toujours apporté.

A mes frères en particulier mon petit frère Wassim.

A toutes mes sœurs.

A mes tantes, oncles en particulier mon oncle Wahid, cousines et cousins et A toute ma famille et nos proches.

Je suis reconnaissant à mon binôme du projet de fin d'étude Badji Imad Eddine pour les moments de découverte et de compréhension communs et pour notre sincère amitié. J'adresse à l'occasion mes salutations à toute la famille Badji et leurs proches.

Tout le respect et gratitude à tous mes enseignants, généralement du département d'Electronique de l'université de Blida, spécialement ma promotrice Mme B. Amrouche.

A mes amis : B. Zakaria, B. Marouane, S. Brahim.

A toute la promotion 2020 sans exception.

Je remercie enfin tous ceux qui m'ont aidé de près ou de loin dans L'accomplissement de ce travail.



Mohamed

ملخص :

يقدم هذا العمل أداة تم تطويرها لتلبية حاجة ملحة وهي تركيب العديد من الاعمال التطبيقية لتشخيص الانظمة التلقائية. و هو تطبيق برمجي تم تطويره بواسطة برنامج LabVIEW يسمى Hy.Dia. و هو تطبيق يحاكي ثلاث أنواع من الأعطاب : النظام، المستشعر و المشغل على نظام هيدروليكي ذو ثلاث أحواض. هذه المذكرة تقدم بالتفصيل مراحل تطور هذا التطبيق البرمجي و تفصل شرح النموذج الفعلي الذي يمكن أن يجسد في الواقع و الذي سيكلف 40000 دج.

الكلمات المفتاحية : تشخيص الأنظمة التلقائية، أعمال تطبيقية، نموذج العمل التطبيقي، الحل البرمجي، نظام هيدروليكي ذو ثلاث أحواض، Hy.Dia، أعطاب.

Résumé :

Le présent travail présente un outil développé pour répondre à un besoin urgent qui est le montage de plusieurs TP pour le diagnostic des systèmes automatiques. Il s'agit d'une application software développée sous LabVIEW appelée Hy.Dia. C'est une application permettant de simuler les trois types de défauts : système, capteur et actionneur sur un système hydraulique à trois cuves. Le présent mémoire détaille les étapes de son développement et détaille la maquette réelle qui peut être réalisée à son image et qui coûterait en tout 40.000 DA.

Mots clés : Diagnostic des systèmes automatiques, TP, maquette de TP, solution software, Système hydraulique à trois cuves, Hy.Dia, défauts.

Abstract:

This manuscript presents a tool developed to meet an urgent need, which is the assembly of several TP for automatic systems' diagnosis. It is a software application developed under LabVIEW called Hy.Dia. This software application give the possibility to deal with three types of faults: system, sensor and actuator on a three-tank hydraulic system. This memorandum details the stages of its development and details its prototype description and its production stages. This prototype would cost in all 40,000 DA.

Keywords: Automatic systems' diagnosis of, TP, Prototype, software solution, three-tank hydraulic system, Hy.Dia, faults.

Table des matières

INTRODUCTION GENERALE	1
<u>CHAPITRE I :GENERALITES, CONTEXTE ET PROBLEMATIQUE</u>	3
I.1 INTRODUCTION.....	4
I.2 DEFINITIONS ET CONCEPTS.....	4
I.3 QU'EST-CE QUE LE DIAGNOSTIC DES SYSTEMES AUTOMATIQUES ?	6
I.4 CLASSIFICATION DES METHODES DE DIAGNOSTIC	8
I.5 LE TP DIAGNOSTIC DES SYSTEMES AUTOMATIQUES CONTEXTE ET PROBLEMATIQUE.....	9
I.6 PRESENTATION DE LA SOLUTION RETENUE	10
I.7 PRESENTATION DU SYSTEME HYDRAULIQUE A TROIS CUVES.....	11
I.7.1 Description du système.....	11
I.7.2 Mode de fonctionnement du système :.....	13
I.7.3 Intérêt :	17
I.8 CONCLUSION :	17
<u>CHAPITRE II : ADAPTATION DU SYSTEME HYDRAULIQUE A TROIS CUVES</u>	
<u>AUX BESOINS DU TP</u>	19
II.1 INTRODUCTION.....	20
II.2 PRESENTATION DE LA REFLEXION AUTOUR DE LA MAQUETTE	20
II.3 LE SYSTEME HYDRAULIQUE A TROIS CUVES.....	21
II.3.1 Les cuves communicantes :.....	21
II.3.2 La partie électrique :.....	22
II.3.2.1 Composants électrique utilisé :.....	22
II.3.2.2 Identification des composants et ses fonctionnements :	22
II.3.3 L'informatique associée :	25

II.3.3.1	Sur le plan matériel :	25
II.3.4	<i>Les critères de choix des composants</i> :	26
II.4	LA COMMANDE DU SYSTEME PAR PID	26
II.4.1	<i>Systèmes de commande en automatique</i> :	26
II.4.1.1	Système de Commande en boucle ouverte :	27
II.4.1.2	Système de commande en boucle fermée :	27
II.4.2	<i>La commande PID</i> :	29
II.4.2.1	Influence des paramètres PID	31
II.4.3	<i>La commande PID dans le système à trois cuves</i> :	31
II.5	ESTIMATION FINANCIERE DE LA MAQUETTE DE TP	33
II.6	LES DEFAUTS DANS LA MAQUETTE :	34
II.6.1	<i>Modèles des défauts</i> :	34
II.6.2	<i>Différentes origines de défaut</i> :	34
II.6.2.1	Défaut sur le capteur :	35
II.6.2.2	Défaut sur l'actionneur :	35
II.6.2.3	Défaut sur le système ou composants :	35
II.6.3	<i>Type de défauts</i> :	35
II.6.4	<i>Classification des différents défauts dans la maquette</i> :	36
II.7	LE LOGICIEL DE PROGRAMMATION	38
II.8	CONCLUSION :	39
CHAPITRE III :	<u>PRESENTATION DU LOGICIEL LABVIEW</u>	40
III.1	INTRODUCTION	41
III.2	ENVIRONNEMENT LABVIEW	41
III.3	INSTRUMENTS VIRTUELS (VI)	42
III.3.1	<i>Face avant</i> :	43
III.3.2	<i>Face arrière (Block Diagram)</i> :	44
III.4	BARRES D'OUTILS ET PALETTES	45
III.4.1	<i>Barre d'outils de la face avant</i> :	45
III.4.2	<i>Barre d'outils du diagramme</i> :	46
III.5	LES PALETTES LABVIEW	46

III.6 LE CHOIX DE LABVIEW	48
III.7 COMPARAISON ENTRE LABVIEW ET D'AUTRES LOGICIELS	49
<i>III.7.1 Quelles sont les différences entre LabVIEW et MATLAB ?</i>	<i>49</i>
<i>III.7.2 Quelles sont les différences entre LabVIEW et C ?.....</i>	<i>51</i>
<i>III.7.3 Quelles sont les différences entre LabVIEW et WinCC ?.....</i>	<i>51</i>
III.8 CONCLUSION.....	52
<u>CHAPITRE IV : PRESENTATION DE L'APPLICATION SOFTWARE « HY.DIA » ..</u>	<u>53</u>
IV.1 INTRODUCTION.....	54
IV.2 PRESENTATION GENERALE DE HY.DIA :.....	54
IV.3 ETAPES DE DEVELOPPEMENT	59
<i>IV.3.1 Système à un seul réservoir :.....</i>	<i>59</i>
IV.4 LES ELEMENTS DE L'INTERFACE ET SES CALLS BACK :.....	63
IV.5 LA COMMUNICATION ENTRE L'INTERFACE ET LES PROGRAMMES.....	67
IV.6 FONCTION DE HY.DIA, EXEMPLES COMMENTE	72
<i>IV.6.1 La commande, suivi et changement de la consigne :</i>	<i>72</i>
<i>IV.6.2 Les défauts dans l'interface :</i>	<i>73</i>
IV.6.2.1 Les défauts dans le réservoir 1 :.....	73
IV.6.2.2 Les défauts dans le réservoir 2 :.....	77
IV.7 CONCLUSION :.....	82
CONCLUSION GENERALE	83
BIBLIOGRAPHIE.....	85
WEBOGRAPHIE	88

Abréviations

- **API** : Automate Programmable Industriel.
- **CAN** : Convertisseur Analogique Numérique.
- **CNA** : Convertisseur Numérique Analogique.
- **DSC**: Datalogging and Supervisory Control Module.
- **FDI** : Fault Detection and Isolation.
- **HMI** : Interface de Machine Humaine.
- **NI**: National Instruments.

Symboles

- **A** : section d'une cuve.
- **a** : surface de la valve.
- **C(s)** : La fonction de transfert de régulateur.
- **C_i** : Capteur de la hauteur d'eau dans le réservoir.
- **E(t)** : L'erreur.
- **g** : Le coefficient de gravité.
- **G(s)** : La fonction de transfert de procédé.
- **H_{BF}(s)** : La fonction de transfert en boucle fermée
- **h_i** : La hauteur d'eau dans le réservoir *i*.
- **K_p** : Le gain de l'action proportionnelle.
- **P1(t), P2(t)** : Perturbations à l'entrée et à la sortie
- **P_i** : La pompe d'alimentation du réservoir *i*.
- **PID** : Correcteur Proportionnel Intégral Dérivée.
- **q_{fi}** : Le débit de fuite dans les vannes *V_{fi}*.
- **q_{i(t)}** : Le débit de la *i*ème pompe.
- **q_{ij}** : Le débit circulant dans la vanne de communication *V_{ij}*.

- q_{in} : Le débit entrant.
- q_{out} : Le débit sortant.
- $R(t)$: La consigne.
- S : La section des cuves de la station hydraulique.
- S_c : Section de vannes de communication.
- τ : Le constant de temps.
- T_0 : Le réservoir tampon.
- T_d : Le constant de dérivation.
- T_i : Le constant d'intégration.
- T_i : Les réservoirs i de la station.
- $U(t)$: La commande.
- U : Vecteur de commande (entrées).
- V_{ci} : Tension issue du $i^{\text{ème}}$ capteur de niveau.
- V_{fi} : Les vannes de fuites pour l' $i^{\text{ème}}$ réservoir.
- V_{ij} : Vanne de communication entre le réservoir i et réservoir j .
- V_{pi} : Tension appliquée à la $i^{\text{ème}}$ pompe d'alimentation.
- ΔV : volume du liquide accumulé
- Y : Vecteur de sortie.
- $Y(t)$: La sortie mesurée de procédé.
- μ_{ij} : Coefficient de viscosité de la vanne de communication entre le réservoir i et le réservoir j .

Liste des figures

<i>Figure I.1</i> : principe général du diagnostic.	7
<i>Figure I.2</i> : Principe du diagnostic des systèmes commandés.....	7
<i>Figure I.3</i> : Une classification générale des méthodes de diagnostic.....	9
<i>Figure I.4</i> : station hydraulique à trois cuves.	11
<i>Figure I.5</i> : Schéma synoptique du système hydraulique.	12
<i>Figure II.1</i> : Schéma synoptique de la mise en œuvre pratique.	22
<i>Figure II.2</i> : schéma de fonctionnement d'une pompe hydraulique	23
<i>Figure II.3</i> : symbole de la pompe à débit variable et à un sens de flux.	24
<i>Figure II.4</i> : la carte National Instrument PCE-6024E.....	26
<i>Figure II.5</i> : structure de système de commande en boucle ouverte.	27
<i>Figure II.6</i> : structure de système de commande en boucle fermée.	28
<i>Figure II.7</i> : l'effet de chaque action du correcteur PID pour un même procédé.	29
<i>Figure II.8</i> : les trois types de régulateur PID.	31
<i>Figure II.9</i> : Les trois types de défauts d'un procédé.....	35
<i>Figure II.10</i> : Evolution temporelle des différents types de défauts.....	36
<i>Figure II.11</i> : organigramme de détection de défaut.	38
<i>Figure III.1</i> : Fenêtre de démarrage du logiciel LabVIEW.....	41
<i>Figure III.2</i> : Composants d'un VI.	43
<i>Figure III.3</i> : VI en face avant.....	44
<i>Figure III.4</i> : VI Face arrière.	45
<i>Figure III.5</i> : Barre d'outils face avant.	45
<i>Figure III.6</i> : Barre d'outils face arrière.	46
<i>Figure III.7</i> : La palette d'outils.....	46
<i>Figure III.8</i> : La palette commande.	47
<i>Figure III.9</i> : La palette fonctions.....	48
<i>Figure IV.1</i> : l'interface de simulation de la maquette.....	56
<i>Figure IV.2</i> : Organigramme : résumé l'utilisation de l'interface.....	57

Figure IV.3 : schéma du réservoir 1.....	60
Figure IV.4 : le schéma fonctionnel d'un niveau de réservoir avec un débit d'entrée et un débit de sortie.....	61
Figure IV.5 : schéma fonctionnel du système de commande.....	62
Figure IV.6 : Les possibilités de l'interface.....	64
Figure IV.7 le programme correspond à un seul réservoir.....	71
Figure IV.8 : Suivi de consigne pour le niveau du réservoir 1 et 2.....	72
Figure IV.9 : fuite dans le réservoir 1.....	74
Figure IV.10 : La réponse à un bouchage dans le réservoir 1.....	75
Figure IV.11 : La réponse à un défaut capteur dans le réservoir 1.	76
Figure IV.12 : La réponse à un défaut sur la pompe dans le réservoir 1.	77
Figure IV.13 : La réponse à une fuite dans le réservoir 2.	78
Figure IV.14 : La réponse à un bouchage dans le réservoir 2.....	79
Figure IV.15 : La réponse à un défaut capteur dans le réservoir 2.	80
Figure IV.16 : La réponse à un défaut sur la pompe dans le réservoir 2.	81
Figure IV.17 : Exemple de capture d'écran illustrant le fonctionnement de l'interface.....	81

Liste des tableaux

<i>Tableau I.1 : Paramètres du système des trois cuves.</i>	<i>17</i>
<i>Tableau II.2 : Effets du réglage indépendant des proportions, intégral et dérivé (PID).</i>	<i>33</i>
<i>Tableau II.1 : budget de la maquette.....</i>	<i>34</i>
<i>Tableau II.3 : types et origines des défauts sur la maquette.</i>	<i>37</i>
<i>Tableau IV.1 : Les éléments de l'interfaces et ses calls back.</i>	<i>66</i>
<i>Tableau IV.2 : Eléments de l'interface et leurs équivalents dans le programme.</i>	<i>70</i>

Introduction Générale

Le diagnostic des systèmes automatique est un axe de l'automatique relativement récent. L'enjeu est de surveiller un système automatique en exploitant ses variables et ses paramètres pour identifier les anomalies et les défauts en temps réel afin d'éviter l'arrêt du système, de préserver la sécurité du système, des opérateurs humains et de l'environnement aussi.

Dans la formation Master en automatique des systèmes électriques au niveau du département d'électronique de l'université Blida1, le diagnostic des systèmes est enseigné via deux unités d'enseignement :

- 1) Diagnostic des systèmes automatiques (cours et ses séances de TD), cette unité vise à présenter les concepts généraux, les définitions et les généralités sur le diagnostic des systèmes automatiques et apprendre aux étudiants comment assurer la surveillance et le diagnostic des systèmes automatiques. Les méthodes de diagnostic sont ainsi présentées et expliquées. Les séances de travaux dirigées sont, quant à elles, centrées sur les exercices et les applications numériques des différents concepts introduits dans les cours.
- 2) Le TP diagnostic des systèmes automatiques : cette unité d'enseignement est sensée appuyée la première unité d'enseignement via des TP conçue pour les différentes parties du cours. A savoir les différentes méthodes de calcul des résidus et le traitement de ces résidus pour la détection puis l'identification des différents défauts.

Cependant, par manque de moyens, le TP sous sa forme actuelle, se fait sous forme de simulation, ce qui néglige pas mal d'aspects liés au diagnostic des systèmes automatiques tels que la difficulté des systèmes, l'imprévisibilité des défauts, leurs types, et les contraintes liées à la fiabilité des informations utilisées pour le diagnostic. Ainsi, un besoin de développer un outil interactif qui permet de gérer, de découvrir toutes ces facettes et difficultés du diagnostic s'est fait sentir. Un tel outil permettrait aux étudiants qui suivent ce cours de mieux comprendre l'enjeu du diagnostic des systèmes automatiques et de bien maitriser les

différentes méthodes. Le défi a été donc de développer une maquette de TP qui permet de remplir les conditions suivantes :

- Illustrer un système automatique ;
- Avec un niveau de complexité moyen ou plus ;
- Comportant plusieurs parties susceptibles de présenter et de simuler des défauts ;
- Ces défauts peuvent être modifiés en types et intensité ;
- L'occurrence des défauts peut être programmée ;
- L'outil va être sous forme d'une application software graphique facilitant l'interaction des étudiants ;
- La maquette physique est étudiée et conçue à l'image de l'application software avec une évaluation économique pour permettre sa réalisation dans un travail future.

Pour concrétiser ces objectifs un système hydraulique automatique à trois bacs a été retenu. Ce système a été analysé et une maquette de TP développée autour de lui de façon à pouvoir simuler et programmer différents types de défauts autour desquels plusieurs TP peuvent être montés. Le développement de ce projet est décrit à travers les différents chapitres du présent mémoire dont l'organisation peut être décrite comme suit :

Chapitre 1 : Présente des concepts généraux sur le diagnostic. Il sera notamment consacré aux concepts fondamentaux du diagnostic des systèmes à bases de modèles et aussi présente la description du système hydraulique à 3 cuves, le décrire, décrire son fonctionnement, ses constituants et son intérêt pour l'enseignement en automatique.

Chapitre 2 : Dans ce chapitre on analyse la possibilité de conception de la maquette autour de ce système à trois bacs en accord avec le dispositif nécessaire avec la commande de système.

Chapitre 3 : Ce chapitre présente l'outil de programmation graphique LabVIEW à caractère universel bien adapté pour la mesure et sa visualisation.

Chapitre 4 : Ce chapitre sera consacré au développement de la maquette sous LabVIEW, on présente tout ce qui concerne l'application, son principe, ses composants, les défauts envisagés, les possibilités qu'elles offrent et la simulation des défauts appliqué sur ce système.

Ce mémoire se termine par une conclusion générale et des perspectives.

Chapitre I : Généralités, contexte et problématique

I.1 Introduction

Le diagnostic des systèmes s'est imposé comme étant axe à part entière de l'automatisme. Il répond à un besoin accru de sécurité et de performance des systèmes automatiques qui consiste à détecter très tôt les défauts et les identifier afin d'éviter des catastrophes qui pourraient coûter cher dans des domaines assez sensibles que l'aéronautique, le nucléaire. Le diagnostic des systèmes automatiques permet aussi de gagner du temps et de l'argent.

De par son importance, le diagnostic fait partie de la formation des étudiants qui suivent une formation de master automatique dans le département d'électronique au niveau de l'université Blida1. Ce chapitre présente les concepts de base de diagnostic et de la surveillance des systèmes automatisés. Les termes, les méthodes ainsi que le besoin en TP de l'unité d'enseignement TP de diagnostic des systèmes automatiques.

I.2 Définitions et concepts

Le domaine du diagnostic des systèmes automatiques utilise une terminologie bien précise. Nous avons retenu les notions qui ont un lien direct avec notre travail et sa compréhension :

❖ ***Perturbation :***

Une perturbation est un signal d'entrée non contrôlé dont la présence est non souhaitable mais considérée comme normale [1].

❖ ***Défaut (fault) :***

Un défaut est généralement défini comme une déviation non-autorisée d'au moins une propriété caractéristique ou d'une variable ou d'un écart du comportement du système lui-même, des capteurs, des actionneurs, ou de tout autre équipement. Il peut être modélisé comme une entrée externe, ou sous forme d'écart de paramètre qui modifie les

caractéristiques du système. Les incertitudes et les perturbations sont modélisées de manière assez similaire aux défauts comme une déviation paramétrique, ou une entrée externe [2].

❖ **Défaillance (failure) :**

Elle peut être une conséquence d'un défaut. Une défaillance correspond à une interruption permanente de la capacité d'un système à effectuer les fonctions attendues dans les conditions de fonctionnement nominales [2].

❖ **Panne (break-down) :**

Elle représente les conséquences d'une défaillance dans la réalisation d'un fonctionnement nominal d'un processus. En d'autres termes, c'est l'incapacité permanente de la capacité du système à remplir sa fonction requise dans des conditions d'opérations spécifiées [1].

❖ **Résidu (Residual) :**

Indicateur de présence ou non d'un défaut. C'est la différence entre le comportement du système observé et le comportement attendu à l'aide d'un modèle de référence [2].

❖ **Seuil (Threshold) :**

C'est la valeur limite de l'écart d'un résidu avec le zéro, ainsi s'il est dépassé, un défaut est déclaré comme détecté [3].

❖ **Symptôme (Symptom) :**

Événement ou ensemble de données au travers duquel le système de détection identifie le passage du procédé dans un fonctionnement anormal [4].

❖ **Détection de défaut (fault detection) :**

Cette fonction détermine la présence ou non d'un défaut ainsi que l'instant de l'apparition [5].

❖ *Localisation de défauts*

Il s'agit de localiser le sous-système affecté par le défaut détecté, responsable de la défaillance du système. La localisation consiste, en effet, à remonter les symptômes pour retrouver l'ensemble des éléments défaillants [5].

❖ *Identification de défaut (fault identification)*

Cette fonction estime l'amplitude et l'évolution temporelle du défaut afin d'expliquer au mieux le comportement du système. Fonction qui suit la localisation [5].

❖ *Diagnostic de défauts (fault diagnosis)*

De manière générale, une procédure de diagnostic détermine le type, l'amplitude, la localisation et l'instant d'apparition du défaut [5].

I.3 Qu'est-ce que le diagnostic des systèmes automatiques ?

Un diagnostic est une procédure permettant de détecter un composant ou un élément défectueux et de le localiser dans un système dynamique. On désigne par détection la capacité à mettre en évidence l'apparition d'un ou plusieurs défauts. On parle de localisation quand on est, de plus, capable de préciser la nature du ou des défauts occurrents. La **figure (I.1)** montre la structure générale de la procédure de diagnostic dans laquelle le module de diagnostic est alimenté par toutes les connaissances disponibles (observation et modèle) sur le système. Cette connaissance inclut les mesures des variables et toute autre information pouvant être utile pour le diagnostic comme la structure du système. Le module de diagnostic traite les observations et produit un "diagnostic" qui est une liste de défauts possibles pouvant affecter le système au fil du temps [6].

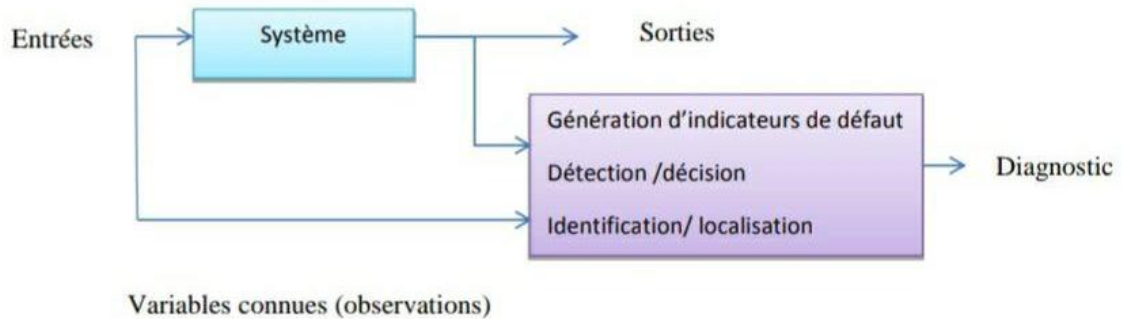


Figure I.1 : principe général du diagnostic [23].

