

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



Ecole Nationale Polytechnique
Département Automatique



Mémoire de Master en Automatique

Thème :

**Commande de la station CE117 par la
méthode de Lyapunov et H_∞ en utilisant
la pompe**

Moufek KHELIFI

Présenté devant le jury composé de :

Président:	H.CHEKIREB	Professeur à l'ENP
Rapporteur:	E.M.BERKOUK	Professeur à l'ENP
Examineur:	L.ABDELOUEL	Chargé de Cours à l'ENP

JUIN 2016

ENP 10, Avenue HassenBadi, BP. 182, 16200 El Harrach, Alger, Algérie

ملخص :

العنوان: تحكم في المحطة CE117 طريقة الخطو الرجعي (الباكستينغ) و اش انفيني باستعمال المضخة لعمل المنجز في المذكرة يتمحور على التحكم في المحطة TecQuipment Process Trainer CE117 باستعمال مسير صناعي ميرمج "سيمنس" من خلال برنامج STEP7 و WinCC. باستعمال مبرمج سيمنس S7 314 IFM قمنا بالتحكم بطريقة الخطو الرجعي في جميع أنظمة المحطة المتحكم فيها بالمضخة.

الكلمات المفتاحية : مسير صناعي ميرمج "سيمنس", برنامج "ستابسات", برنامج "وينسيبي فلكسيل", نموذج محطة ضخ المياه TecQuipment Process Trainer CE117, طريقة الخطو الرجعي (الباكستينغ), اش انفيني .

Abstract:

Subject: Control of the station CE117 Process Trainer with the Backstepping and H_{∞} method using the Pump.

The work presented in this paper is based primarily on the control of the station TecQuipment Process Trainer CE117 by programmable logic controller SIEMENS via the software STEP7 and WinCC.

Using the programmable logic controller S7 314 IFM, we regulated by Backstepping method all sub-system controlled by the proportional valve of the station CE117.

Key words: CE117 Process Trainer, Siemens PLC S7-313C, STEP7, WinCC, Backstepping method, H_{∞} .

Résumé:

Le travail présenté dans ce mémoire est basé essentiellement sur la commande de la station TecQuipment Process Trainer CE117 par un automate programmable SIEMENS par le biais des logiciels STEP 7 et WinCC.

À l'aide de l'automate S7 IFM 314, nous avons régulé par la méthode de backstepping et H_{∞} l'ensemble de sous-systèmes contrôlé par la pompe constituant cette installation.

Mots clés : CE117 Process Trainer, Siemens S7-300, WinCC, Step7, commande backstepping, H_{∞} .

Dédicace

Je dédie ce travail tout d'abord à mon père qui m'a tout appris, tant donné sans rien demandé en retour.

A ma très chère mère et grande mère sans lesquelles je ne serais pas l'homme que je suis.

A mes frères Amine et Adel

A mon ami et binôme de la vie Moufek sur qui j'ai toujours pu compter.

A mes amis et camarades avec qui j'ai passé de bons moments.

A tous ceux qui ont contribué de loin ou de prêt à notre travail.

Moufek KHELIFI

Remerciements

Je tiens à remercier dieu de m'avoir donné la force morale, physique et l'aide pour accomplir ce modeste travail.

Je tiens à remercier mon promoteur Pr. BERKOUK pour m'avoir accepté encadré et dirigé durant l'élaboration de ce travail ainsi que pour leur assistance et tous leurs conseils.

Je remercie chaleureusement les membres du jury pour l'honneur qu'ils m'ont fait en acceptant d'évaluer mon projet.

Je souhaite aussi remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, et en particulier, Nos professeurs d'Automatique qui nous ont encadrées auparavant et tous nos enseignants pour les connaissances qu'ils nous ont transmis, leur disponibilité et leurs efforts.

Je remercie chaleureusement ma familles, pour leurs soutien sans faille, leurs présence émotionnelle ainsi que les nombreux conseils qu'elles m'ont prodigué, et qui m'ont indéniablement permis de mener à bien ce travail.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail trouvent ici l'expression de notre sincère gratitude.

Table des matières

Liste des tableaux

Liste des figures

Chapitre01 : Description et identification

PARTIE A: Description de la station TecQuipment Process Trainer CE117...11

1.A.1. Description de la station TecQuipment Process Trainer CE11711

1.A.2. Les contraintes des différents systèmes de la station CE11713

1.A.3. Les caractéristiques techniques des différents éléments de la station CE11713

PARTIE B : Identification des différents systèmes de réglage commandés

par la pompe P214

1.B.1. Identification du système pompe - débit14

1.B.2. Identification du système pompe-niveau15

1.B.3. Identification du système pompe-pression.....16

Chapitre 2 : Description et synthèse de la commande par backstepping et H_{∞}

Introduction19

PARTIE A: Description des différentes commandes utilisées pour le contrôle

de la station19

2.A.1. La commande de Backstepping19

2.A.1.1. Théorie de Lyapunov19

2.A.1.2. Stabilité asymptotique.....19

2.A.1.3. Exemple d'application.....20

2.A.2. La commande H_{∞} 21

2.A.2.1. Théorème du petit gain21

2.A.2.2. Conditions de la stabilité robuste21

2.A.2.3. Formulation du problème standard H_{∞}22

2.A.2.4. Synthèse de la commande.....23

PARTIE B : Synthèse des commandes24

2.B.1. Système Pompe-débit24

2.B.1.1. Synthèse de la commande de backstepping24

2.B.1.2. Teste de robustesse et de performances25

1- La poursuite et le rejet de perturbation	25
2- La robustesse	25
2.B.2. Système Pompe-niveau	26
2.B.2.1. Commande par backstepping	29
2.B.2.2. Teste de robustesse et de performances	30
1- La poursuite et le rejet de perturbation	30
2- robustesse	30
2.B.2.3. La commande H_{∞}	31
1- Formulation du problème standard H_{∞}	31
2- Le choix des fonctions de pondérations	32
2.B.2.4. Teste des performances et de robustesse	33
2.B.3. Système Pompe-pression.....	34
2.B.3.1. Synthèse de la commande par backstepping	34
2.B.3.2. Teste de robustesse et de performances	35
Conclusion.....	35
 Chapitre 3: Implémentation des commandes sur le Step7 et supervision par le	
WinCC	
PARTIE A : Implémentation des commandes sur le Step7	37
3.A.1. Implémentation des commandes sur le Step7	37
PARIE B : Supervision et Archivage sous le WinCC	41
3.B.1. Supervision sous le WinCC.....	41
3.B.2. Archivage sous le WinCC	43
3.B.2.1. Les étapes d'archivage d'une variable	43
3.B.2.2. Application sur le système pompe-débit.....	45
Bibliographie.....	47

Liste des tableaux

Tableau 1.1 : les caractéristiques techniques des équipements de la station CE117.....13

Tableau 