Fréquemment, le système est régulé par un contrôleur dans le but d'améliorer ses performances. Dans ce cas, les variables connues sont les sorties du contrôleur et les mesures de sorties fournies par les capteurs. Ce cas est illustré par la **figure (I.2)** qui représente une complexité fondamentale pour la synthèse du module de diagnostic due à la présence non seulement des défauts mais aussi de perturbations. Ces deux types d'entrées non contrôlées et généralement non mesurables affectent l'évolution du système et dégradent ses performances. Les perturbations appelées aussi entrées inconnues, ne sont pas considérées comme des défauts mais influencent aussi l'évolution du système [6].

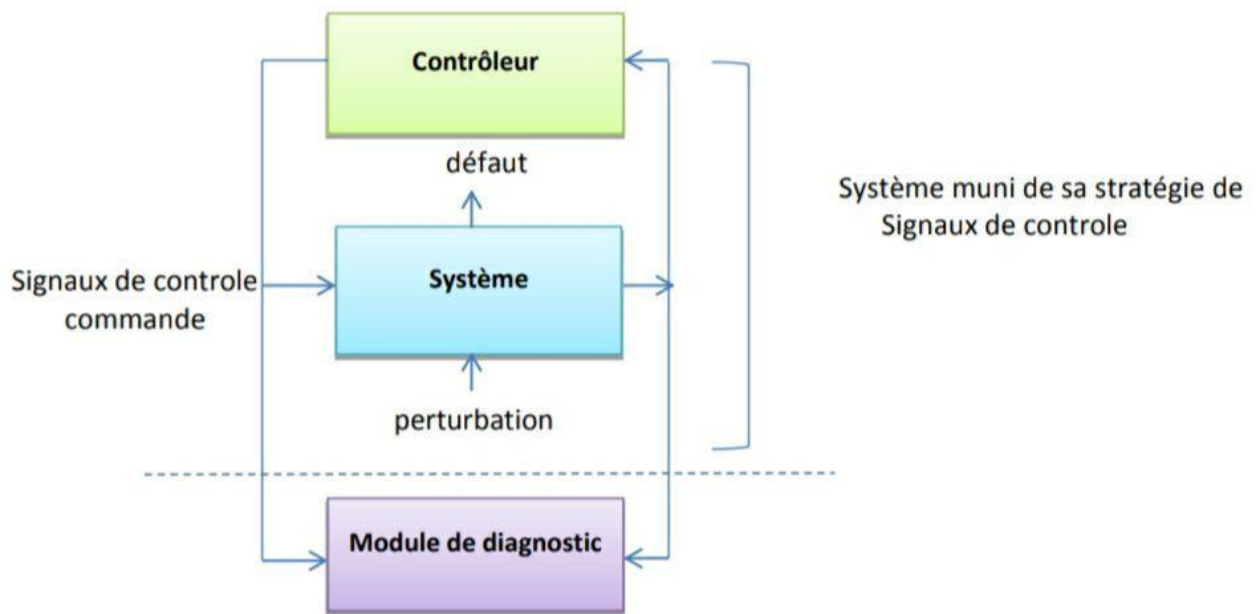


Figure I.2 : Principe du diagnostic des systèmes commandés [23].

Le module de diagnostic doit distinguer de ce fait l'influence provoquée par ces entrées inconnues et celle causée par les défauts [6].

I.4 Classification des méthodes de diagnostic

Les méthodes de diagnostic varient selon différents critères : la dynamique de procédé (discret, continu ou hybride), sa complexité, l'implémentation de diagnostic en ligne et/ou hors ligne, la nature d'information (qualitative et/ou quantitative), sa profondeur (structurelle, fonctionnelle et/ou temporelle), sa distribution (centralisée, décentralisée ou distribuée) ...etc. Dans ce contexte, plusieurs classifications sont proposées dans la littérature [11][12][13][14]. Ces classifications sont influencées par les terminologies et les contextes particuliers de chaque communauté et ne sont pas toujours homogènes. Nous présentons une classification non exhaustive des méthodes de diagnostic en deux grandes familles (**figure (I.4)**) :

- Les méthodes sans modèle mathématique qui ne nécessitent pas de connaissances accrues du système physique, mais des connaissances superficielles.
- Méthodes fondées sur des modèles qui exigent qui nécessitent une connaissance approfondie du système physique.

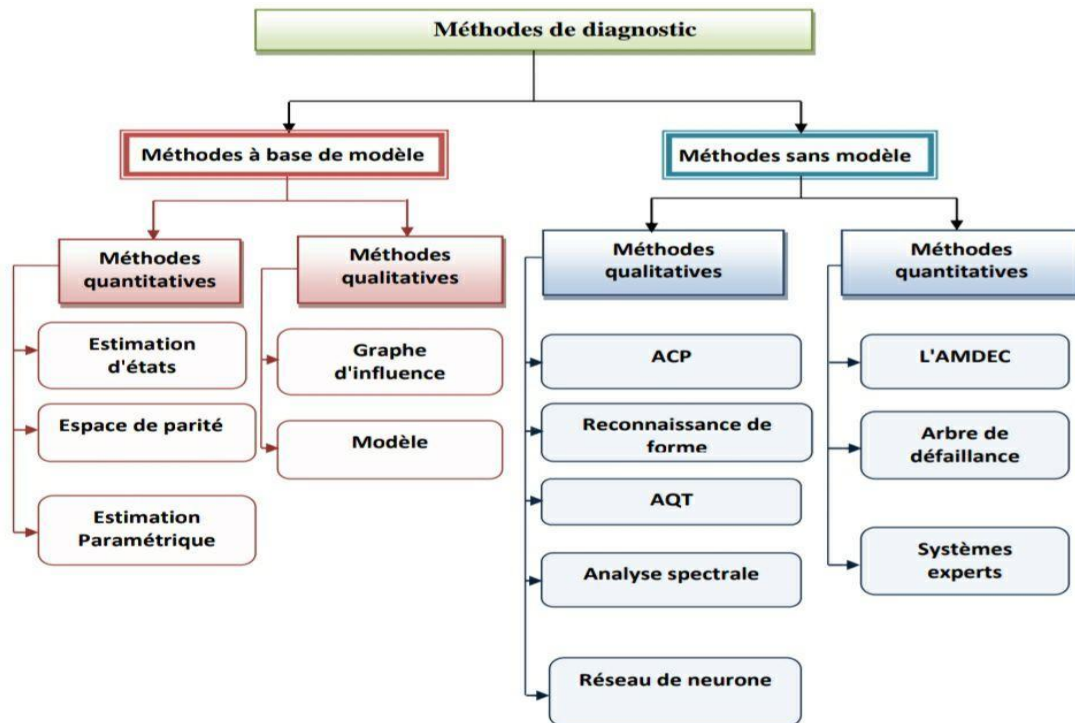


Figure I.3 : Une classification générale des méthodes de diagnostic [23].

I.5 Le TP diagnostic des systèmes automatiques contexte et problématique

L'unité d'enseignement « TP de diagnostic des systèmes automatiques » est constitué de plusieurs séances de TP qui devraient permettre aux étudiants d'expérimenter les différentes notions apprises dans l'unité d'enseignement « Diagnostic des systèmes automatiques ». Il s'agit principalement de trois méthodes de calcul des résidus à savoir 1) la redondance analytique, 2) les observateurs d'état, 3) la méthode paramétrique et en dernier un TP monté autour de la notion de traitement des résidus afin de poser le diagnostic approprié.

Afin, que cette unité puisse remplir toutes ses objectifs, il est indispensable de disposer d'un outil de travail permettant d'offrir aux étudiants la possibilité de percevoir l'impact des défauts sur un systèmes automatique et de tester les différentes notions apprises durant le

cours. Et pour ce, il est nécessaire d'avoir un outil de travail qui permet de simuler les défauts afin de mettre les étudiants en situation de défis pour les détecter et les identifier. D'où le souhait du chargé de cette unité d'enseignement de développer un outil de travail qui serait une application software la plus fidèle possible à un système automatique présentant des défauts, qui pourrait par la suite réalisée pour avoir une maquette réelle avec ce même système physique pour permettre aux étudiants d'expérimenter et d'apprendre dans des conditions réelles. Ainsi, cet outil de travail doit répondre à certaines exigences qui sont :

- Illustrer un système automatique ;
- Avec un niveau de complexité moyen ou plus ;
- Comportant plusieurs parties susceptibles de présenter et de simuler des défauts ;
- Ces défauts peuvent être modifiés en types et intensité ;
- L'occurrence des défauts peut être programmée ;
- L'outil va être sous forme d'une application software graphique facilitant l'interaction des étudiants ;
- La maquette physique est étudiée et conçue à l'image de l'application software avec une évaluation économique pour permettre sa réalisation dans un travail future.

I.6 Présentation de la solution retenue

L'un des systèmes automatiques qui regroupent toutes les critères didactiques est sont doute le système hydraulique à plusieurs bacs. En effet, ce système présente les avantages suivants :

- 1) C'est un système automatique ;
- 2) Il est assez complexe et sa complexité est modulable par augmentation de nombre de bac ;
- 3) Son principe de fonctionnement est simple à comprendre puisqu'il s'agit du domaine de l'hydraulique associant réservoirs, tuyaux et pompes ;
- 4) Il est très documenté en littérature ;

- 5) Il s'apprête bien à plusieurs volets de l'automatique telle que la modélisation et la commande ;
- 6) C'est un système visuel qui offre plusieurs possibilités pour le développement d'une interface graphique tant en illustration qu'en animation.
- 7)



Figure I.4 : station hydraulique à trois cuves [15].

I.7 Présentation du système hydraulique à trois cuves

I.7.1 Description du système

Le système hydraulique représenté sur la **figure (I.5)**, est composé de trois réservoirs cylindriques verticaux identiques de section S numérotés de gauche à droite par réservoir 1 noté T_1 , réservoir 3 noté T_3 et réservoir 2 noté T_2 , et d'un réservoir tampon noté T_0 , qui constitue le socle de ce dispositif.

Le liquide s'écoulant de ces cuves est recueilli dans le réservoir de rétention T_0 qui approvisionne deux pompes P_1 et P_2 commandées par des moteurs à courant continu. Ces deux pompes alimentent les réservoirs 1 et 2 avec les débits $q_1(t)$ et $q_2(t)$.

Les trois réservoirs sont reliés entre eux par des conduites de communication de sections identiques S_c . Ces derniers sont équipés d'un système de vannes à billes v_{13} et v_{32} manuellement réglables. L'écoulement de l'eau en dehors des réservoirs cylindriques vers le réservoir tampon est assuré par la sortie contrôlée par la vanne v_{20} à l'extrémité du réservoir T_2 .

En outre, le système est équipé de vannes v_{f1} , v_{f2} et v_{f3} appelés vannes de fuites dont le rôle est de simuler des fuites dans les réservoirs correspondants. Ces vannes de fuites peuvent être utilisées pour introduire des perturbations ou produire des défauts. Les six vannes peuvent être utilisées pour changer la configuration du processus qui nous permet de travailler avec un dispositif composé d'un, deux ou trois réservoirs.

Par ailleurs, Les cuves sont équipées de capteurs de pression piézo-résistifs pour mesurer les niveaux du liquide dans T_1 , T_2 et T_3 , respectivement notés $h_1(t)$, $h_2(t)$ et $h_3(t)$ [15].

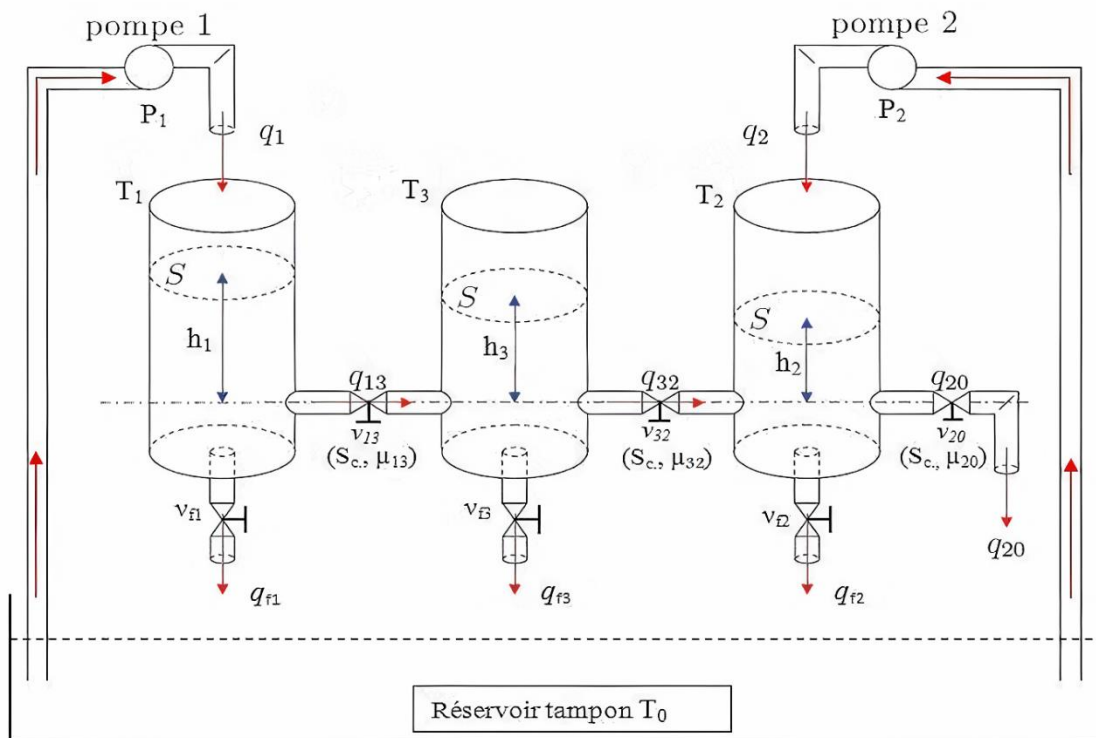


Figure I.5 : Schéma synoptique du système hydraulique [15].

Avec :

T_0 : Réservoir tampon.

T_i : Réservoir i .

P_i : Pompe d'alimentation du réservoir i .

h_i : Hauteur d'eau dans le réservoir i [cm].

v_{fi} : Vannes de fuite du réservoir i vers le réservoir tampon.

v_{ij} : Vannes de communication entre le réservoir i et le réservoir j .

μ_{ij} : coefficient de viscosité de la vanne de communication entre le réservoir i et le réservoir j .

q_i : Débit de la pompe i .

q_{fi} : Débit de fuite dans les vannes v_{fi} .

q_{ij} : Débit circulant dans la vanne de communication v_{ij} .

S : Section du Réservoir cylindrique [m^2].

S_c : Section de vannes de communication et des vannes de fuite [m^2].

I.7.2 Mode de fonctionnement du système :

Dans notre application, nous avons choisi de configurer le processus indiqué dans la **figure (I.5)**. Ce dernier nous permet de travailler avec un dispositif composé de trois réservoirs, et le débit d'eau des réservoirs de cylindriques au réservoir de tampon sera garanti par la sortie à l'extrémité du réservoir T2, qui nécessite l'ouverture complète des vannes d'appel V13, V32 et V20, et la fermeture complète des vannes de fuite V_{F1} , V_{F2} et V_{F3} .

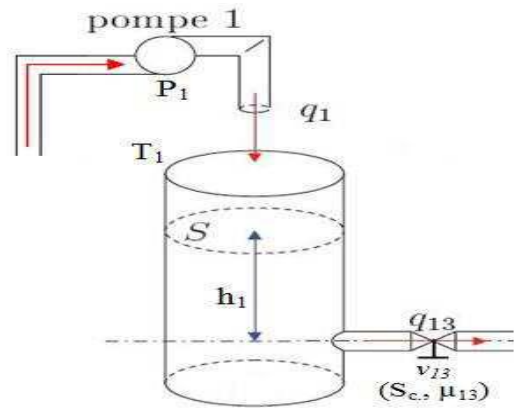
Les trois réservoirs sont des cylindres de révolution, de section $S = 0.0154 \text{ m}^2$. Selon la loi fondamentale de conservation de la matière, la variation de volume d'eau emmagasiné par

unité de temps par un réservoir est donnée par la différence entre le débit entrant au réservoir et le débit sortant de ce dernier, soit :

$$\left[\begin{array}{l} \text{l'accumulation de l'eau} \\ \text{dans le reservoir " i " } \end{array} \right] = \left[\begin{array}{l} \text{débit d'eau entrant} \\ \text{au réservoir" i " } \end{array} \right] - \left[\begin{array}{l} \text{débit d'eau sortant} \\ \text{du réservoir" i " } \end{array} \right]$$

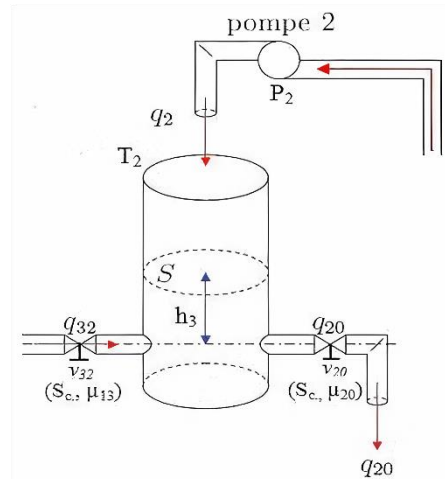
- Le bilan de masse dans le réservoir 1 donne la relation suivante :

$$S \cdot \frac{dh_1}{dt} = q_1 - q_{13} \quad (I.1)$$



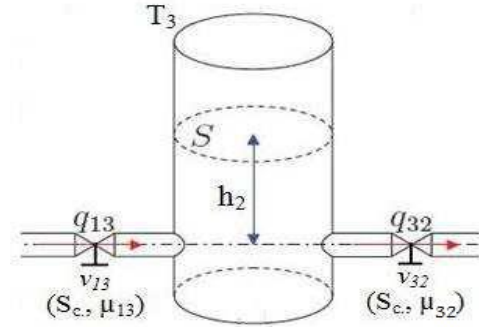
- Le bilan de masse dans le réservoir 2 donne la relation suivante :

$$S \cdot \frac{dh_2}{dt} = q_2 + q_{32} - q_{20} \quad (I.2)$$



- Le bilan de masse dans le réservoir 3 donne la relation suivante :

$$S \cdot \frac{dh_3}{dt} = q_{13} - q_{32} \quad (I.3)$$



En explicitant le fonctionnement de chaque réservoir, on obtient alors un modèle analytique du système représenté par trois équations différentielles non linéaires de premier ordre selon les équations d'état suivantes :

$$\sum NL: \begin{cases} S \cdot \frac{dh_1}{dt} = q_1 - q_{13} \\ S \cdot \frac{dh_2}{dt} = q_2 + q_{32} - q_{20} \\ S \cdot \frac{dh_3}{dt} = q_{13} - q_{32} \end{cases} \quad (I.4)$$

Où $q_{ij}(t)$ représente le débit de liquide de la $i^{\text{ème}}$ cuve vers la $j^{\text{ème}}$ ($i, j=1,2,3 \forall i \neq j$), qui peut être exprimé en utilisant la loi de TORRICILI par :

$$q_{ij}(t) = \mu_{ij} \cdot s_c \cdot \text{sign}(h_i(t) - h_j(t)) \cdot \sqrt{2g|h_i(t) - h_j(t)|} \quad (I.5)$$

Et q_{20} représente le débit de sortie, avec :

$$q_{20}(t) = \mu_{20} \cdot s_c \cdot \sqrt{2g \cdot h_2(t)} \quad (I.6)$$

Sans restreindre notre étude, nous considérons le système tel que les niveaux vérifient les inégalités suivantes : $h_1 > h_3 > h_2$. En d'autres termes nous considérons un sens particulier des débits inter-cuves($q_{ij}(t)$).

$$\begin{cases} q_{13} = s_c \cdot \mu_{13} \cdot \text{sign}(h_1 - h_3) \cdot \sqrt{h_1 - h_3} \\ q_{32} = s_c \cdot \mu_{32} \cdot \text{sign}(h_3 - h_2) \cdot \sqrt{h_3 - h_2} \\ q_{20} = s_c \cdot u_{20} \cdot \sqrt{h_2} \end{cases} \quad (I.7)$$

En remplaçant les équations (I.7) dans (I.4), nous obtenons le modèle mathématique de ce procédé et qui est donné par les équations suivantes :

$$\begin{cases} \dot{h}_1 = \frac{1}{S} [q_1 - C_{13} \cdot \sqrt{h_1 - h_3}] \\ \dot{h}_2 = \frac{1}{S} [q_2 + (C_{32} \cdot \sqrt{h_3 - h_2}) - (C_{20} \cdot \sqrt{h_2})] \\ \dot{h}_3 = \frac{1}{S} [(C_{13} \cdot \sqrt{h_1 - h_3}) - (C_{32} \sqrt{h_3 - h_2})] \end{cases} \quad (I.8)$$

L'équation (I.8) peut se réécrire dans l'espace d'état de la façon suivante :

$$\begin{cases} \dot{x}_1 = \frac{1}{S} [u_1 - C_{13} \sqrt{x_1 - x_3}] \\ \dot{x}_2 = \frac{1}{S} [u_2 + (C_{32} \cdot \sqrt{x_3 - x_2}) - (C_{20} \cdot \sqrt{x_2})] \\ \dot{x}_3 = \frac{1}{S} [(C_{13} \cdot \sqrt{x_1 - x_3}) - (C_{32} \sqrt{x_3 - x_2})] \end{cases} \quad (I.9)$$

$$\begin{cases} y_1 = x_1 \\ y_2 = x_2 \end{cases} \quad (I.10)$$

Où $x = [x_1, x_2, x_3]^T = [h_1, h_2, h_3]^T$ représente le vecteur d'état et $u = [u_1, u_2]^T = [q_1, q_2]^T$ est le vecteur de commande. Les sorties à commander sont données par $y_1(t)$ et $y_2(t)$.

Les paramètres C13, C32 et C20 sont déterminés à partir des mesures expérimentales.

Variable	Symbole	Valeur
Section de la cuve	S	0.0154 m ²

Coefficients	$C_{13} = C_{32}$	0.9×10^{-4}
	C_{20}	2×10^{-4}
Débit maximal	$q_{1 \max}$	$7.2079 \times 10^{-5} \text{ m}^3/\text{s}$
	$q_{2 \max}$	$7.2738 \times 10^{-5} \text{ m}^3/\text{s}$
Niveau maximal	$h_{i \max} (i=1, 2, 3)$	0.60 m

Tableau I.1 : Paramètres du système des trois cuves [15].

I.7.3 Intérêt :

Dans le domaine de l'industrie et de la production industrielle le champ de la compétitivité repose sur plusieurs critères de performance, de robustesse et de qualité. Mais malgré toutes les assiduités pour l'établissement de ces conceptions, les défaillances de celles-ci se font fréquentes et peuvent résulter soit d'une fin de cycle de vie d'un élément et aussi de l'apparition soudaine d'une défaillance des systèmes automatiques. Alors que ces éléments de production représentent pour la plupart du temps des enjeux majeurs soit à buts lucratifs ou pour assurer un service. Elles peuvent aussi toucher à un élément des plus importants qui représente la sécurité matérielle des systèmes automatique, environnementale, et même humaine. Pour éviter des conséquences irréversibles, des méthodes de supervision et de surveillance ont été mises en place et qui reposent toutes sur le diagnostic des systèmes [16], et il faut bien noter que les systèmes à trois cuves est muni d'une sécurité contre tout débordement possible. Cette sécurité consiste à arrêter les pompes une fois que la hauteur spécifiée a été dépassée. Ce dispositif empêchera alors les capteurs de pression de couler et le protégera contre toute destruction [15].

I.8 Conclusion :

Ce chapitre permet de présenter le contexte et la problématique du présent travail. Il s'agit d'un besoin de développer un outil de travail pour assurer le TP de diagnostic des systèmes automatique. Cet outil doit permettre aux étudiants de mettre en application les différentes notions apprises dans le cours. Pour ce, le système hydraulique à trois cuves a été

retenu. Ce chapitre a été aussi l'occasion de rappeler quelques notions nécessaires à la conception d'un système de diagnostic pour détecter et localiser un défaut avant qu'il ne puisse entraîner une défaillance grave du système automatisé (FDI : Fault Detection and Isolation) et a présenté la description physique du système hydraulique à trois cuves et de son fonctionnement.

Chapitre II : Adaptation du système hydraulique à trois cuves aux besoins du TP

II.1 Introduction

Le système hydraulique choisi pour développer un TP peut être exploité de différentes manières. Après le choix de ce système et le choix de nombre des cuves, l'enjeu a été comment adapter ou exploiter ce système pour la conception d'une maquette de TP qui répond aux différentes exigences liées au contenu du TP à savoir simuler différents types de défauts pour pouvoir ensuite tester les différentes notions du diagnostic.

Dans ce chapitre, nous présenterons la maquette des trois cuves et l'organisation du travail pratique réalisé autour de l'application pédagogique. Ainsi, il sera question de présenter le matériel expérimental, les différents composants électriques et mécaniques en plus de l'idée de conception générale de la maquette.

II.2 Présentation de la réflexion autour de la maquette

La maquette repose sur le système hydraulique à trois cuves, autour duquel toute une stratégie de développement a été mise en place pour répondre aux besoins du TP. Ainsi le développement du présent projet peut être décrit comme suit :

- 1) La modélisation du système hydraulique à trois cuves ;
- 2) La simulation du système hydraulique à trois cuves ;
- 3) Considérer la partie commande du système hydraulique, la plus simple possible car ce n'est pas l'objet du TP ;
- 4) Identifier les parties du système qui peuvent présenter des défauts ;
- 5) Lister les défauts, modéliser leurs impacts sur le système et son fonctionnement ;
- 6) Trouver la possibilité (la solution de programmation) de ces défauts pour qu'ils puissent être apparents lors des simulations ;
- 7) Développer une interface graphique permettant de concrétiser tous les points cités auparavant ;
- 8) Se projeter dans la conception et la réalisation physique de cette maquette en choisissant le matériel, les composants et en estimant son coût.

Ainsi, cette réflexion peut être décomposée en plusieurs parties :

- Le système hydraulique ;
- La commande du système ;
- La maquette physique : composant et estimation budgétaire.
- Les défauts qui répondent aux besoins des TP ;
- La partie programmation ;

II.3 Le système hydraulique à trois cuves

La maquette est composée de :

- Trois cuves communicantes
- Une partie électrique
- L'informatique associée

II.3.1 Les cuves communicantes :

L'installation physique de cette application consiste en un système hydraulique à trois réservoirs, un coffret d'interface, une carte d'acquisition PCI 6024E et un ordinateur, tandis que cette dernière est utilisée comme unité d'entrée et de sortie. Chacune de ces entités sera séparée en détail (**figure (II.1)**).

Avec :

C_i : Capteur de la hauteur d'eau dans le réservoir.

V_{pi} : Tension appliquée à la $i^{\text{ème}}$ pompe d'alimentation.

V_{ci} : Tension issue du $i^{\text{ème}}$ capteur de niveau.

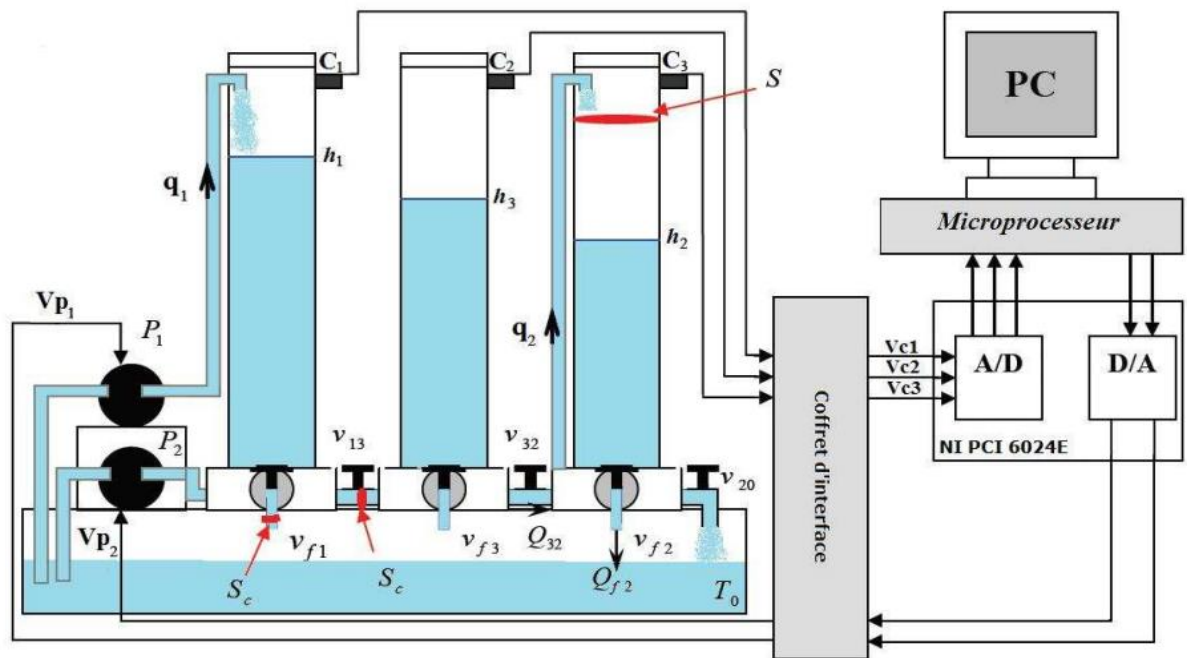


Figure II.1 : Schéma synoptique de la mise en œuvre pratique [15].

II.3.2 La partie électrique :

II.3.2.1 Composants électrique utilisé :

- 03 capteurs de pression différentielle à résistance piézo-résistif
- 02 pompes
- 01 coffret d'interface

II.3.2.2 Identification des composants et ses fonctionnements :

❖ Capteur :

Capteur de pression piézo-résistif : Les capteurs de pression piézo-résistifs basés sur le silicium se composent d'une membrane mince avec des résistances intégrées formant un pont de Wheatstone. En présence d'une sollicitation en pression, la valeur des résistances électriques change sous l'action des contraintes mécaniques (l'effet piézoélectrique). Si le pont de Wheatstone fonctionne à une tension électrique, un signal proportionnel à la pression est obtenu à la sortie du capteur [17].

Pression différentielle : La pression différentielle est la différence entre deux pressions quelconques. Les capteurs de pression différentielle ont deux liaisons de pression et peuvent être étalonnés pour mesurer les pressions différentielles positives et négatives (capteurs différentiels bidirectionnels) [17].

➤ **Son rôle dans la maquette :**

Chaque réservoir cylindrique, en haut, est équipé d'un capteur de pression différentielle avec une résistance piézo- électrique, et un tuyau rempli est monté à la pleine hauteur de la colonne. Son extrémité supérieure est bouchée par le capteur de pression, l'extrémité inférieure est maintenue au fond de la colonne, de sorte que le poids de l'eau dans la colonne est pressé sur le capteur qui détermine la hauteur de l'eau à travers la résistance.

Le capteur donne comme sortie une tension qui correspond au niveau de liquide dans le réservoir (+5 V pour 0 cm et -5 V pour 60 cm de niveau d'eau). Les informations envoyées à l'ordinateur seront converties en hauteur d'eau (cm).

❖ **Les pompes :**

Définition : Une pompe volumétrique transforme l'énergie mécanique en énergie hydraulique, elle aspire l'eau contenu dans le réservoir puis le refoule sous pression dans les cuves comme le montre la **figure (II.2)** [18] :

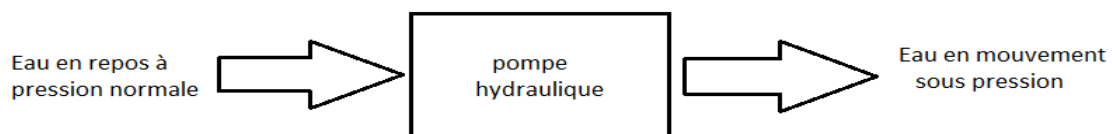


Figure II.2 : schéma de fonctionnement d'une pompe hydraulique [18]

Une pompe volumétrique est constituée de [18] :

- D'un corps fixe ou stator,
- D'un ou de plusieurs éléments mobiles participant au déplacement du fluide à l'intérieur de la pompe.

➤ **Symbole :**

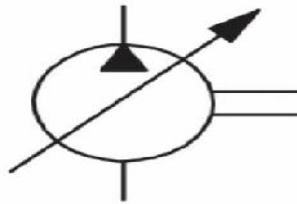


Figure II.3 : symbole de la pompe à débit variable et à un sens de flux [18].

➤ **Son rôle dans la maquette :**

Les deux pompes indépendantes $P1$ et $P2$ sont entraînées par deux moteurs DC. Elles sont conçues pour fournir un débit précis en tournant. Le liquide (eau distillée) recueilli dans le réservoir tampon par les flux de sortie est injecté par ces deux pompes avec le débit q_1 dans le réservoir T1 et le débit q_2 dans le réservoir T2. Les deux pompes sont commandées par deux actionneurs dont la tension d'entrée contrôlée le débit d'eau vers les réservoirs. Le débit fourni par chaque pompe est proportionnel à la tension appliquée au moteur CC (-10 V correspond à un débit de 0% et +10 V correspond à un débit de 100%, note q_{1max} de la pompe 1 et q_{2max} de la pompe 2).

❖ **Coffret d'interface :**

Les informations sont échangées entre l'ordinateur et le système hydraulique via le boîtier d'interface. Cette boîte grise effectue les trois tâches importantes suivantes [15] :

- Alimentation électrique.
- Commande des pompes.

- Mise en forme des signaux issus des capteurs.

II.3.3 L'informatique associée :

L'informatique associée est constituée de :

II.3.3.1 Sur le plan matériel :

1 PC de développement équipé de carte d'acquisition PCI-6024E

❖ *Carte d'acquisition :*

La carte National Instrument PCI-6024E (la **figure (II.4)**) fournit des données fiables pour répondre aux besoins d'un large éventail d'applications. Les caractéristiques de la carte sont [15] :

- Nombre de CNA : 2 CNA0 et CNA1
- Nombre de bits : 12
- Plage de Tension : [-10, 10] V
- Résolution : 4.88 mV
- Impédance de sortie : 0.1 Ω
- Dynamique (slew rate) : 10 V/ μ s
- Courant max : \pm 5 mA
- Nombre de voies simples : 16
- Voies logiques : 8 entrées et 8 sorties



Figure II.4 : la carte National Instrument PCE-6024E [19].

II.3.4 Les critères de choix des composants :

- Budget disponible
- Application concernée
- Surface disponible
- Environnement de fonctionnement
- Service nominal
- Le fonctionnement de certain composant

II.4 La commande du système par PID

II.4.1 Systèmes de commande en automatique :

L'objectif de la commande est de travailler sur les entrées du système afin que les sorties soient mieux suivies par des consignes ou des références malgré diverses perturbations qui entravent le développement du système commandé. À cette fin, deux systèmes de commande étaient disponibles pour l'élaboration de loi de commande [20].

II.4.1.1 Système de Commande en boucle ouverte :

Le calcul de la grandeur de commande à appliquer aux entrées peut être réalisé en une boucle ouverte qui n'est développée qu'à partir des grandeurs désirées $r(t)$. Cette structure de commande ne permet pas de lutter contre les perturbations du fait de l'absence de retour d'information concernant l'évolution réelle du système [20].

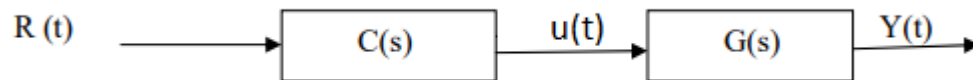


Figure II.5 : structure de système de commande en boucle ouverte [20].

Dans la **Figure (II.5)**, on a :

$R(t)$: consigne.

$U(t)$: commande en boucle ouverte

$Y(t)$: sortie mesurée de procédé

La fonction de transfert du système commandé en boucle ouverte est :

$$H_{BO} = C(s) G(s) \quad (II.1)$$

$C(s)$: fonction de transfert de régulateur.

$G(s)$: fonction de transfert de procédé.

II.4.1.2 Système de commande en boucle fermée :

Le calcul de la grandeur de commande à appliquer aux entrées, dans le cas d'un système en boucle fermée, est réalisé à partir de grandeurs désirées $r(t)$ et de grandeurs mesurées $y(t)$. Cela fournit une rétroaction sur le développement réel du système qui permet d'ajuster la grandeur de la commande.

Une telle structure bouclée peut lutter contre les perturbations et réduire la sensibilité des erreurs de modèle [20].

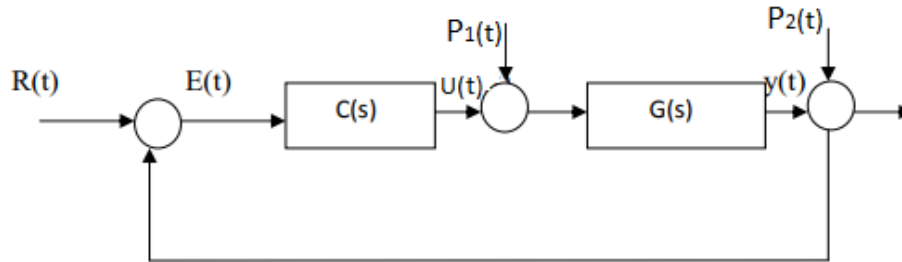


Figure II.6 : structure de système de commande en boucle fermée [20].

La fonction de transfert en boucle fermée est [20] :

$$H_{BF}(s) = \frac{H(s)}{1 + H(s)} \quad (II.2)$$

Le signal d'erreur est défini par :

$$E(s) = R(s) - Y(s) \quad (II.3)$$

La sortie du procédé est donnée par :

$$Y(s) = \frac{C(s) G(s)}{1 + C(s) G(s)} R(s) + \frac{C(s) G(s)}{1 + C(s) G(s)} P_1(s) + \frac{C(s) G(s)}{1 + C(s) G(s)} P_2(s) \quad (II.4)$$

Dans la **Figure (II.6)**, on a :

$E(t)$: erreur.

$U(t)$: commande en boucle fermée.

$P_1(t), P_2(t)$: perturbations à l'entrée et à la sortie

$Y(t)$: sortie mesurée du procédé.

II.4.2 La commande PID :

En pratique, une certaine catégorie de systèmes à asservir correspond à un type de correcteur adopté. Pour faire un choix judicieux, il est important de connaître les effets des différentes actions de régulateur PID : proportionnelle, intégrale et dérivée. Présenté sur le schéma (**Figure II.7**) [21].

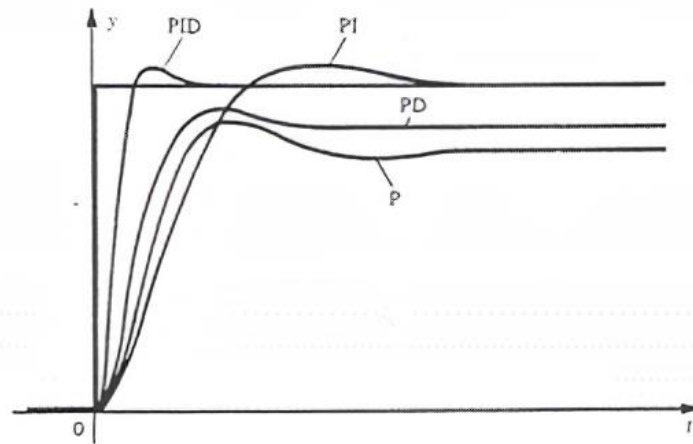


Figure II.7 : l'effet de chaque action du correcteur PID pour un même procédé [21].

Le régulateur PID est obtenu en combinant ces trois actions et remplit essentiellement les trois fonctions suivantes [21] :

1. Il fournit un signal de commande tenant compte de l'évolution du signal de sortie par rapport au point de consigne
2. Il élimine l'erreur statique avec le terme intégrateur
3. Il anticipe les variations de la sortie grâce au terme dérivateur.

La commande $u(t)$ donnée par le régulateur PID, dans sa forme Classique est décrite par [21] :

$$U(t) = K_p \left[\varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau + T_d \frac{d\varepsilon(t)}{dt} \right] \quad (\text{II.5})$$

Elle est la somme de trois termes [21] :

- Le terme proportionnel :

$$P = K_p \varepsilon(t) \quad (II.6)$$

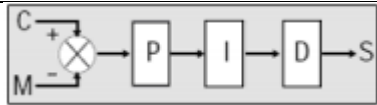
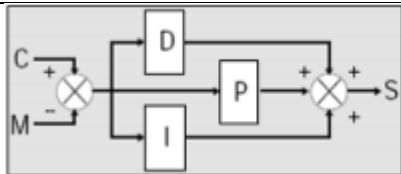
- Le terme intégral :

$$I = K_p \frac{1}{T} \int_0^t \varepsilon(\tau) d\tau \quad (II.7)$$

- Le terme dérivatif :

$$D = K_p T_d \frac{d\varepsilon(t)}{dt} \quad (II.8)$$

Les paramètres du régulateur PID sont le gain proportionnel K_p , le temps intégral T_i et le temps dérivatif T_d , les temps étant exprimés en secondes. Un régulateur PID possède différentes structures (**Figure II.8**) [21].

Structure du régulateur	Schémas de fonction de transfert
Série	 $K_p \left(\frac{T_i - T_e}{T_i} + \frac{1}{pT_i} + pT_d \right)$
Parallèle	

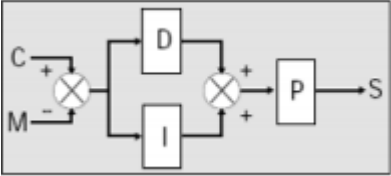
	$K_p + \frac{1}{pT_i} + pT_d$
Mixte	 $K_p \left(1 + \frac{1}{pT_i} + pT_d \right)$

Figure II.8 : les trois types de régulateur PID [21].

II.4.2.1 Influence des paramètres PID

Les illustrations suivantes montrent les effets des actions PID sur une réponse Système en boucle fermée [21] :

- À mesure que le gains KP augmente, la stabilité diminue, la rapidité augmente, la précision augmente ;
- À mesure que le Ti augmente, la stabilité augmente, la rapidité diminue et la précision reste optimale ;
- À mesure que le Td augmente, la stabilité augmente, la rapidité augmente et la précision ne se déplace pas.

II.4.3 La commande PID dans le système à trois cuves :

D'un point de vue pratique, nous pouvons exécuter des contrôleurs dans deux modes différents : manuel ou automatique. En mode manuel, l'opérateur manipule directement la sortie du contrôleur, généralement en appuyant sur des boutons qui l'augmentent ou la diminuent. En mode automatique, un ordinateur ou un autre automatisme génère les signaux de sortie du contrôleur. Le mode manuel est plus utile pendant le démarrage et l'arrêt de la maquette ou dans un état d'urgence. D'autres théories alternatives de conception de

contrôleur peuvent contrôler le niveau de fluide dans les réservoirs tout en mode automatique. Les contrôleurs PID sont formés dans un grand nombre dans toutes les industries. Ces contrôleurs ont plusieurs fonctions importantes : ils fournissent un retour d'information ; ils ont la capacité d'éliminer les décalages à l'état stationnaire grâce à l'action intégrale ; ils peuvent anticiper l'avenir grâce à l'action dérivée ; et ils peuvent faire face à la saturation des actionneurs. Beaucoup de bonnes pratiques de contrôle est conçu en eux. Ils sont donc des composants importants dans la boîte à outils des techniciens d'automatisation [22].

Un contrôleur PID standard, également appelé contrôleur à trois termes, a une fonction de transfert $C(s)$ donnée par [22] :

$$C(s) = K_p \left(1 + \frac{1}{pT_i} + pT_d \right) \quad (II.9)$$

Où K_p est le gain proportionnel, T_i la constante de temps intégrale et T_d la constante de temps dérivée. En modifiant correctement les paramètres du contrôleur PID, d'autres variantes, telles que PI, PD ou P, sont facilement obtenues. Pour les fonctionnalités à trois termes [22] :

- Le terme proportionnel fournit une action de contrôle globale proportionnelle au signal d'erreur par le facteur de gain toutes passes ;
- Le terme intégral réduit les erreurs en régime permanent grâce à une compensation à basse fréquence par l'intermédiaire d'un intégrateur ;
- Le terme dérivé améliore la réponse transitoire par compensation haute fréquence via un différenciateur.

Le **tableau (II.2)** résume les effets individuels de ces trois termes sur le rendement en boucle fermée. Les chercheurs ont mis au point de nombreuses variantes de PID pour améliorer la performance du système [22].

Réponse en B.F	Temps montant	Dépassement	Temps de stabilisation	Erreur en régime permanent	Stabilité
Augmentation du K_P	Diminue	Augmente	Petite augmentation	Diminue	Dégrader
Augmentation du K_P/T_I	Petite diminution	Augmente	Augmente	Forte diminution	Dégrader
Augmentation du $K_P T_D$	Petite diminution	Diminue	Diminue	Changement mineur	Améliorer

Tableau II.1 : Effets du réglage indépendant des proportions, intégral et dérivé (PID) [22].

II.5 Estimation financière de la maquette de TP

L'estimation financière du cout de la maquette prend en considération le cout des différents composants en plus le cout des différentes prestations qui correspondent aux interventions de certains artisans telle que le raccordement et l'installation des cuves au-dessus du grand réservoir et cela nécessite plus de dépenses estimées aux alentours de 5000 DA. En considérant les prix actuels des différents composants résumés dans le **tableau II.1**, le cout global de la maquette serait de 40000 DA.

Le **tableau (II.1)** détaille le budget total pour la réalisation de la maquette :

Le composant	Le nombre	Le prix
<i>Carte d'acquisition et son interface</i>	<i>1</i>	<i>20000 da</i>
<i>Capteur de pression différentielle</i>	<i>3</i>	<i>1800 da</i>
<i>Pompe hydraulique</i>	<i>2</i>	<i>3600 da</i>
<i>Cuve d'eau</i>	<i>3</i>	<i>3000 da</i>
<i>Réservoir tampon</i>	<i>1</i>	<i>2000 da</i>
<i>Fils et tuyaux</i>	<i>7</i>	<i>1500 da</i>
<i>Les valves</i>	<i>6</i>	<i>3000 da</i>

Tableau II.2 : budget de la maquette

II.6 Les défauts dans la maquette :

II.6.1 Modèles des défauts :

Un modèle de défaut est une représentation formelle des défauts et des façons d'affecter le système. En général, l'utilisation de modèles de défaut assure une meilleure performance du diagnostic. Plus précisément, des défauts de faibles amplitudes peuvent être détectés et différents types de défauts peuvent être localisés [06].

II.6.2 Différentes origines de défaut :

Il y a plusieurs causes possibles de panne qui peuvent modifier les relations qui causent un processus. Les défauts sont des événements qui se produisent dans différentes parties du système. Dans la littérature, les défauts sont classés par localisation, ce qui permet d'identifier

trois types de défauts comme le montre la **figure (II.9)**. Il y a trois types de défauts dans le processus [23] :

II.6.2.1 Défaut sur le capteur :

La mesure fournie par le capteur n'est pas réaliste car elle comporte une erreur [23].

II.6.2.2 Défaut sur l'actionneur :

L'actionneur ne répond pas correctement à la commande qui lui est imposé [23].

II.6.2.3 Défaut sur le système ou composants :

Correspond au changement constant et adéquat de ses caractéristiques physiques [23].

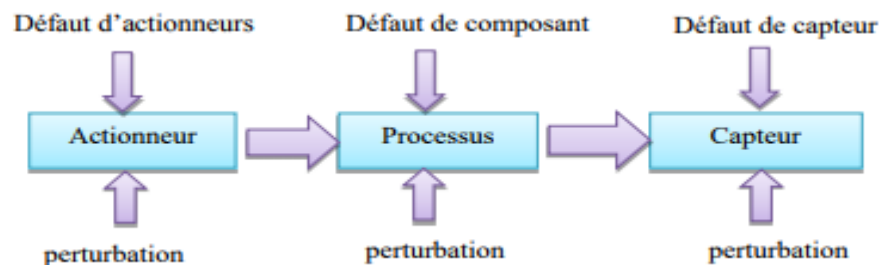


Figure II.9 : Les trois types de défauts d'un procédé [23].

II.6.3 Type de défauts :

Les défauts peuvent être différenciés selon leur forme et leur comportement dans le temps,

En générale on distingue trois types [23] :

- **Défaut abrupt** : Fait une pause dans l'évolution temporelle de la variable [23].

Il est caractéristique d'une panne brutale : arrêt total ou partiel de connexion [8].

- **Défaut intermittent** : Il s'agit d'un cas particulier de défaut abrupt de la caractéristique particulière dans laquelle le signal revient de façon aléatoire à sa valeur normale. Ce type de défauts caractérise les mauvais contacts [8].
- **Défaut graduel** : c'est un défaut caractéristique d'une usure d'une pièce ou d'un encrassement. Il est très difficile à détecter en raison de son évolution temporelle qui peut être confondu avec le changement lent des paramètres qui non stationnarité du processus [23].

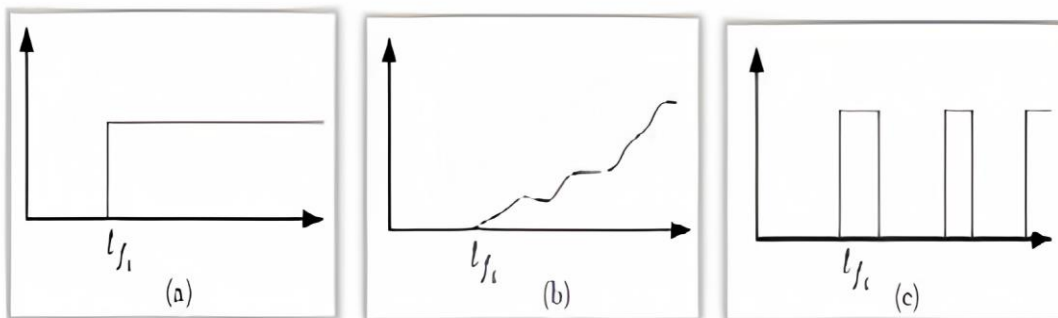


Figure II.10 : Evolution temporelle des différents types de défauts

a) Abrupt, b) graduel et c) intermittent [23].

II.6.4 Classification des différents défauts dans la maquette :

Afin d'adapter au mieux l'outil à développer aux besoins du TP, il a été impératif d'identifier un nombre minimal de défauts à considérer. Ainsi, il a été question de diversifier l'emplacement de ces défauts, leurs natures et de pouvoir simuler leur impact sur le fonctionnement du système tout en combinant réalisme et difficulté. Les défauts peuvent être classés selon leur type et origine comme indiqué dans le **tableau (II.3)**.

CHAPITRE II : ADAPTATION DU SYSTEME HYDRAULIQUE AUX BESOINS DU TP

L'origine de défauts	Défauts sur les capteurs	Défauts sur les actionneurs	Défauts sur le système ou composant
Types de défauts			
Défaut abrupt	<ul style="list-style-type: none"> ○ Détecteur endommagé, dérégulé, inactif ou déconnecté ○ L'exploitation du signal du capteur n'est pas possible 	<ul style="list-style-type: none"> ○ Pompes hors service ○ Fonctionnement dégénéré d'actionneur <ul style="list-style-type: none"> ○ Diminution de puissance 	<ul style="list-style-type: none"> ○ Fil coupé ○ Composant endommagé
Défaut graduel	<ul style="list-style-type: none"> ○ Dommages ascendants du capteur 	<ul style="list-style-type: none"> ○ Défaillance au stator du moteur ○ Défaillance au rotor du moteur ○ Défaut roulement en moteur ○ Sur charge en vitesse ou en puissance de la pompe ○ La température de fonctionnement élevée 	<ul style="list-style-type: none"> ○ Une charge excessive ○ Dégradation des matériaux qui la maquette composent ○ Les fuites dans les réservoirs ○ Bouchage des valves entre les réservoirs
Défaut intermittent	<ul style="list-style-type: none"> ○ Un écart entre la valeur réelle de la grandeur et la mesure ○ Un défaut au niveau de calibrage, un biais ou un bruit ○ La relation entre le signal généré par le capteur et celui du système réel est toujours plus ou moins adéquate 	<ul style="list-style-type: none"> ○ La corrosion due à l'eau ○ L'usure du roulement 	<ul style="list-style-type: none"> ○ Erreur dans la chaîne d'acquisition de l'information ○ Déséquilibre de la tension

Tableau II.3 : types et origines des défauts sur la maquette.

La logique de détection des défauts est illustrée par la **figure (II.11)** sous forme d'un organigramme :

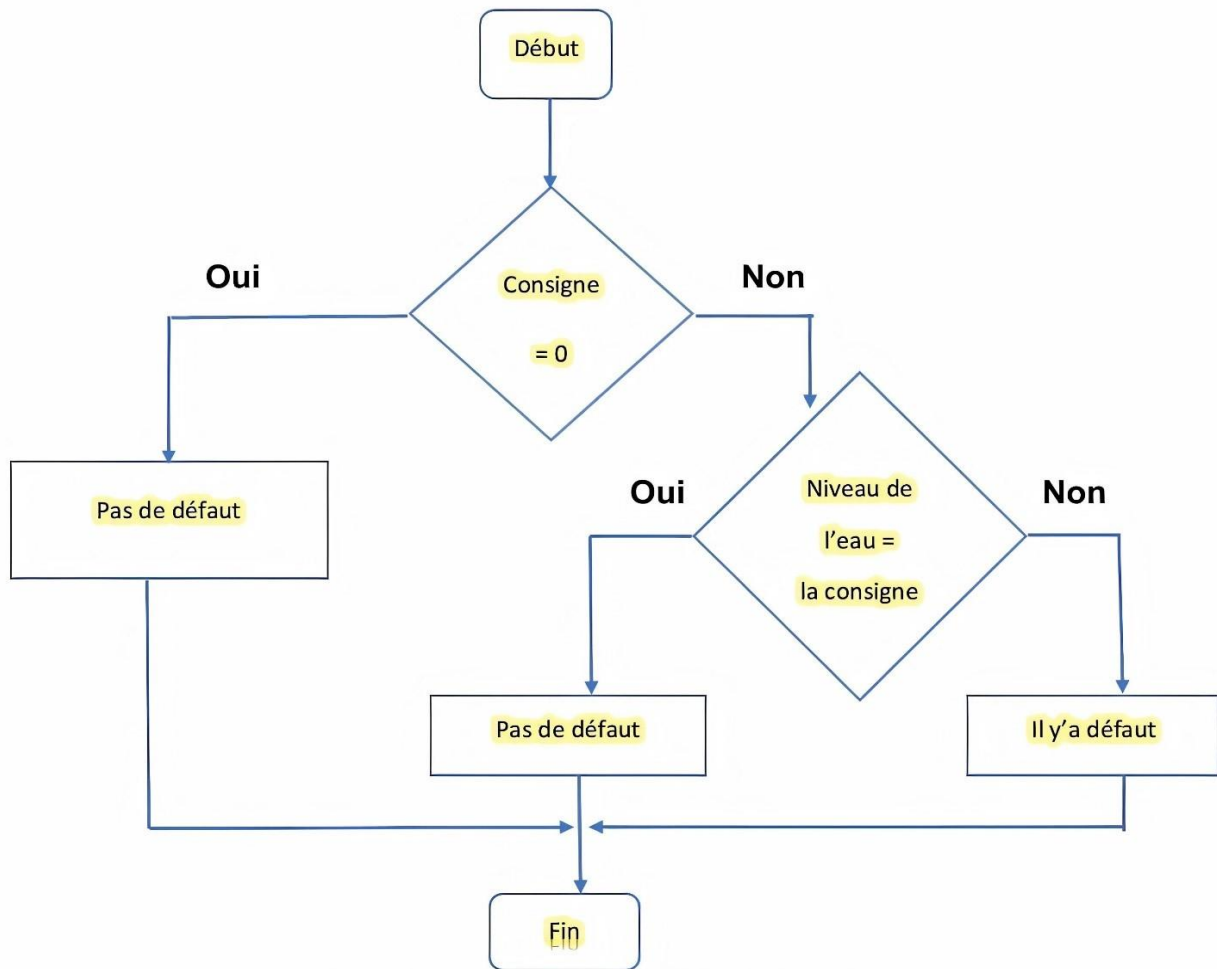


Figure II.11 : organigramme de détection de défaut.

II.7 Le logiciel de programmation

La première version de la solution pour le TP diagnostic se veut être une application software qui soit la plus réaliste possible. Et pour ce faire on avait besoin d'un logiciel qui permettrait la visualisation du système avec ses cuves. Un logiciel qui permettrait de voir les

variations des niveaux de l'eau et l'impact des défauts. C'est pour ces raisons, que notre choix s'est porté sur le logiciel **LabVIEW** qui est un langage de programmation graphique à caractère universel bien adaptée pour la mesure, les tests, l'instrumentation et l'automatisation qui utilise des icônes à la place de texte pour créer des applications interactives.

II.8 Conclusion :

Ce chapitre présente une grande partie de réflexion autour de la solution pour le TP diagnostic des systèmes automatiques. Nous y avons exposé les solutions concernant les différents points à prendre en considération pour passer du besoin à la solution en passant par différentes adaptations qui ont permis d'éclairer tout le travail qui doit être fait sur les différentes composantes de la maquette à savoir : le système, sa commande, ses défauts, son estimation budgétaire ainsi que le logiciel de programmation qui sera présenté dans le chapitre suivant.

Chapitre III : Présentation du logiciel LabVIEW

III.1 Introduction

La solution du TP diagnostic sera programmée moyennant le logiciel LabVIEW qui est un langage de programmation graphique à caractère universel bien adapté pour la mesure, les tests, l'instrumentation et l'automatisation qui utilise des icônes à la place de lignes de texte pour créer des applications. Contrairement aux langages de programmation textuels où les instructions déterminent l'exécution du programme.

III.2 Environnement LabVIEW

LabVIEW est un outil d'acquisition, d'analyse et d'affichage de données. Dans LabVIEW, une interface utilisateur (face avant) est créée à l'aide d'une combinaison d'outils d'objets. L'interface utilisateur est compatible avec ce qui apparaîtra sur l'écran du PC en mode fonctionnement, ce qui permettra à l'utilisateur de contrôler le programme (donner des entrées).

Dans le cadre du programme Affichage de l'information (résultats), la principale préoccupation de LabVIEW est de permettre le développement facile de ces interfaces à l'aide des bibliothèques existantes [24].

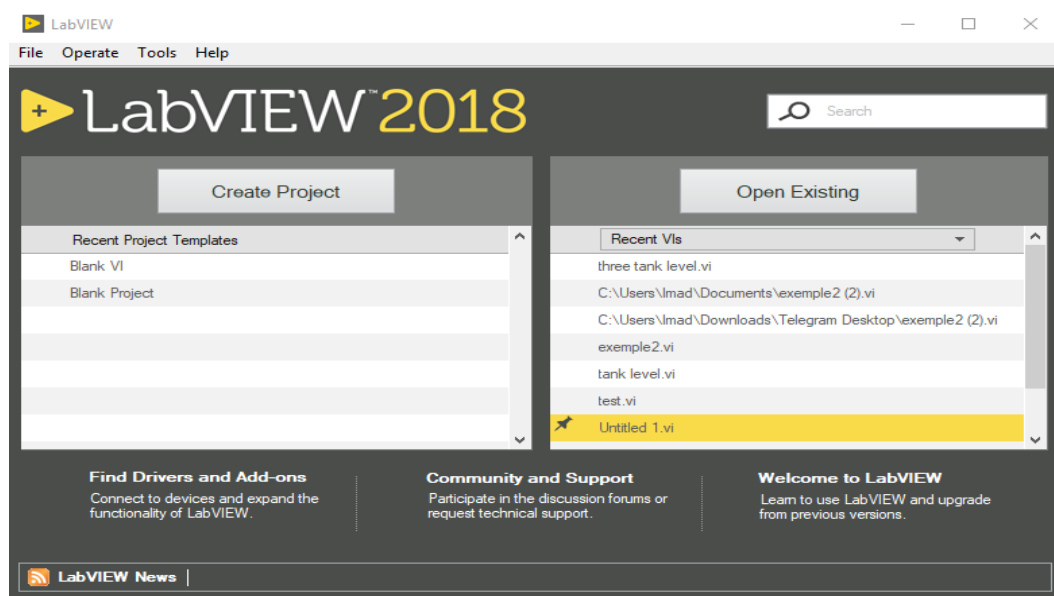


Figure III.1 : Fenêtre de démarrage du logiciel LabVIEW [24].

Le même programme est ajouté sur la face arrière un ensemble de symboles utilisant un langage de programmation graphique qui permet le contrôle des objets sur la face avant dans une certaine direction, le diagramme sur la face arrière ressemble à un flowchart [24].

III.3 Instruments virtuels (VI)

Les programmes LabVIEW sont appelés outils virtuels (Virtual instruments en anglais) ou encore VI, parce qu'ils ressemblent et fonctionnent comme des instruments physiques par exemple les oscilloscopes ou les multimètres. Chaque VI est constitué de ces éléments suivants :

- ✓ Face avant (Front Panel) : Sert d'interface utilisateur
- ✓ Diagramme (Block Diagram) : Contient le code source sous forme graphique qui définit les fonctionnalités du VI.
- ✓ Icône et connecteurs : ça permet d'identifier un VI au sein d'un autre. Un VI utilisé dans un autre est appelé un sous VI, cela correspond à un sous-programme dans le langage textuel.

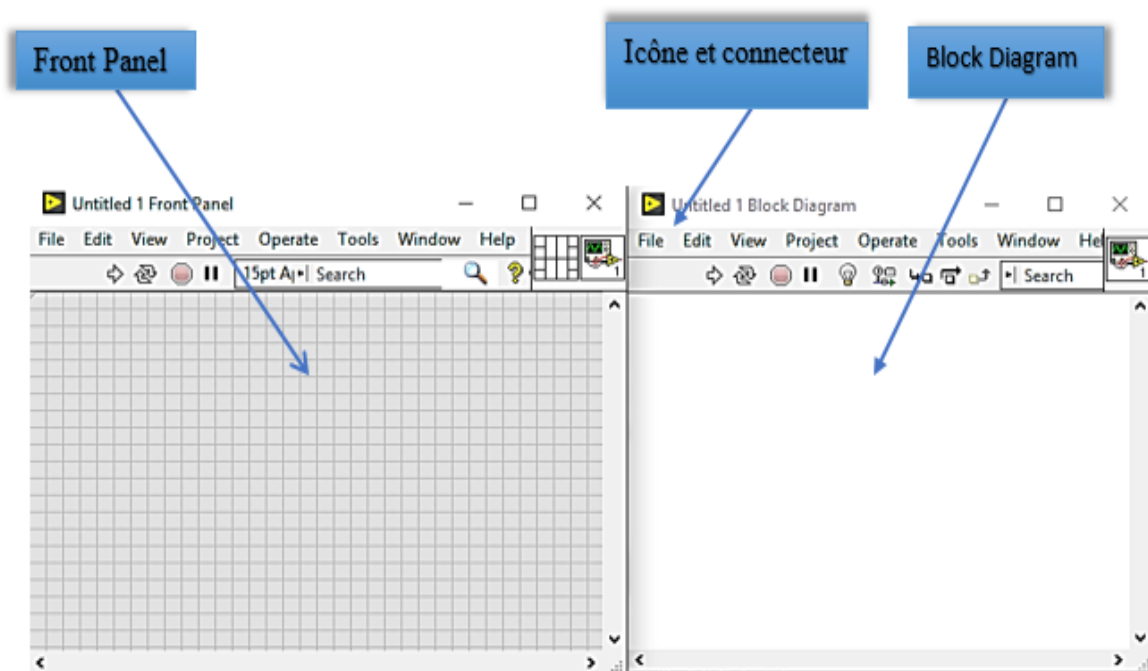


Figure III.2 : Composants d'un VI.

III.3.1 Face avant :

La face avant c'est l'interface utilisateur du VI. Nous créons la face avant avec les commandes (entrées) et les indicateurs (sorties) qui sont les entrées et sorties du VI qui interagissent avec l'utilisateur. Les commandes peuvent être des commutateurs. Des boutons poussoir, des boites de dialogue et d'autres composants d'entrée. Les indicateurs sont des graphiques de LEDs et d'autres systèmes d'affichage. Les contrôles simulent des composants d'entrées qui fournissent au diagramme du VI des données. Les indicateurs simulent les outils de sortie qui affichent les données d'un graphique ou qui ont été créés par le diagramme ou qui ont été générées.

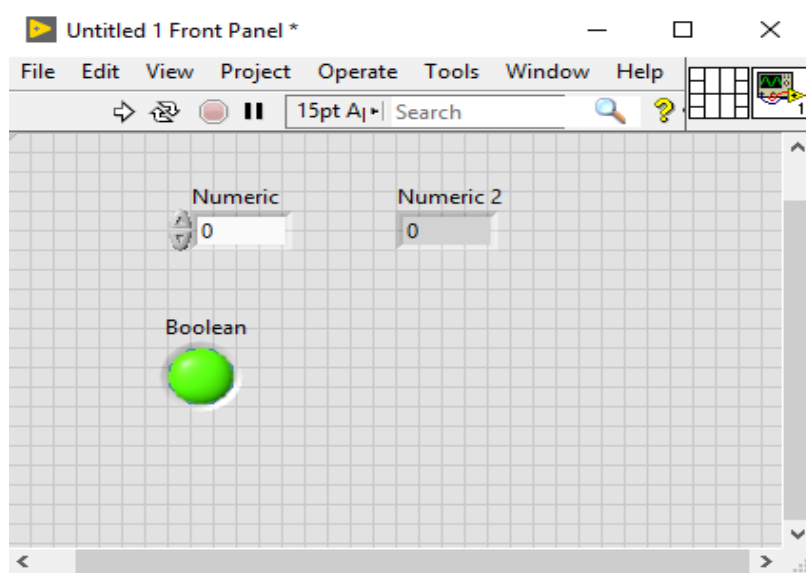


Figure III.3 : VI en face avant.

III.3.2 Face arrière (Block Diagram)

Après la création de la face avant, l'icône est ajoutée en utilisant des représentations graphiques des fonctions pour contrôler les objets sur la face avant. Le diagramme d'arrière-plan comprendra le code source sous forme de graphique. Les objets sur le devant apparaîtront comme extrémités (entrée ou sortie) du graphique. En outre, le graphique sera en mesure de contenir les fonctions et les structures disponibles dans les bibliothèques de LabVIEW VI pour relier chaque nœud dans le graphique, y compris les fonctions, les structures de contrôle et les indicateurs.

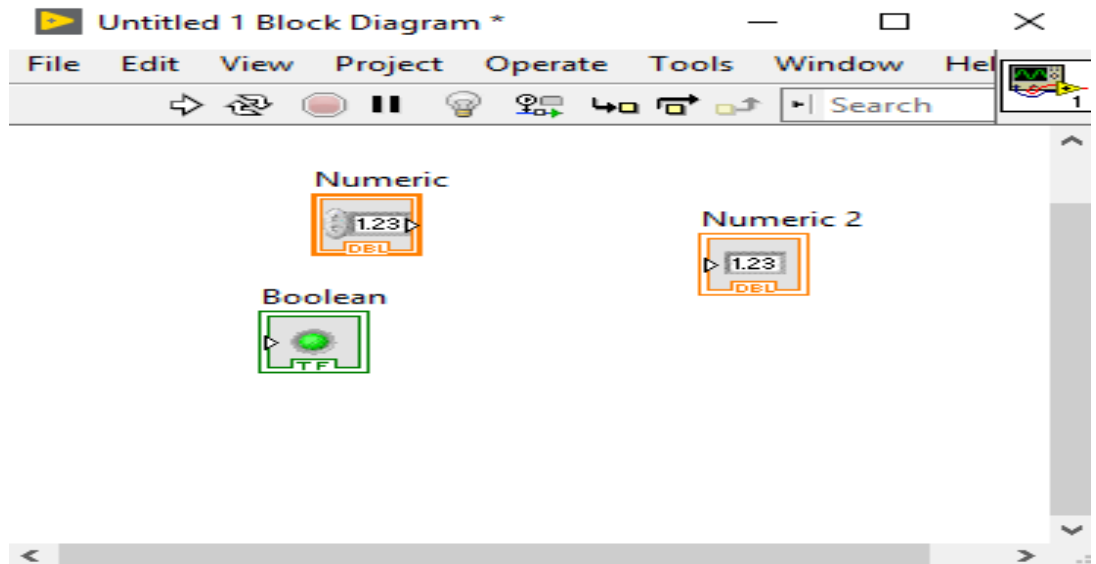


Figure III.4 : VI Face arrière [24].

III.4 Barres d'outils et palettes

III.4.1 Barre d'outils de la face avant :

Cette barre d'outils donne accès aux outils d'exécution et de présentation du programme [24].

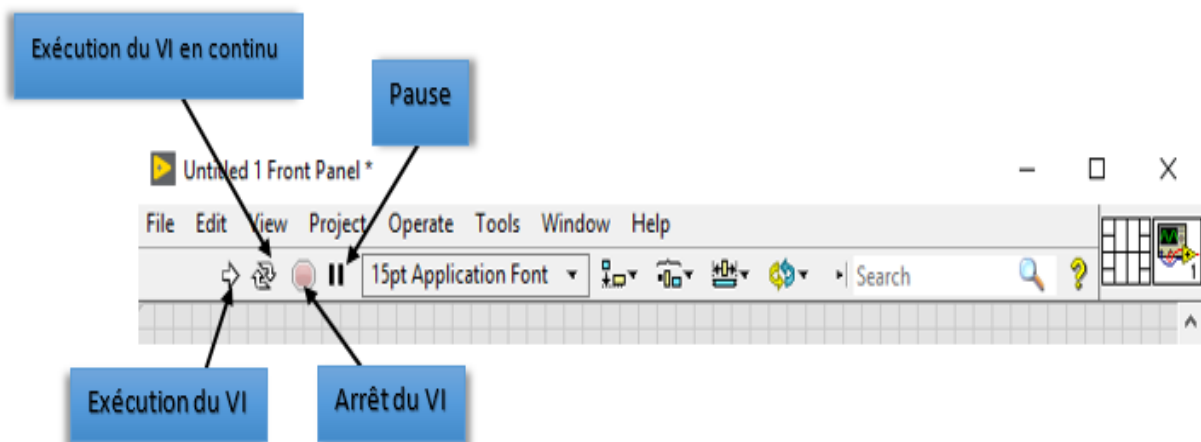


Figure III.5 : Barre d'outils face avant [24].

III.4.2 Barre d'outils du diagramme :

Cette barre d'outils donne accès aux outils d'exécution et de présentation du programme [24].

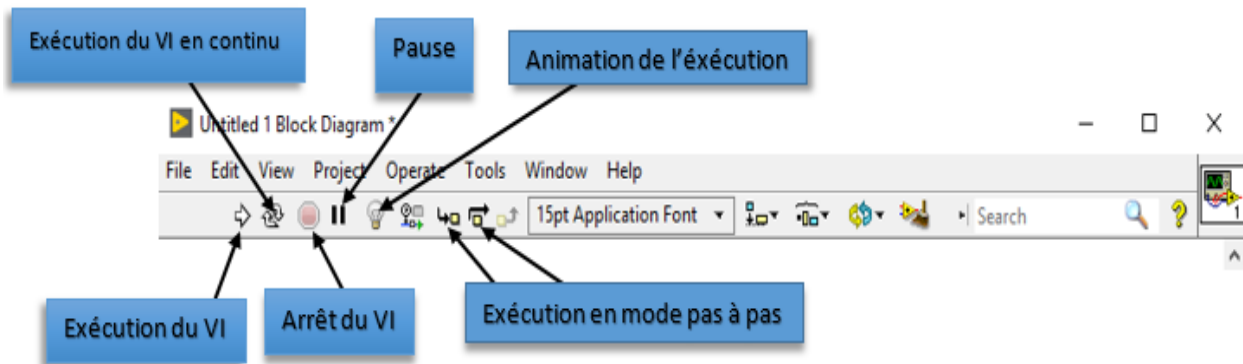


Figure III.6 : Barre d'outils face arrière.

III.5 Les palettes LabVIEW

Les palettes LabVIEW vous fournissent les outils qui vous permettront de créer et d'éditer la face avant ou le diagramme de la face arrière. La palette outils est disponible aussi bien sur la face avant que la face arrière un outil est un mode de fonctionnement spécial du curseur de la souris. Quand vous sélectionnez un outil le curseur de la souris se modifie.



Figure III.7 : La palette d'outils.

Si la sélection automatique d'outils est autorisée et que vous déplacez la souris sur des objets à la face avant ou à l'arrière, LabVIEW sélectionne automatiquement l'outil approprié dans la palette de couleurs. La palette commande est uniquement accessible depuis la face avant la palette commande contient les contrôles et les indicateurs que vous pouvez utiliser pour créer la face avant sélectionner fenêtre palette commande ou faites un clic droit sur la fenêtre de travail de la face avant pur faire s'afficher la palette commande vous pouvez placer celle-ci n'importe où sur l'écran.

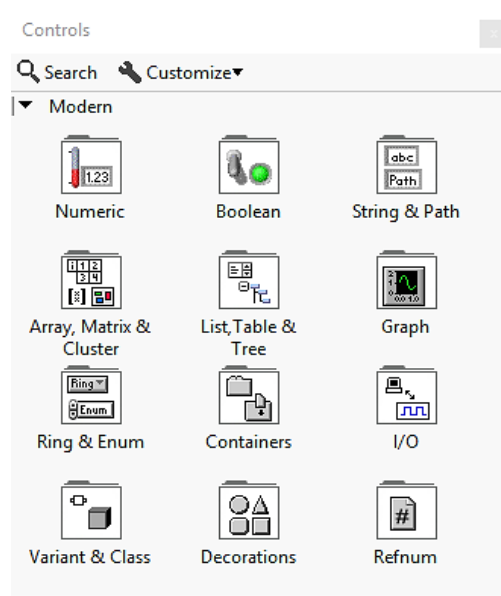


Figure III.8 : La palette commande.

Il existe déjà un ensemble de composants qui ont déjà été créés pour vous permettre de créer facilement une interface utilisateur.

Le panneau de fonctions est accessible uniquement à l'arrière. Le panneau de fonctions contient VIs et les fonctions que vous pouvez utiliser pour créer le diagramme dans la fenêtre Sélectionner les fonctions Panneau de couleurs à l'arrière ou cliquez avec le bouton droit sur la fenêtre de diagramme pour afficher le panneau de fonctions que vous pouvez placer des fonctions de panneau de couleurs n'importe où sur l'écran [24].

C'est là que vous trouverez toutes les fonctions qui ont été précédemment écrites sous LabVIEW, mais vous pouvez également trouver les fonctions que vous avez créés ensuite

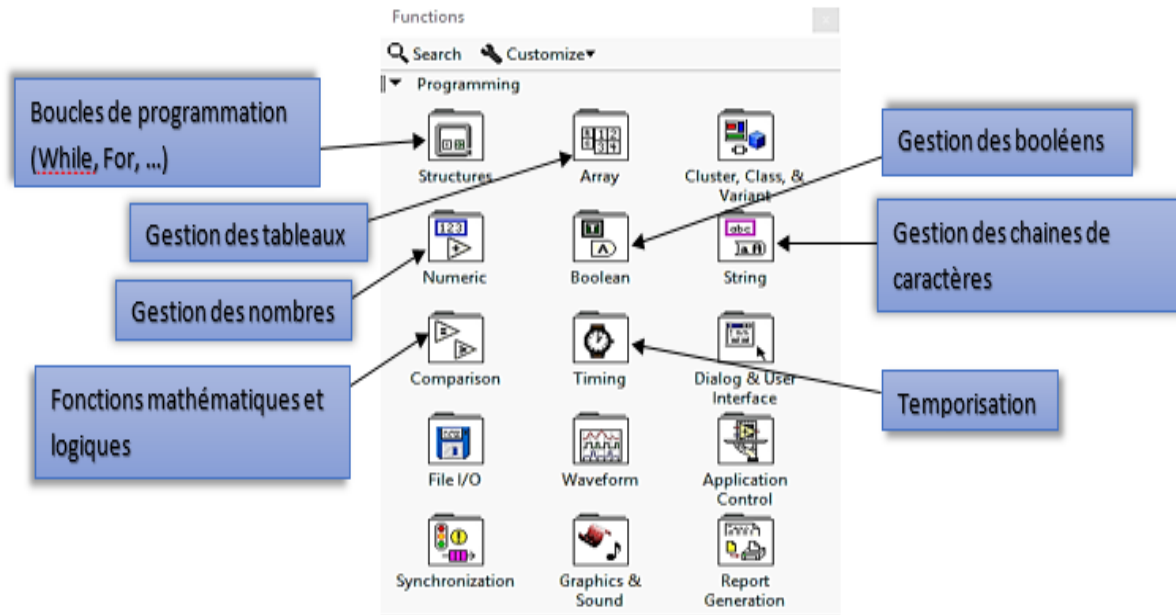


Figure III.9 : La palette fonctions [24].

Bien que les options puissent être énormes pour le novice, la documentation complète vous permet d'apprendre rapidement les bases du programme. Les ressources en ligne comprennent des tutoriels étape par étape sur la création d'applications avancées et de méthodes de programmation.

Un aspect important est l'intégration de dispositifs avec des mesures qui facilitent l'ajout de capteurs et d'analyseurs de signaux aux applications. L'interface VI vous permet de concevoir des systèmes de contrôle visuellement complexes en utilisant la programmation de flux de données.

III.6 Le choix de LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est langage de programmation personnalisé pour contrôler le matériel et les données analytiques. Contrairement à la nature séquentielle des langages textuels, LabVIEW est basé sur un environnement de programmation graphique utilisant le concept de flux de données pour programmer les opérations [24].

Le temps nécessaire pour assembler un système de mesure ou de contrôle/ commande est généralement négligeable par rapport au temps nécessaire à la programmation en classique (C, Pascal, ...). Les interfaces utilisateur développées dans ces langages, souvent en langage texte leur syntaxe doit être reconnue. Les utilisateurs de LabVIEW peuvent disposer d'un outil intégré pour obtenir, analyser et afficher des données. Le principal avantage est de gain de temps car ce langage de programmation graphique est beaucoup plus naturel à implémenter. Cependant, ce système est basé sur les principes généraux de tout système programmé. LabVIEW dispose d'un grand nombre de fonctions graphiques préexistantes qui facilitent l'obtention, le stockage, le traitement et l'affichage des résultats [24].

Le plus grand avantage de LabVIEW, c'est qu'il existe des drivers (VIs) pour une multitude d'appareils de laboratoire ou industriels. De plus de nombreuses bibliothèques, dont certaines gratuites, sont disponibles pour tous les types d'applications. Il offre aussi la possibilité d'écrire rapidement des applications à des personnes qui sont allergiques à la ligne de code. Il est d'un accès très facile pour toute personne, un tant soit peu rigoureuse, ne connaissant pas de langage de programmation. LabVIEW Crée un environnement d'apprentissage favorable et associer la théorie à la pratique avec des exemples du monde physique et aussi enseigner aux étudiants des techniques d'analyse et de résolution de problèmes [24].

III.7 Comparaison entre LabVIEW et d'autres logiciels

III.7.1 Quelles sont les différences entre LabVIEW et MATLAB ?

LabVIEW et MATLAB ont été développés par deux entreprises américaines différentes. Le premier est développé par National Instruments et le second par Math Works. LabVIEW. LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un environnement de programmation graphique basé sur le langage de programmation graphique G. MATLAB (**MAT**rix **LAB**oratory) est le nom utilisé pour le langage de programmation et l'environnement de programmation spécialisé dans les calculs numériques. Les deux plateformes aident les ingénieurs et les scientifiques du monde entier à divers stades de conception, de modélisation,

de simulation, de mise à l'essai de prototypes ou de déploiement de nouvelles technologies. Les scientifiques utilisent principalement MATLAB pour leurs simulations, notamment en raison de l'abondance de bibliothèques supplémentaires et de l'ajout de Simulink. Les bibliothèques contiennent des fonctions spécifiques de niveau supérieur d'un champ particulier. Ces fonctions accélèrent le développement d'applications avancées. Dès qu'une interface utilisateur graphique fonctionnelle et intuitive ou une interaction avec le matériel (acquisition et génération de signaux) est requise au lieu de MATLAB, LabVIEW est principalement utilisé. Certains utilisateurs plus expérimentés choisissent également une combinaison des deux outils, qui communiquent via l'API ou les bibliothèques DLL. Les données peuvent être exportées d'un et importées dans un autre programme pour un traitement ultérieur. Il est également possible d'implémenter le code MATLAB dans LabVIEW. La plupart des fonctions de MATLAB peuvent être intégrées dans LabVIEW en utilisant le module Math Script function (disponible avec « Math Script RT Module » add on) [35].

Les deux langages sont presque identiques et contiennent de nombreuses bibliothèques similaires, mais l'utilisation de graphiques programmables LabVIEW est plus facile que la programmation de scripts Matlab. Nous pouvons également connecter facilement des appareils tiers à l'aide de LabVIEW. Supposons que vous voulez prendre des données à partir d'un environnement externe qui est mieux pour LabVIEW parce qu'il contient les dispositifs de capture de données disponibles à cet effet. LabVIEW permet de concevoir facilement et de manière attrayante des interfaces graphiques de contrôle utilisateur avec des systèmes plus réels que Simulink/Matlab. Il s'agit d'un système Simulink/Matlab intégré qui permet de modéliser et de contrôler les systèmes, et nous pouvons effectuer quelques petits tests rapides avec des systèmes réels simples, etc. Alors que LabVIEW permet vraiment de créer un modèle d'interface final très rapidement. La langue que vous préférez est LabVIEW pour l'automatisation ou Matlab pour la manipulation mathématique [35].

III.7.2 Quelles sont les différences entre LabVIEW et C ?

LabVIEW et C sont des outils utiles qui, dans les mains d'utilisateurs expérimentés, peuvent résoudre presque n'importe quel problème [26].

Pour Simon Hogg, « *si vous envisagez de construire un système de mesure ou de contrôle, alors le logiciel de conception de systèmes NI LabVIEW est un outil qui peut vous éviter de prendre des risques, de dépenser de l'argent ou toute autre sorte d'inconvénient lié au développement de systèmes à l'aide de langages bas niveau comme le C* » [26].

Le C est mieux adapté aux applications de tâches de bas niveau et riches en calcul, souvent plus adaptées aux applications avec moins de ressources à gérer étroitement, tandis que LabVIEW est plus orienté vers l'exécution parallèle, les applications de test, la mesure et le contrôle de haut niveau, plus rapide en raison des nombreux modules qui effectuent des calculs. Le langage C est plus pratique, mais pour un besoin rapide d'obtenir de la valeur, Ni et LabVIEW sont plus utiles [26].

III.7.3 Quelles sont les différences entre LabVIEW et WinCC ?

LabVIEW est un langage de programmation qui combine l'acquisition de données avec le développement de l'interface utilisateur dans un environnement de développement graphique simple à utiliser.

WinCC est un logiciel d'interface de machine humaine (HMI). Bien qu'il ait quelques fonctionnalités "d'acquisition de données", il n'est RIEN comme LabVIEW. Vous comparez des pommes à des oranges. Le produit de National Instruments qui est semblable à WinCC s'appelle Lookout [27].

"LabVIEW Datalogging and Supervisory Control Module" pour donner LabVIEW standard HMI fonctionnalités (comme les balises). DSC est construit sur LabVIEW un langage de programmation graphique. En raison de sa nature graphique, intuitive et de flux de données, presque n'importe qui peut "coder" avec LabVIEW, Il peut y avoir de bonnes raisons de le faire [27].

III.8 Conclusion

Ce chapitre décrit l'environnement et les techniques de programmation graphique pour la construction des applications dans LabVIEW. Ce choix est argumenté par les différentes possibilités qu'offre ce logiciel en comparaison aux autres solutions de programmation possibles telles que Matlab et le langage C.

Chapitre IV : Présentation de l'application software « Hy.Dia »

IV.1 Introduction

Dans ce chapitre on va présenter l'application software qui sera utilisée pour monter les différents TP de diagnostic des systèmes automatiques. Ces TP ont pour but d'expérimenter les différentes méthodes de calcul des résidus qui vont servir à la détection et l'isolation des défauts qui surgissent dans un système automatique. Pour notre solution software, nous avons choisi un système hydraulique à trois cuves qui répond à notre cahier de charge. C'est pour cette raison que cette solution porte l'appellation « Hy.Dia » qui renvoie à la nature hydraulique du système et au diagnostic.

Dans ce qui suit, nous présentons les étapes de développement de l'application Hy.Dia et les résultats de sa simulation avec les différents défauts qu'elle prend en charge.

IV.2 Présentation générale de Hy.Dia :

Hy.Dia est une application software qui se présente comme une solution qui simule un système hydraulique à trois cuves et ses défauts. Cette application est développée dans une logique qui permet le test des différentes notions, concepts et méthodes de diagnostic des systèmes automatiques. Il faut rappeler que pour rendre cette solution interactive pour une utilisation optimale pour l'enseignant et les étudiants en même temps, elle a été pensée de manière à permettre :

- 1) Aux étudiants, de voir, de visualiser le système avec une animation du niveau de l'eau, et de certains défauts comme les fuites d'eau.
- 2) A l'enseignant de programmer des défauts sans qu'ils soient apparents pour les étudiants afin qu'ils puissent trouver des solutions à l'aveugle. Et donc développer et tester leurs connaissances.

Hy.dia, comme toutes solutions software, est la combinaison de deux parties. Une partie visuelle, interface graphique, qui montre à l'utilisateur ce qu'on veut lui faire voir. Et une partie programmation qui se trouve en arrière-plan, donc invisible mais très importante puisque c'est elle qui régit l'interface graphique. Ces deux parties sont développées en prenant en considération les exigences suivantes :

- 1- L'interface graphique doit être très simple, interactive et surtout attrayante ;
- 2- L'interface graphique doit permettre à l'étudiant de comprendre d'une manière simple le fonctionnement du système hydraulique ;
- 3- L'interface graphique permet de simuler des défauts ;
- 4- Les défauts peuvent être simulés par les étudiants afin de se familiariser avec l'application ;
- 5- Les défauts peuvent être programmés par l'enseignant afin de tester la capacité des étudiants à les diagnostiquer sans qu'ils aient la possibilité de le faire autrement que par l'application d'une méthode de diagnostic.
- 6- Les défauts peuvent être modifiés en jouant sur leurs types, amplitudes, fréquences et instant d'occurrence.

Pour répondre à ces exigences, la réflexion s'est portée sur deux axes qui ont été développés en parallèle puisque l'un dépendait de l'autre. Il s'agit de l'interface graphique représentée dans la **figure (IV.1)**, et le programme qui la régit dont l'organigramme est représenté par la **Figure (IV.2)**.

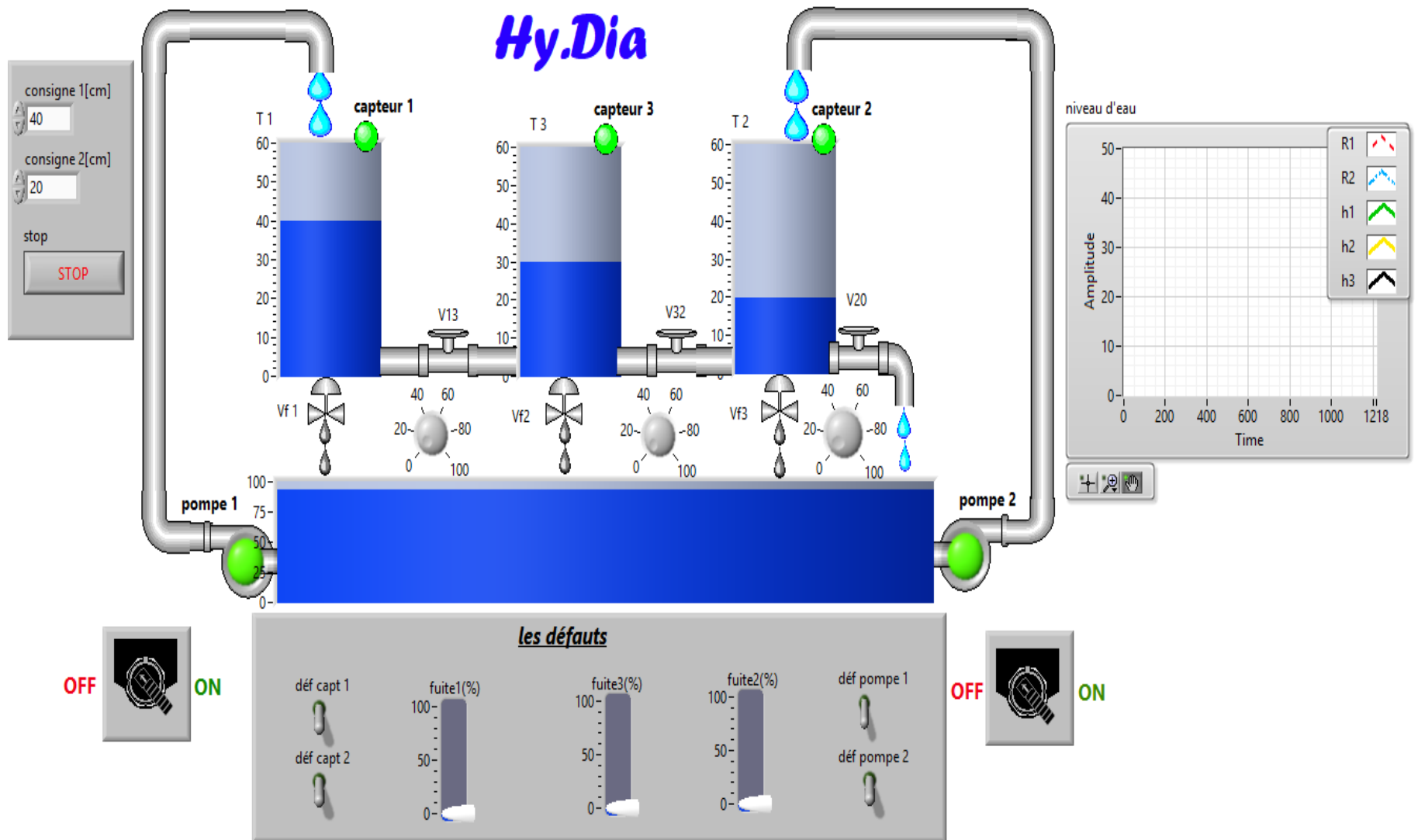


Figure IV.1 : l'interface de simulation de la maquette.

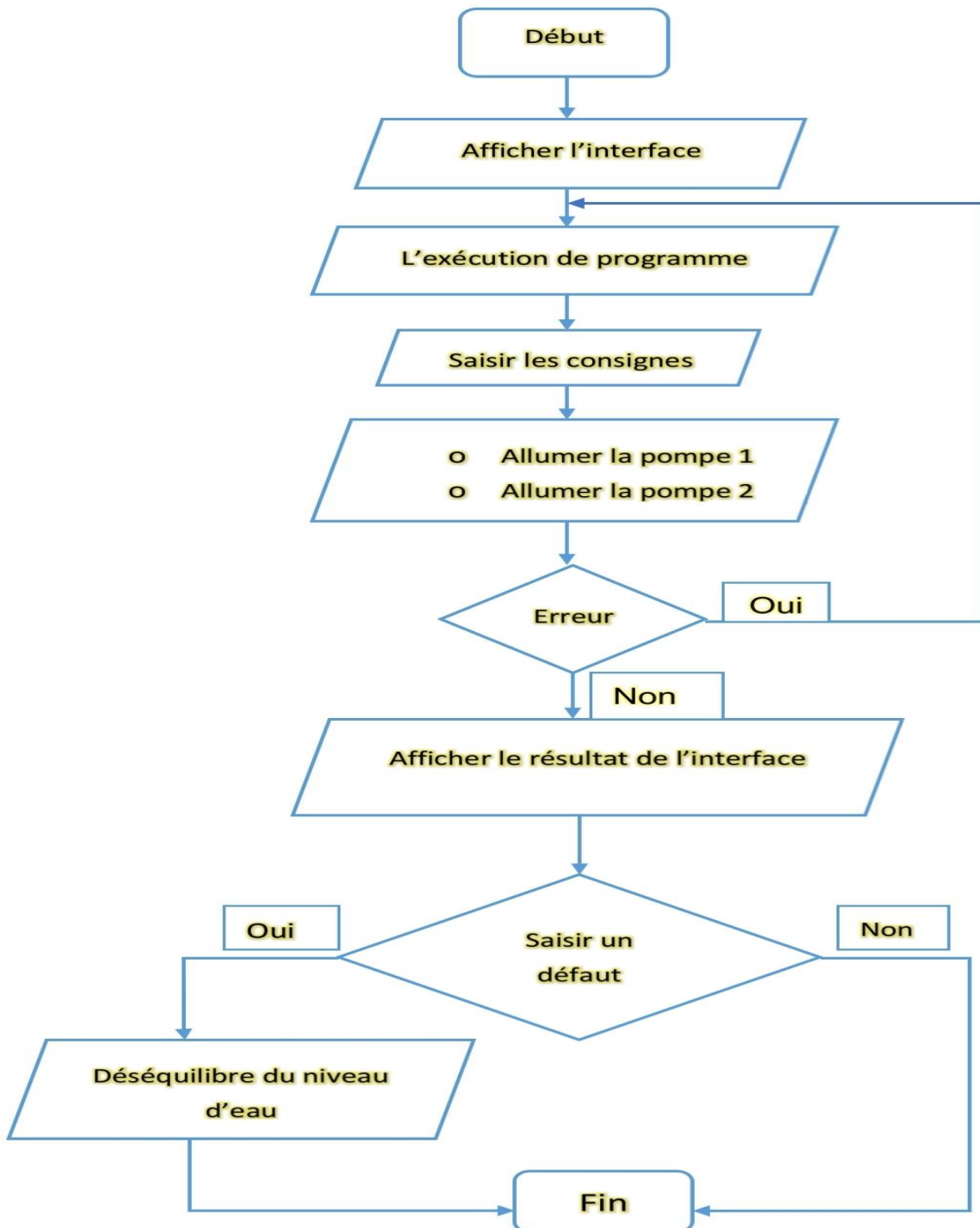


Figure IV.2 : Organigramme : résumé l'utilisation de l'interface.

Comme illustré par la **figure (IV.2)**, l'interface présente deux valeurs de référence (point de consigne), le "point de consigne1" représente le niveau à atteindre pour le réservoir 1, et le "point de consigne 2" représente la différence à atteindre pour le réservoir 2.

Dans la partie centrale sont montrés les trois réservoirs, seul le réservoir 2 retourne dans le réservoir tampon et seuls les réservoirs 1 et 2 sont réapprovisionnés par les pompes.

Dans cette interface se trouvent également les commandes de valves entre et au-dessous des réservoirs et de retour vers le réservoir tampon. Ces contrôles peuvent fonctionner pendant la simulation, ce qui entraîne le changement du système au fil du temps et un défaut du système est un bouchage effectué par les valves entre les réservoirs et par les valves sous les réservoirs.

Pour l'instant, l'interface permet de simuler deux autres types de défauts (défaut du capteur, défaut d'actionneur) avec des boutons de référence qui déterminent le type et la localisation de chaque défaut, qui provoque un déséquilibre du niveau d'eau et permet d'apprendre comment le système interagit lorsqu'un défaut est présent. Il y a aussi un bouton pour arrêter la simulation (STOP).

Pour le coté programmes d'arrière-plan, tout commence par une remise à zéro, avant d'entrer les deux consignes désirées avant de démarrer la pompe 1 avant de démarrer la deuxième pompe avec un peu de retard dans le temps pour augmenter le niveau dans les réservoirs 1 et 3 pour vérifier le cas $h_1 > h_3 > h_2$. Les réservoirs 1 et 2 sont alimentés par les pompes aux débits h_1 et h_2 au niveau requis, si le niveau dépasse la valeur requise, la pompe s'arrêtera automatiquement, pour éviter le débordement du réservoir si la pompe resterait en marche.

S'il y a une erreur de simulation, par exemple, il est n'est pas possible de laisser de délai entre le démarrage des pompes provoquant un état indéterminé, une réexécution est

nécessaire, s'il n'y a pas d'erreur, le résultat apparaît dans l'interface en forme de niveau d'eau et des graphes montrent le changement de ce dernier.

IV.3 Etapes de développement

IV.3.1 Système à un seul réservoir :

Pour expliquer les étapes de développement des trois réservoirs (cuves), nous prenons l'exemple du premier réservoir en partant de ses équations. Considérons le réservoir d'eau (voir **la figure IV.3**) avec la section transversale A , L'eau est des pompes dans le réservoir au taux supérieur de débit de q_{in} mètres cubes par seconde. L'eau sort hors du réservoir par un trou dans le fond du réservoir. Le débit sortant à travers ce trou est exprimé selon l'équation de Bernoulli donnée par :

$$q_{out} = a\sqrt{(2gh)} \quad (IV.1)$$

Où h est le niveau du réservoir et g est l'accélération de la gravité. La conservation de la masse donne l'équation :

$$A \frac{dh}{dt} = q_{in} - q_{out} = q_{in} - a\sqrt{(2gh)} \quad (IV.2)$$

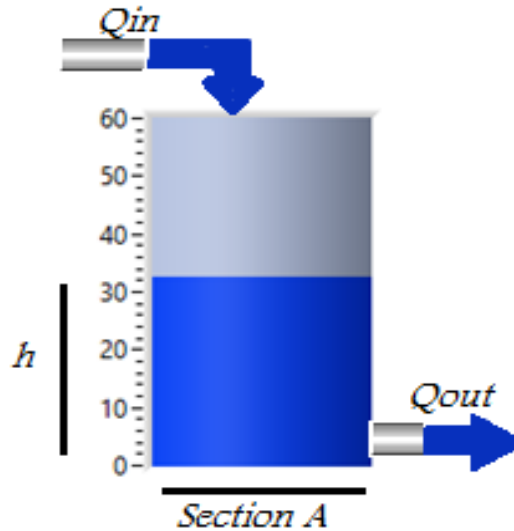


Figure IV.3 : schéma du réservoir 1.

La quantité d'eau qui s'accumule dans le réservoir est la différence de la quantité entrante et la quantité sortante :

- *Le volume cumulé :*

$$q_{in}[m^3] - q_{out}[m^3] = \Delta V [m^3] \quad (IV.3)$$

Où q_{in} , q_{out} sont les débits entrants et sortants et ΔV est le volume du liquide accumulé (eau).

En divisant l'équation ci-dessus par le temps, l'équation suivante sera obtenue :

- *La valeur cumulée en unité de temps :*

$$q_{in} \left[\frac{m^3}{s} \right] - q_{out} \left[\frac{m^3}{s} \right] = \Delta V \left[\frac{m^3}{s} \right] \quad (IV.4)$$

- *Le débit sortant :*

$$q_{out} = a\sqrt{(2gh)} \quad (IV.5)$$

On peut illustrer les calculs ci-dessus sous forme de schéma fonctionnel. La soustraction de q_{out} de q_{in} est multipliée par $1/A$ et le résultat est intégré. Le résultat de cette intégration est le niveau actuel du réservoir et en conséquence il détermine le q_{out} .

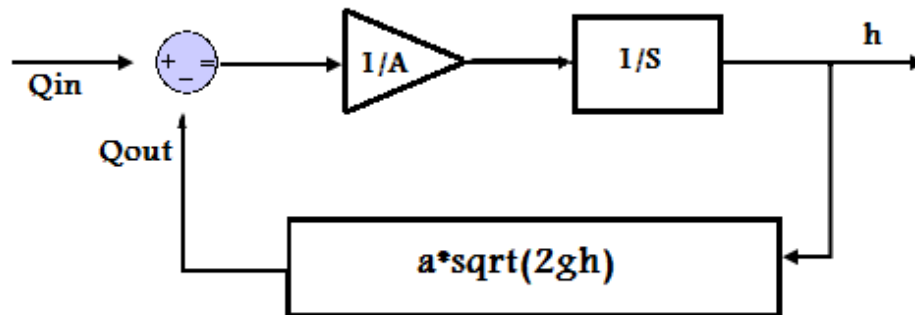


Figure IV.4 : le schéma fonctionnel d'un niveau de réservoir avec un débit d'entrée et un débit de sortie

❖ **Conception d'un système de contrôle du niveau du réservoir (à l'aide d'un contrôleur PID) :**

Dans ce système, qui est le réservoir d'eau, le point de consigne (niveau désiré) est introduit (édité), puis une interface numérique/analogique convertit cette valeur en tension, puis la valeur sortante du capteur (pour mesurer la hauteur de l'eau) est soustraite de la sortie de l'interface numérique/analogique. Le résultat de la soustraction (erreur) est fourni à un contrôleur. Le résultat du contrôleur (qui est aussi une certaine tension) est amplifié (car il devrait y avoir assez de puissance pour faire fonctionner la pompe). La tension amplifiée est reçue par la pompe et par conséquent un début de pompage de l'eau dans le fond du réservoir qui draine l'eau. Le débit d'écoulement est fonction de la hauteur de l'eau dans le réservoir.

La hauteur de l'eau est mesurée par un capteur de pression différentielle. Ce capteur convertit la hauteur actuelle de l'eau en tension. Cette valeur est envoyée à l'entrée de la chaîne pour calculer l'erreur. Cependant, la pompe ne fonctionne pas immédiatement après avoir reçu l'électricité. Il devrait qu'un certain temps s'écoule après avoir reçu l'électricité et commence à pomper.

En outre, la tension sortant de l'amplificateur ne peut pas être supérieure à une certaine valeur et elle ne peut pas être inférieure à zéro. La même condition s'applique au débit sortant de la pompe et au débit le plus élevé de l'eau. Le débit d'entrée vers le réservoir ne peut pas être négatif ni supérieur à la capacité de la pompe. Le niveau d'eau ne peut être inférieur à zéro ni supérieur à la hauteur du réservoir. Une telle restriction (saturation) est appelée non-linéarité.

La **figure (IV.5)** illustre le schéma fonctionnel de la chaîne de commande. Notons que le bloc illustré sous forme de réservoir peut être remplacé par la **figure (IV.4)**.

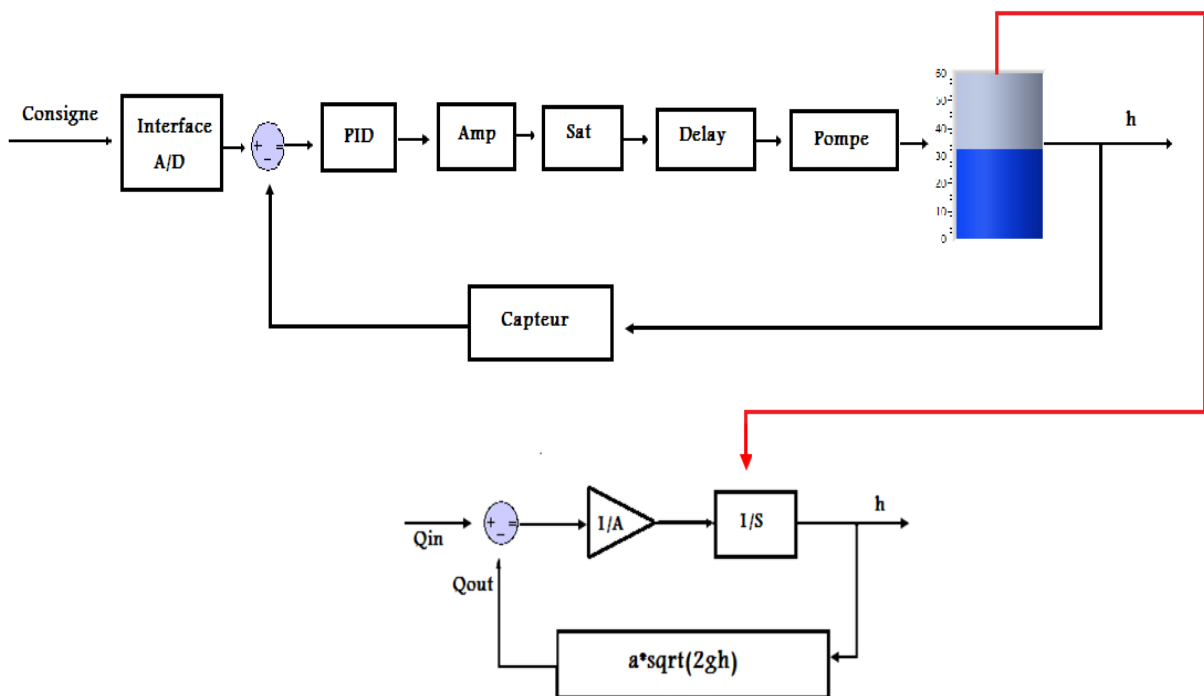


Figure IV.5 : schéma fonctionnel du système de commande.

Afin de concevoir ce système, il est nécessaire de disposer des fonctions de transfert de l'amplificateur, de la pompe, de l'interface numérique/analogique et du capteur de niveau d'eau. Malheureusement, ces données ne sont pas faciles à recueillir, donc certaines hypothèses concernant ces fonctions de transfert sont faites.

Les hypothèses sont les suivantes :

- *Fonction de transfert supposée de la pompe :*

$$\text{F.T de la pompe} = \frac{0.008}{s+0.016}$$

Où F.T la fonction de transfert.

- *Fonction de transfert supposée de l'interface numérique/analogique :*

$$\text{F.T de A/D} = 0.1$$

Cela signifie, l'interface convertit la valeur donnée à une tension qui est un dixième de la valeur d'entrée, si l'entrée de l'utilisateur 3 (m) la sortie de l'interface sera 0,3 (v).

- *Fonction de transfert supposée de capteur :*

$$\text{F.T de Capteur} = 0.1$$

Cela signifie, le capteur convertit la valeur donnée à une tension qui est un dixième de la valeur d'entrée, si le niveau de l'eau 2 (m) la sortie de capteur sera 0,2 (v).

- *Fonction de transfert supposée de l'amplificateur :*

$$\text{F.T de l'amplificateur} = 100$$

Cela signifie que l'amplificateur augmente la tension entrante de 100 fois, il ne peut pas produire une tension négative (cela signifie que la pompe ne peut pas aspirer l'eau).

IV.4 Les éléments de l'interface et ses calls back :

La **figure (IV.6)** présente les éléments de l'interface globalement :

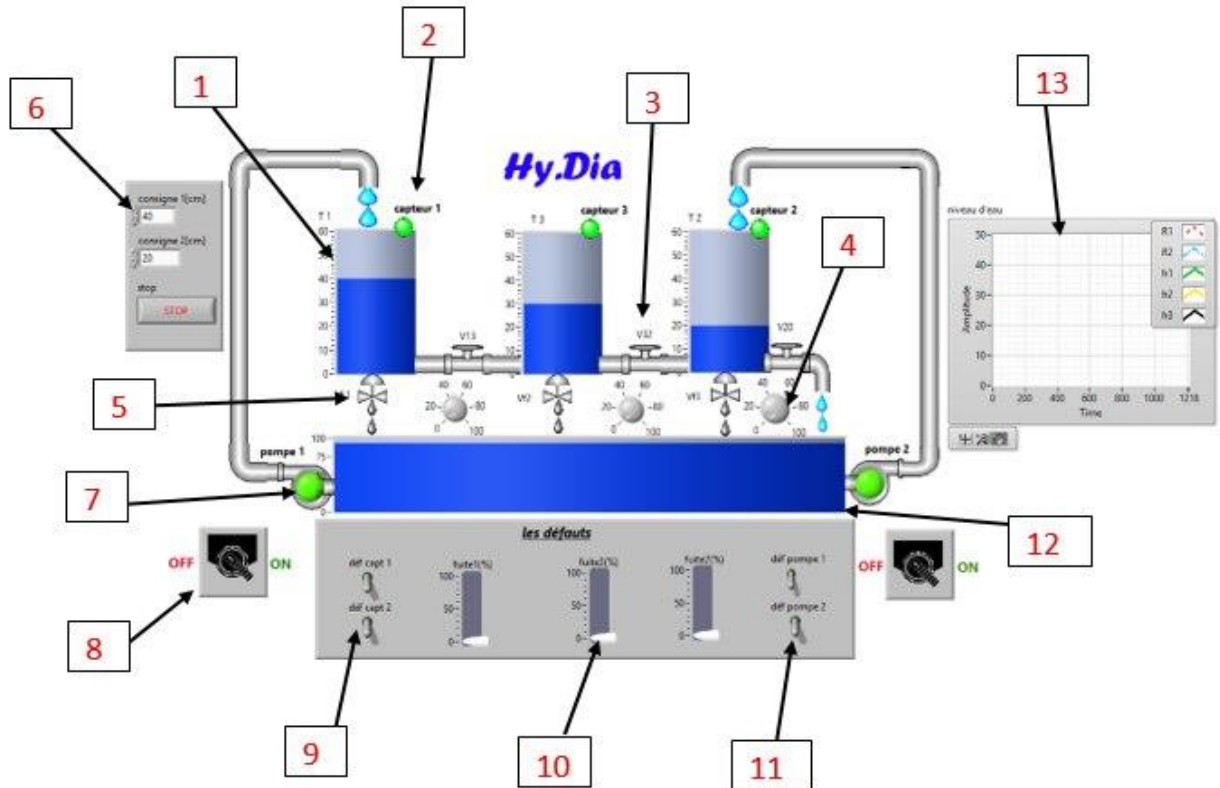


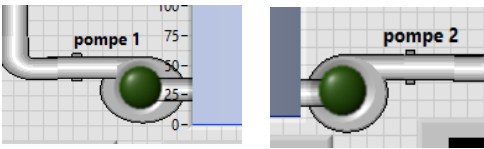
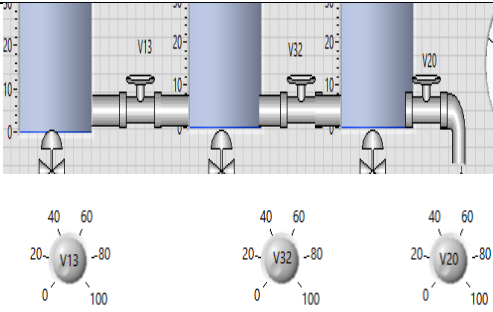
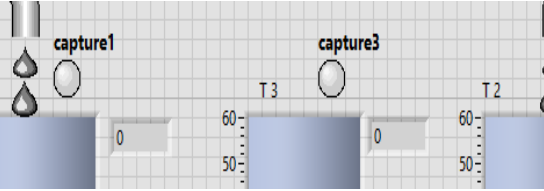


Figure IV.6 : Les possibilités de l'interface

1. Réservoir 1	8. Bouton marche/arrêt de la pompe 1
2. Capteur 1	9. Défaut capteur 2
3. Vanne de communication (V32)	10. Fuite dans le réservoir 2
4. La fermeture de la vanne V20 manuellement réglable	11. Défaut pompe 2
5. Vanne de fuite (Vf1)	12. Réservoir tampon
6. Consigne 1	13. Les graphes de niveau d'eau avec les consignes
7. Pompe 1	

Le **tableau (IV.1)** représente les différents éléments de l'interface et les actions que l'utilisateur peut faire avec certain élément.

Les éléments de l'interface	Calls back
	<p>Les points de consigne :</p> <ul style="list-style-type: none"> – Consigne1 : représente le niveau du réservoir 1 – Consigne 2 : représente le niveau du réservoir 2.
	<p>Un bouton marche / arrêt de la pompe.</p>
	<p>Deux pompes 1et 2 pour le remplissage de fluide dans les réservoirs 1 et 2 respectivement.</p>
	<p>Les conduites de communication :</p> <ul style="list-style-type: none"> – Les valves v13 et v32 permettent le passage de fluide entre les réservoirs T1 et T2 sont manuellement réglables par des contrôleurs du pourcentage. – La valve v20 permet l'écoulement de l'eau en dehors des réservoirs cylindriques vers le réservoir tampon
	<p>Les capteurs 1,2 et 3 pour mesurer les niveaux du liquide dans T1, T2 et T3, respectivement.</p>

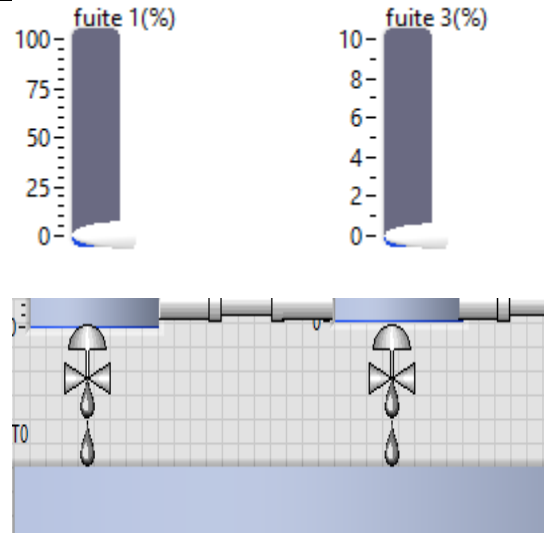
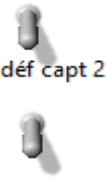


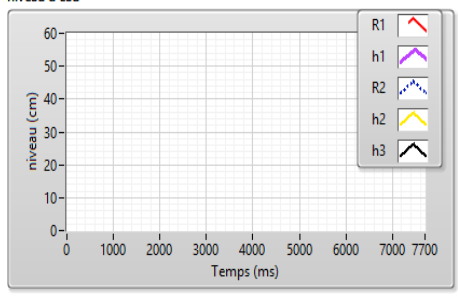
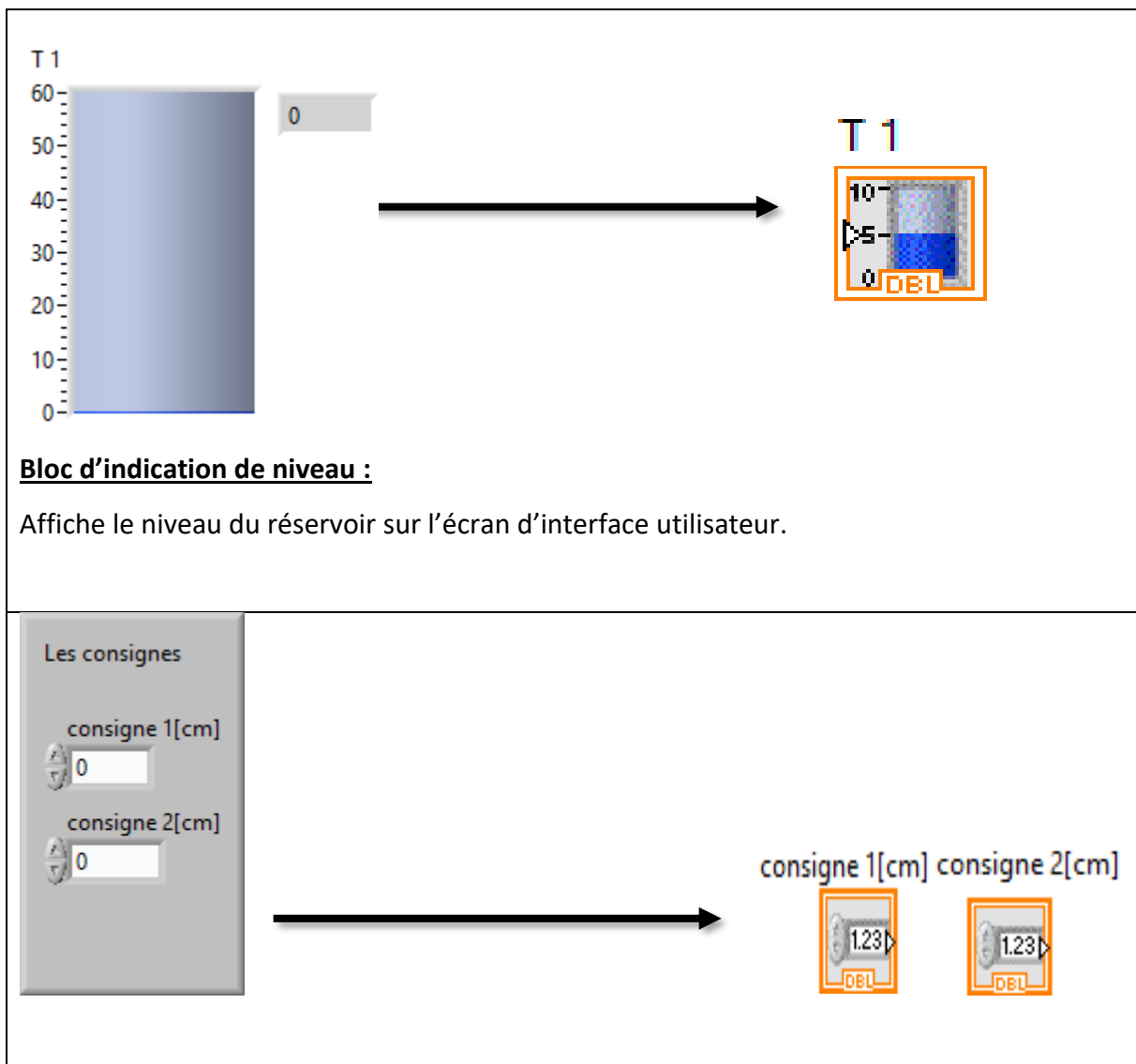
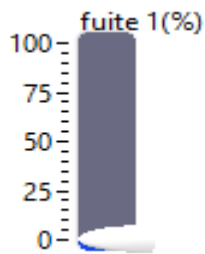
	<p>Le système est équipé de vannes V_{f1}, V_{f2} et V_{f3} appelés vannes de fuites dont le rôle est de simuler des fuites dans les réservoirs correspondants pour produire des défauts sont manuellement réglables par des contrôleurs du pourcentage.</p>
<p>déf capt 1</p>  <p>déf capt 2</p>	<p>Les contrôleurs de défauts capteurs : « déf capt1 » et « déf capt2 » Pour produire des défauts sur les capteurs correspondants.</p>
<p>déf pompe 1</p>  <p>déf pompe 2</p>	<p>Les contrôleurs de défauts pompes : « déf pompe1 » et « déf pompe2 » Pour produire des défauts sur les pompes correspondants.</p>
<p>stop</p> 	<p>Un bouton STOP pour l'arrêt du système.</p>
<p>niveau d'eau</p> 	<p>Waveform chart : The waveform chart Le graphique de forme d'onde est un type spécial d'indicateur numérique qui affiche un ou plusieurs tracés de données généralement acquises à un taux constant.</p>

Tableau IV.1 : Les éléments de l'interfaces et ses calls back.

IV.5 La communication entre l'interface et les programmes

Sans communication entre le programme et l'interface, cette dernière serait figée, une simple image. La communication entre les programmes et l'interface se fait dans les deux sens. L'utilisateur peut introduire des informations ou des actions sur l'interface que les call back transmettront aux programmes, ces derniers vont prendre en considération les différentes actions de l'utilisateur et apporter les changements nécessaires dans l'interface graphique. Cela donne du dynamisme et une interactivité à l'application entière. Le **tableau (IV.2)** illustre quelques exemples de communication entre les éléments de l'interface et les différents programmes ou les différentes parties du programme d'arrière-plan :





Contrôle numérique :

Pour saisir ou modifier la valeur d'un contrôle numérique, on peut utiliser les flèches d'incrément ou entrer une valeur avec l'outil texte.



déf capt 1



déf capt 2



déf pompe 1

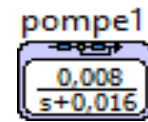
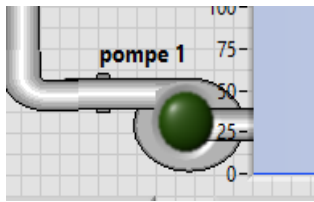


déf pompe 2



Contrôles booléens :

A utiliser pour saisir et afficher des valeurs binaires. Les objets booléens les plus communs sont les interrupteurs à bascules et les diodes LED.



Fonction de transfert :

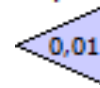
Exigences : *Control Design and Simulation Module*

Met en œuvre un modèle de système sous forme de fonction de transfert. Vous définissez le modèle système en spécifiant le numérateur et le dénominateur de l'équation de la fonction de transfert de la pompe.

capteur 1



capteur 1



Fonction Gain :

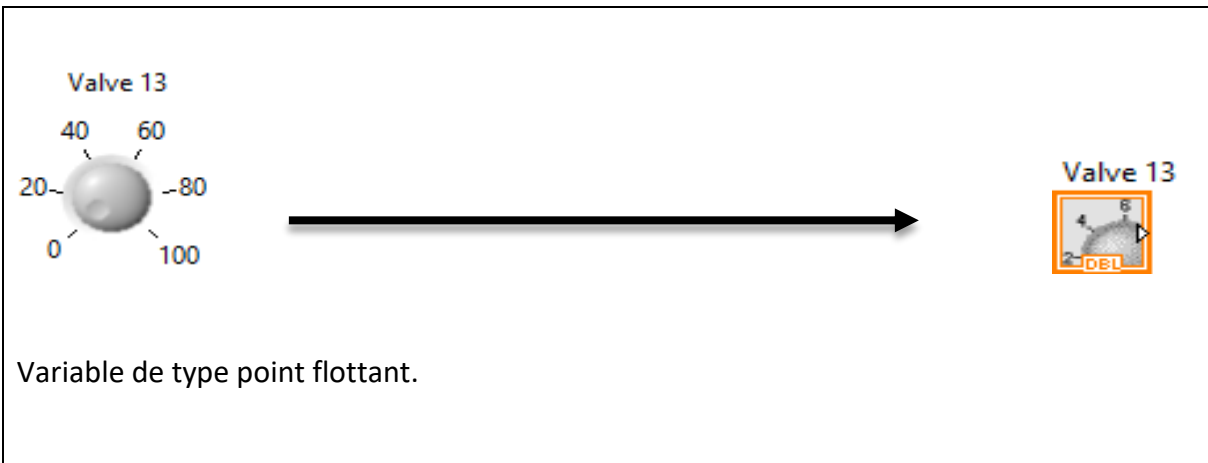
Exigences : *Control Design and Simulation Module*

Multiple l'entrée que vous spécifiez par le gain.

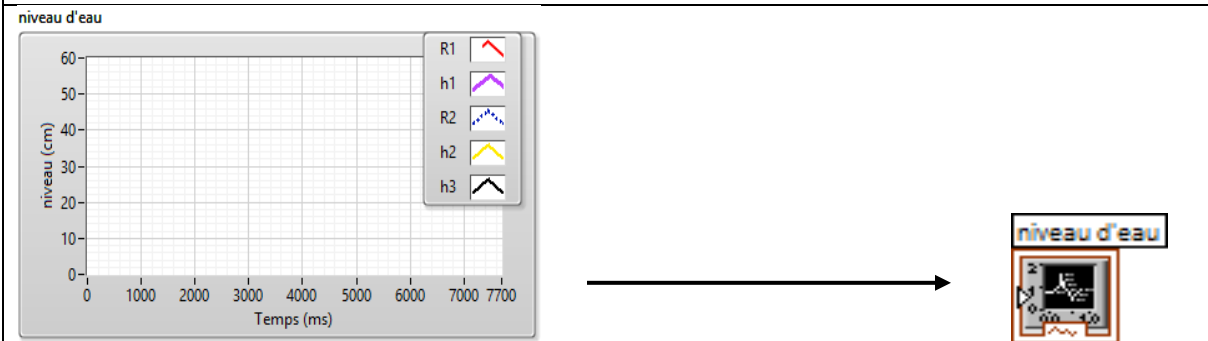


DSC Module :

DSC intègre des contrôles 2D et 3D pour les applications industrielles (tuyau, vannes, ventilateurs, réservoirs) et différents graphes (visualisation des limites, statistique). DSC fournit également une bibliothèque d'objet dans « **Tools/DSC module/Image Navigator** » qui peuvent être utilisés pour créer des contrôles personnalisés ou de simples décorations.



Variable de type point flottant.



Waveform Charts :

Utilisez la fonction Build Waveform (Analog Waveform) pour tracer le temps sur l'axe des x du graphique et utilisez automatiquement l'intervalle correct entre les marqueurs sur l'échelle des x du graphique. Une forme d'onde qui spécifie t0 et un tableau Y à un seul élément est utile pour tracer des données qui ne sont pas échantillonnées uniformément parce que chaque point de données a son propre horodatage.

Tableau IV.2 : Eléments de l'interface et leurs équivalents dans le programme.

❖ **La programmation d'un seul réservoir :**

Dans ce système (la figure IV.7), qui est le réservoir d'eau, le point de consigne (niveau désiré) est introduit (édité), puis une interface numérique/analogique convertit cette valeur en tension, puis la valeur sortante du capteur (pour mesurer la hauteur de l'eau) est soustraite de la sortie de l'interface numérique/analogique. Le résultat de la soustraction (erreur) est fourni à un contrôleur. Le résultat du contrôleur (qui est aussi une certaine tension) est

amplifié (car il devrait y avoir assez de puissance pour faire fonctionner la pompe). La tension amplifiée est reçue par la pompe et par conséquent un débit de pompage de l'eau dans le fond du réservoir qui draine l'eau. Le débit d'écoulement est fonction de la hauteur de l'eau dans le réservoir.

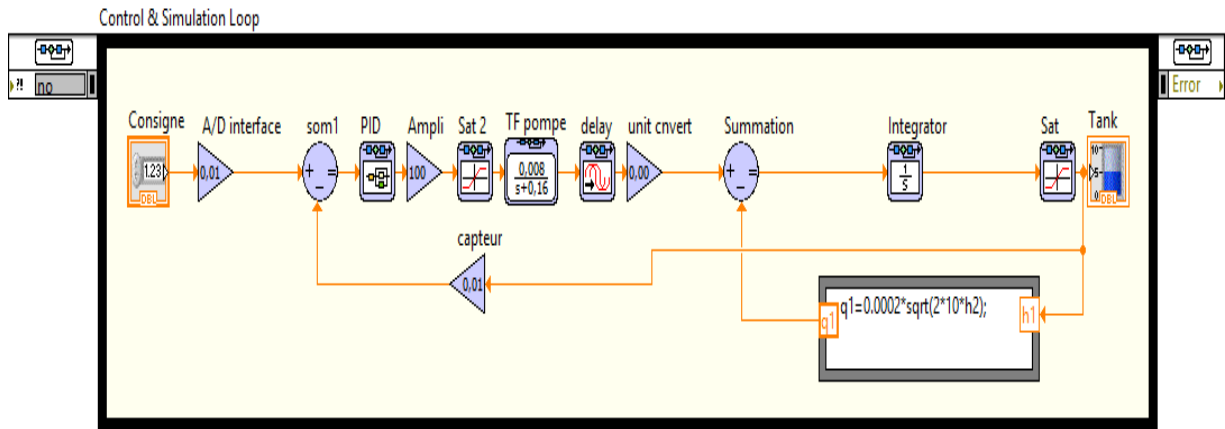


Figure IV.7 le programme correspond à un seul réservoir

La hauteur de l'eau est mesurée par un capteur de pression différentielle. Ce capteur convertit la hauteur actuelle de l'eau en tension. Cette valeur est envoyée à l'entrée de la chaîne pour calculer l'erreur. Cependant, la pompe ne fonctionne pas immédiatement après avoir reçu l'électricité. Il devrait qu'un certain temps s'écoule après avoir reçu l'électricité et commence à pomper.

En outre, la tension sortant de l'amplificateur ne peut pas être supérieure à une certaine valeur et elle ne peut pas être inférieure à zéro. La même condition s'applique au débit sortant de la pompe et au débit le plus élevé de l'eau. Le débit d'entrée vers le réservoir ne peut pas être négatif ni supérieur à la capacité de la pompe. Le niveau d'eau ne peut être inférieur à zéro ni supérieur à la hauteur du réservoir. Une telle restriction (saturation) est appelée non linéarité.

IV.6 Fonction de Hy.dia, Exemples commenté

L'application Hy.Dia via son interface permet plusieurs tâches qui peuvent être listées comme suit :

IV.6.1 La commande, suivi et changement de la consigne :

Comme il est connu, un des intérêts majeurs de la régulation est que le système régulé doit suivre l'évolution de la consigne avec une certaine précision, plus ou moins rapidement. Afin de traiter cet aspect, nous allons observer dans cette partie les réponses du système régulé à des échelons de consigne. En considérant $R_1(t)$ et $R_2(t)$, les signaux de références, h_1 et h_2 les deux sorties qui indiquent le niveau d'eau dans les réservoirs 1 et 2. La **figure (IV.8)**, montre les réponses du système.

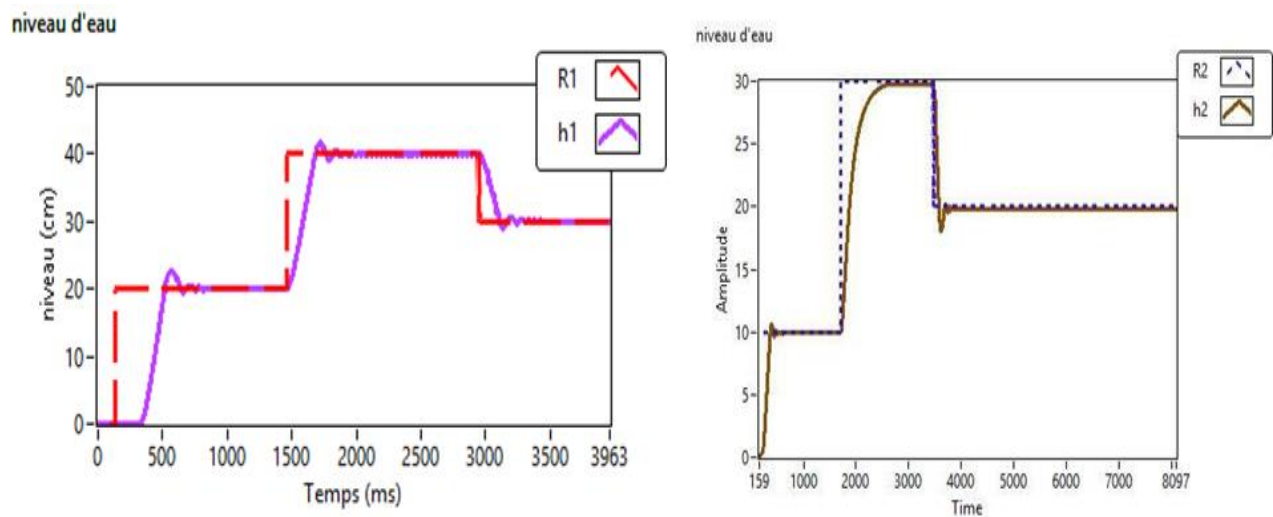


Figure IV.8 : Suivi de consigne pour le niveau du réservoir 1 et 2.

❖ Observations :

Sortie : la **figure (IV.8)** montre le suivi de consigne pour les niveaux des réservoirs 1 et 2
On applique trois consignes différentes :

- $R_1 = 20 \text{ cm}, 40 \text{ cm}, 30 \text{ cm}$ pour les réservoirs 1

- $R_2 = 10 \text{ cm}, 30 \text{ cm}, 20 \text{ cm}$ pour les réservoirs 2

On constate que les sorties h_1 et h_2 rejoignent rapidement les consignes avec un petit dépassement. Ceci confère à cette boucle une précision dynamique acceptable.

IV.6.2 Les défauts dans l'interface :

L'application donne la possibilité de configurer et d'actionner des défauts pendant la simulation tout en répercutant leurs effets respectifs sur le système et son apparence sur l'interface graphique.

IV.6.2.1 Les défauts dans le réservoir 1 :

Trois types de défauts sont pris en charge par l'application Hy.Dia. Il s'agit de : 1) défaut système : fuite et bouchage, 2) défaut capteur et 3) défaut actionneurs qui peuvent se manifester au niveau des réservoirs. Les détails de ces défauts peuvent être listés comme suit :

IV.6.2.1.1 Défaut système :

Fuite dans le réservoir 1 : Supposons que q_{f1} est le débit de fuite qui se retranche du débit total du premier réservoir pour causer une fuite de 30 %, de sorte que le nouveau bilan massique dans le réservoir 1 donne la relation suivante :

$$S \cdot \frac{dh_1}{dt} = q_1 - q_{13} - q_{f1} \quad (\text{IV.6})$$

$$\text{Où : } q_{f1} = C_{13f} \cdot \sqrt{h_1 - h_3} \quad (\text{IV.7}) \quad \text{telque} \quad C_{13f} = \frac{C_{13} \cdot 30\%}{100\%} \quad (\text{IV.8})$$

On applique une fuite du 10% au niveau du réservoir 2 et on obtient les résultats dans la **figure (IV.9)** :

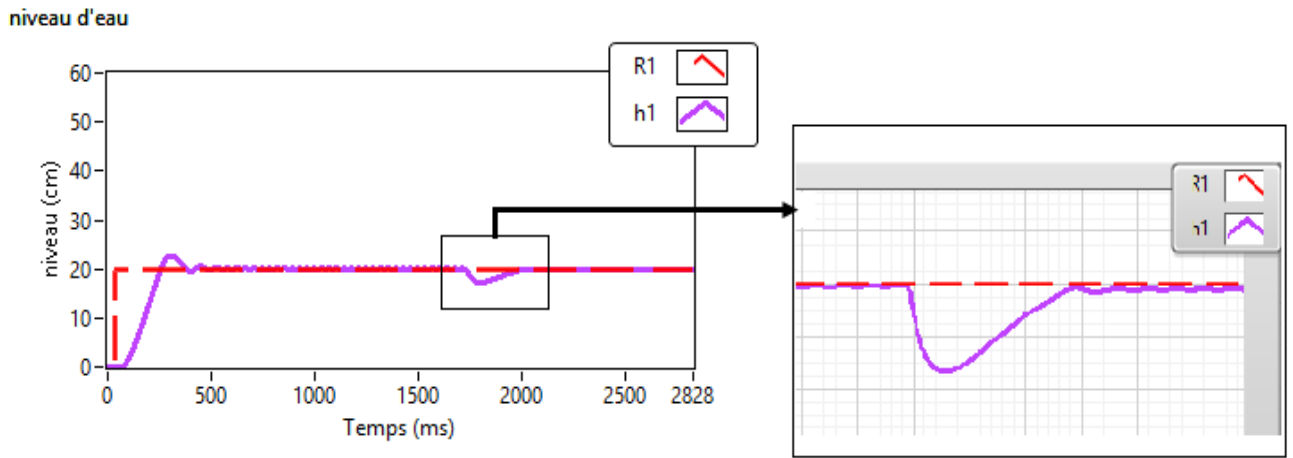


Figure IV.9 : fuite dans le réservoir 1.

❖ **Observations :**

Sortie : On constate que la sortie h_1 rejoint rapidement avec très petit dépassement la consigne, la fuite apparait sous forme d'écart statique rapidement compensé, ce qui prouve le peu de sensibilité de la sortie h_1 vis-à-vis de la fuite.

b. Bouchage dans le réservoir 1 :

Supposons que q_{b1} est le débit de bouchage qui est ajouté au débit total du premier réservoir pour créer un bouchage de 60 %, de sorte que le nouveau bilan massique du réservoir 1 fournit la relation suivante :

$$S \cdot \frac{dh_1}{dt} = q_1 - q_{13} + q_{b1} \quad (\text{IV.9})$$

$$\text{Où : } q_{b1} = C_{13b} \cdot \sqrt{h_1 - h_3} \quad (\text{IV.10}) \text{ telque } C_{13b} = \frac{C_{13} \cdot 60\%}{100\%} \quad (\text{IV.11})$$

On applique un bouchage du 10% au niveau du réservoir 1 et on obtient les résultats dans la **figure (IV.10)** :

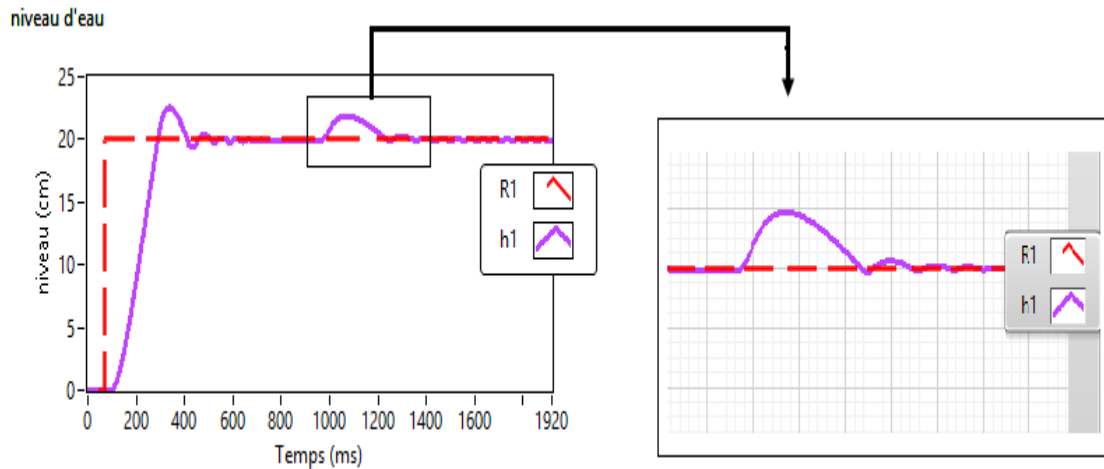


Figure IV.10 : La réponse à un bouchage dans le réservoir 1.

❖ **Observations :**

Sortie : On constate que la sortie h_1 rejoint rapidement avec petit dépassement la consigne, le bouchage apparait sous forme d'un dépassement de consigne rapidement compensé.

IV.6.2.1.2 Défaut capteur :

Si on applique un défaut capteur au niveau du réservoir 1, on obtient le résultat illustré par la figure (IV.11).

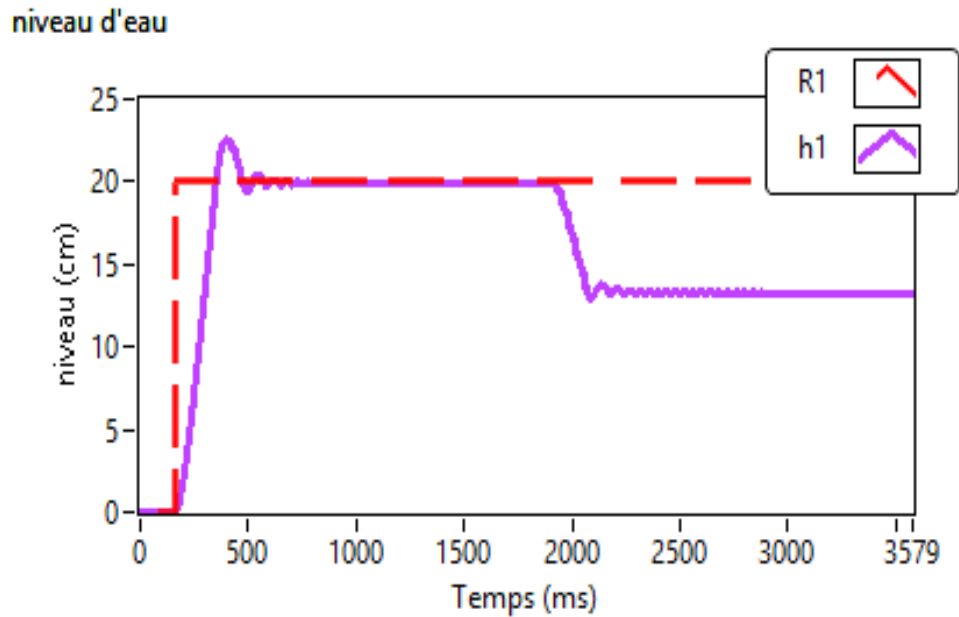


Figure IV.11 : La réponse à un défaut capteur dans le réservoir 1.

❖ **Observations :**

Sortie : On constate que la sortie h_1 diminue sous la consigne désirée, le défaut provoqué une perturbation du niveau d'eau car le capteur défectueux a donné des valeurs incorrectes sur le niveau d'eau.

IV.6.2.1.3 Défaut actionneur :

Un défaut actionneur au niveau de la pompe de réservoir 1 donne le résultat illustré par la figure (IV.12) :

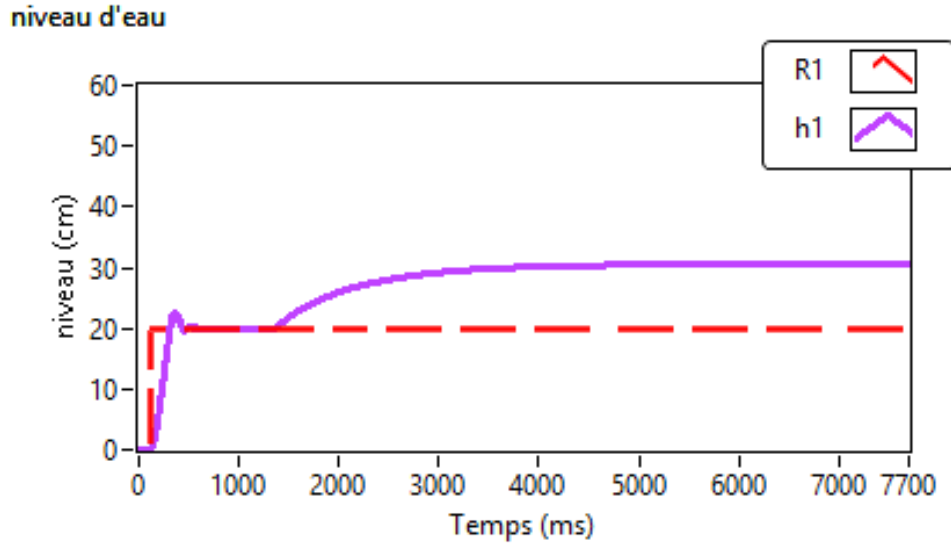


Figure IV.12 : La réponse à un défaut sur la pompe dans le réservoir 1.

❖ **Observations :**

Sortie : On constate que la sortie h_1 augmente et fini par dépasser puis s'éloigner de la consigne désirée, le défaut provoque une perturbation du niveau d'eau car la pompe défectueuse donne un débit élevé à la commande de régulateur.

IV.6.2.2 Les défauts dans le réservoir 2 :

On applique 3 défauts (défaut système : fuite et bouchage, défaut capteur et défaut actionneur) dans le réservoir 1 et on remarque les résultats obtenus à chaque défaut :

IV.6.2.2.1 Défaut système :

a. Fuite dans le réservoir 2 :

Supposons que q_{f2} est le débit de fuite qui se retranche du débit total du deuxième réservoir pour causer une fuite de 30 %, de sorte que le nouveau bilan massique dans le réservoir 2 donne la relation suivante :

$$S \cdot \frac{dh_2}{dt} = q_2 + q_{32} - q_{20} - q_{f2} \quad (IV.12)$$

Où :

$$q_{f2} = C_{20f} \cdot \sqrt{h_2} \quad (IV.13) \quad \text{telque} \quad C_{20f} = \frac{C_{20} \cdot 30\%}{100\%} \quad (IV.14)$$

On applique une fuite du 10% au niveau du réservoir 2 et on obtient les résultats suivants (figure (IV.13)) :

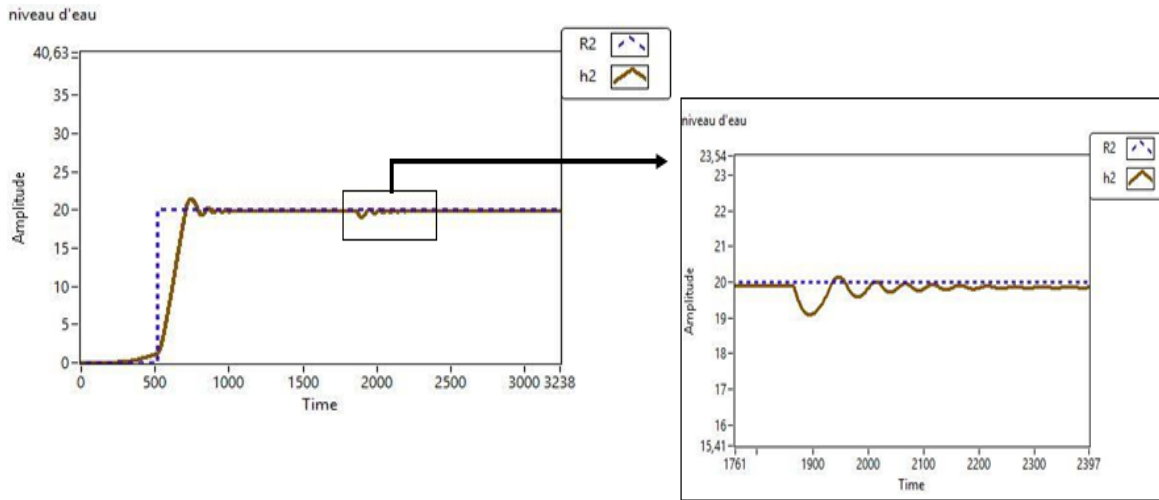


Figure IV.13 : La réponse à une fuite dans le réservoir 2.

❖ **Observations :**

Sortie : On constate que la sortie h_2 rejoint rapidement avec très petit dépassement la consigne, la fuite apparait sous forme d'écart statique rapidement compensé et sans créer de dépassement.

b. Bouchage dans le réservoir 2 :

Supposons que q_{b2} est le débit de bouchage qui est ajouté au débit total du deuxième réservoir pour créer un bouchage de 60 %, de sorte que le nouveau bilan massique du réservoir 2 fournit la relation suivante :

$$S \cdot \frac{dh_2}{dt} = q_2 + q_{32} - q_{20} + q_{b2} \quad (IV.15)$$

Où :

$$q_{b2} = C_{20b} \cdot \sqrt{h_2} \quad (\text{IV.16}) \quad \text{telque} \quad C_{20b} = \frac{C_{20} \cdot 60\%}{100\%} \quad (\text{IV.17})$$

On applique un bouchage de 10% au niveau du réservoir 2 ce qui se traduit par les sorties illustrées par la **figure (IV.14)** :

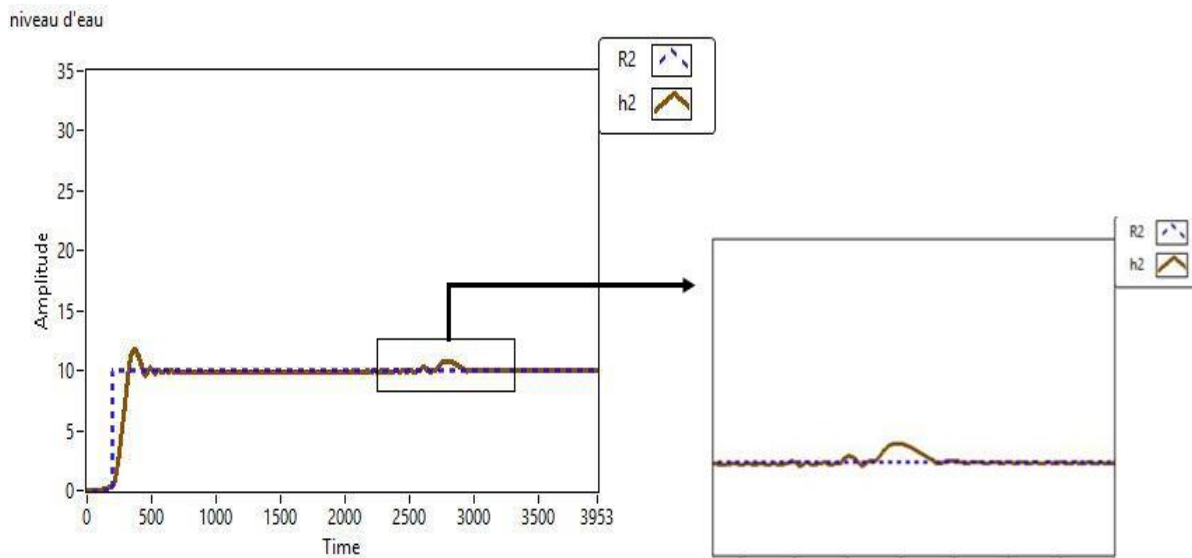


Figure IV.14 : La réponse à un bouchage dans le réservoir 2.

❖ **Observations :**

Sortie : On constate que la sortie h_2 rejoint rapidement, avec petit dépassement, la consigne. Le bouchage apparait sous forme d'un dépassement de consigne rapidement compensé.

IV.6.2.2.2 Défaut capteur :

On applique un défaut au niveau du capteur du réservoir 2 ce qui permet d'obtenir le résultat illustré par la **figure (IV.15)** :

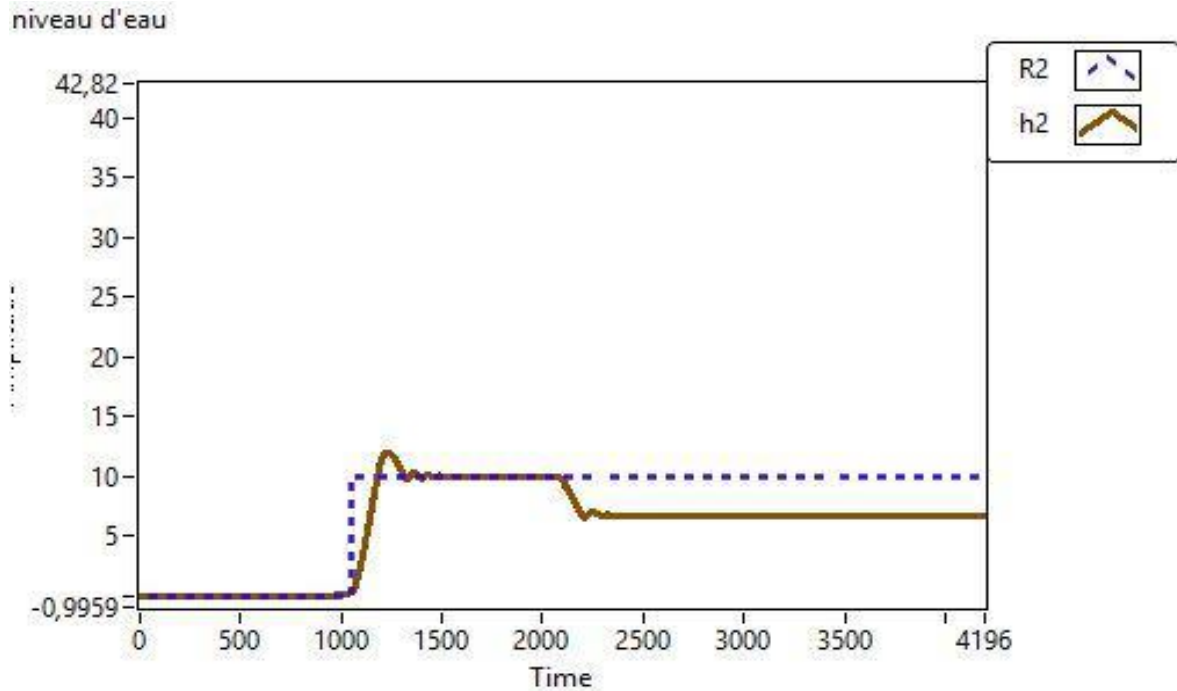


Figure IV.15 : La réponse à un défaut capteur dans le réservoir 2.

❖ **Observations :**

Sortie : On constate que la sortie h_2 diminue sous la consigne désirée, le défaut provoque une perturbation du niveau d'eau car le capteur défectueux donne des informations (valeurs) incorrectes sur le niveau d'eau.

IV.6.2.2.3 Défaut actionneur :

Si on applique un défaut au niveau de la pompe de réservoir 2, on obtient le résultat illustré dans la **figure (IV.16)**.

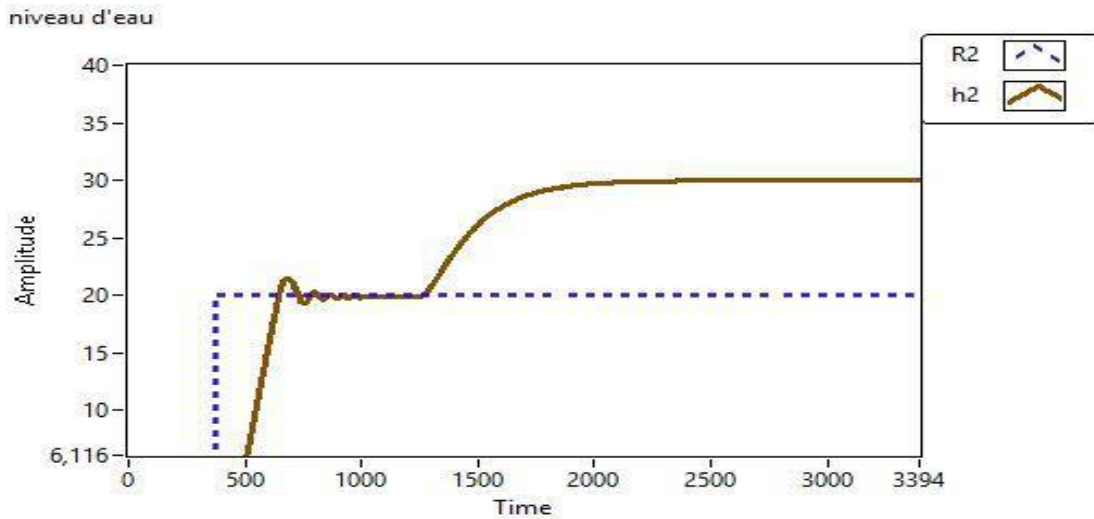


Figure IV.16 : La réponse à un défaut sur la pompe dans le réservoir 2.

❖ **Observations :**

Sortie : On constate que la sortie h_2 dépasse la consigne désirée, le défaut provoqué une perturbation du niveau d'eau car la pompe défectueuse a donné un débit élevé à la commande de régulateur.

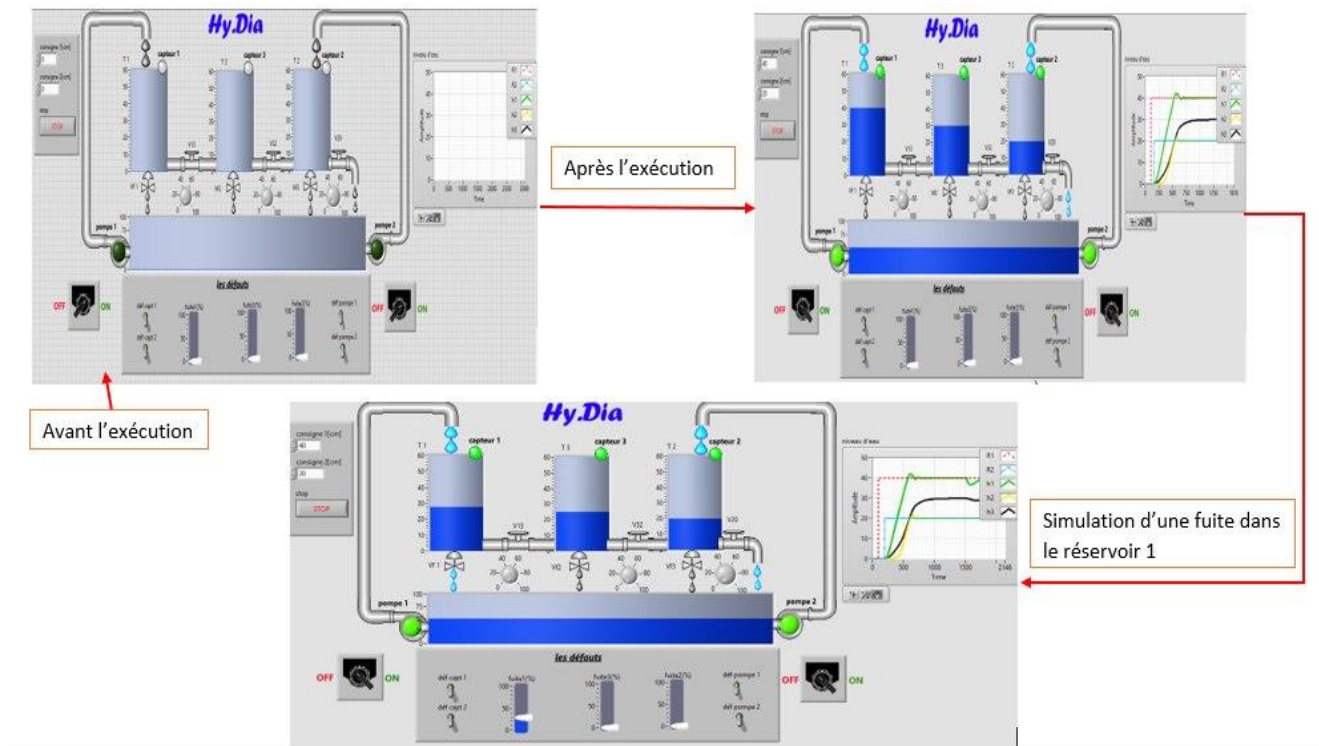


Figure IV.17 : Exemple de capture d'écran illustrant le fonctionnement de l'interface.

IV.7 Conclusion :

Hy.Dia est une application software qui se présente comme un premier pas vers un outil de TP de l'unité d'enseignement Diagnostic des systèmes automatique. Elle est conçue autour d'un système hydraulique à trois cuves. Cet outil peut être utilisé pour monter plusieurs TP de cette unité d'enseignement.

Hy.Dia permet, grâce à sa logique de programmation, de programmer plusieurs défauts séparément ou simultanément. De choisir leur type, leur amplitude et leurs fréquences. Elle prend aussi en considération trois types de défauts ; système, capteur et actionneur. Ceci permet son exploitation pour la génération des résidus moyennant différentes méthodes. Et donne donc, plusieurs possibilités pour les enseignants chargés de ce TP et plusieurs opportunités d'apprentissage, de test et d'expérimentation pour les étudiants.

Conclusion Générale

Tout système automatique peut présenter des anomalies de fonctionnement durant sa durée de vie. Prévenir ces défaillances est une tâche très importante, elle relève de la sécurité du système de l'humain et de l'investissement. Le diagnostic des systèmes automatiques est devenu axe à part entière de l'automatisme. Le diagnostic devient de plus en plus présent dans les systèmes automatiques. Son but n'est plus de détecter les pannes, mais les premiers signes des défauts.

Dans la formation Master automatique, le diagnostic des systèmes automatiques est enseigné à travers deux unités d'enseignement, l'une correspond aux cours avec ses TD et la deuxième unité concerne le TP qui permet aux étudiants d'appliquer et tester les différentes notions et les différentes méthodes de diagnostic apprises dans la première unité d'enseignement. Le manque de moyens pour le TP, a fait naître le besoin de développer un outil personnalisé, une application software en premier temps qui pourrait être réalisé pour offrir aux étudiants la possibilité de travailler sur un système le plus fidèle possible pour appliquer le savoir acquis dans ce domaine. Ainsi, un cahier de charge a été établi en fonction des besoins suivant :

- 1) La solution doit être construite autour d'un système automatique ;
- 2) Le système automatique doit être de complexité moyenne à élever ;
- 3) La solution doit être intuitive pour attirer l'attention de l'étudiant et faciliter sa compréhension et son utilisation ;
- 4) La solution doit être orientée vers la possibilité de programmer (choisir) des défauts pour ensuite appliquer les méthodes de diagnostic ;
- 5) La solution doit prendre en considération aux moins deux des trois types de défauts avec la possibilité de régler leurs propriétés ;

CONCLUSION GENERALE

- 6) La solution doit être pensée et conçue de manière à se transformer et d'évoluer, d'une manière lisse, vers une maquette de TP réelle.

Ainsi, la première étape de notre travail, pour répondre à ce cahier de charge et à ces exigences consistait, à choisir un système automatique dont le fonctionnement soit facile à comprendre mais avec un degré de complexité qui pourrait mettre en exercice les notions apprises et mettre en défis les étudiants. Notre choix s'est porté sur un système hydraulique à trois cuves. Ce qui a conduit au développement de la solution Hy.Dia qui est une application software permettant de construire plusieurs TP pour le diagnostic, en d'autres termes la détection et l'identification des défauts dans un système automatique hydraulique. Hy.Dia a été développé sous LabVIEW, et sa maquette réelle de TP conçue et financièrement évaluée. En considérant les prix actuels des différents composants et le coût de la main d'œuvre artisanale, son coût est estimé aux alentours de 400.000 DA. Ce qui fait qu'un labo équipé de 10 maquettes de TP pour un groupe de 20 élèves coûterait 400.000 DA, ce qui représente une somme raisonnable et coûte à une très grande valeur ajoutée pour les étudiants et l'université.

Le présent travail, a été réalisé dans des conditions spéciales liées à la situation sanitaire qui a touché le monde et l'Algérie soudainement, nous avons subi le confinement sans y être préparé. Cela a largement affecté l'avancement de notre travail. Néanmoins, le résultat obtenu reste satisfaisant, il donne le début d'une solution plus complexe. Comme suite à ce travail, trois perspectives restent à être explorées et concrétisées : 1) Le montage des différents TP avec leur manuel et 2) réaliser la maquette.

Bibliographie

- [1] Y. Kourd. Génération des résidus par les outils de l'intelligence artificielle pour le diagnostic des systèmes complexes. PhD thesis, Thèse de Doctorat, Université Badji Mokhtar, Annaba, 2012.
- [2] DJ.Abdelaghani, Diagnostic des Systèmes Non Linéaire A Base d'Observateur, Thèse de Doctorat, faculté des Sciences de l'Ingéniorat, Université Badji Mokhtar , Annaba,2017.
- [3] F. Sallem. Détection et Isolation de Défauts Actionneurs Basées sur un Modèle de l'Organe de Commande. PhD thesis, Thèse de doctorat, Université Toulouse III, 13 septembre 2013.
- [4] H. Hamdi. Multi-Model Approach for State Observation and Fault Diagnosis dedicated to Nonlinear Descriptor Systems. PhD thesis, Thèse de doctorat, École Supérieure des Sciences et Techniques de Tunis, 24 novembre 2012.
- [5] G. Graton. Diagnostic des systèmes à l'aide d'observateurs à mémoire finie. Application au Common Rail. PhD thesis, Thèse de doctorat, Université d'Orleans, 14 décembre 2005.
- [5.5] M. R. Zemouri. Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance. PhD thesis, Thèse de Doctorat, Université de Franche-Comté, 2003.
- [6] H. Baikeche.Diagnostic des systèmes linéaires en boucle fermée, Thèse doctorat de l'Institut National Polytechnique de Lorraine, spécialité automatique et traitement de signal
- [7] A. Akhenak. Conception d'observateurs non linaires par approche multi modèle, application au diagnostic, Thèse de Doctorat, Ecole doctorale IAEM de Lorraine, 2004
- [8] N.S.Boukhalifa.Synthèse d'observateurs non linéaires , Application au diagnostic de défauts. Thèse magister, Université Mouloud Mammeri, Tizi- Ouzou.
- [9] R.J.Pattan. Fault Detection and Diagnosis in Aerospace Systems using AnalyticalRedundancy, Computing and Control Engineering, Vol.2, pp127-136, 1991.

BIBLIOGRAPHIE

- [10] M.L.Leuschen, I.D. WALKER et J.R. Cavallaro. Nonlinear Analytical Redundancy for Fault Detection, IEEE Transactions on Automatic control,2002
- [11] P.M Frank and B Köppen-Seliger. New developments using AI in fault diagnosis, Engineering.Applic. Artif. Intell. vol. 10(1), pages: 3-14, 1997.
- [12] S. Dash, Venkatasubramanian, V. Challenges in the industrial applications of fault diagnostic systems. Proceedings of the conference on Process Systems Engineering Comput. &Chem, 2000.
- [13] R.Isermann, Fault-diagnosis systems: an introduction form fault detection to faul tolerance. Springer, 2006.
- [14] V. Venkatasubramanian, R.Rengaswamy, Yin, K., Kavuri, S. N. A review of process fault detection and diagnosis,Computers & Chemical Engineering vol. 27, pp. 293-346,2003
- [15] M.Amine, A.Mohamed, “Commande Multi boucle des Systèmes Multivariables : Applications au Dispositif Expérimental des Trois Cuves DTS-200” , Faculté de génie électrique et d’informatique , université de Tizi-Ouzou, 2010.
- [16] JOIN. Cédric. « *Diagnostic des systèmes non linéaires Contribution aux méthodes de découplage* ». Thèse de Doctorat, Université Henri Poincaré, Nancy I, 2002.
- [18] M.L.Fass , Cours “ Capteurs et Actionneurs ”, Département d’Electronique, Spécialité Automatique ,Université de blida 1 , 2018.
- [21] D.Amine ,C.Oussama, “Application à la régulation en cascade de niveau d’eau dans une cuve parfaitement agitée”, filière génie des procédés, Lauréat EMI, 2012.
- [22] N.Duro, R.Dormido , H.Vargas, S.Dormido-Canto, J.Sánchez, G.Farias,... & F.Esquembre, (2008). “ An integrated virtual and remote-control lab: The three-tank system as a case study”. *Computing in Science & Engineering*, 10(4), 50-59.
- [23] KH.Amel, H.F.Inchirah, “ Diagnostic de Défauts du Système Hydraulique Cinq Cuves ”, Faculté des nouvelles technologies de l’information et de la communication, université Kasdi Merbah-Ouergla, 2017.

BIBLIOGRAPHIE

[24] S.Ikediche. *Etude et planification d'un système de supervision (SCADA) sous le logiciel LabVIEW*. Mémoire de Master. Université Djilali Bounaama Khemis Miliana, 2018.

[25] T. TAŠNER, D. LOVREC, F. TAŠNER [et al.]. COMPARISON OF LabVIEW AND MATLAB FOR SCIENTIFIC RESEARCH. *ROMANIA : Tome X, 2012, p. 389-394*.

Webographie

[17] First Sensor AG. *Glossaire capteurs de pression*. Disponible sur : <<https://www.first-sensor.com/fr/glossaire/index.html>> (consulté le 11/10/2020).

[19] Picclick. *National Instrument PCI-6024 DAC Card*. Disponible sur : <<https://picclick.fr/National-Instruments-PCI-6024E-NI-DAQ-Card-Analog-Input-142534830100.html>> (consulté le 11/10/2020).

[26] H.Romaric. *Que faut-il préférer : LabVIEW ou le C ?* Le10 octobre 2014. <<https://www.developpez.com/actu/76199/Que-faut-il-preferer-LabVIEW-ou-le-C-Le-responsable-produit-LabVIEW-livre-son-analyse/>> (consulté le 11/10/2020).

[27] Mohsen. *WinCC or LabVIEW ?* Disponible sur : <<https://control.com/forums/threads/wincc-or-labview.19854/>>(consulté le 11/10/2020).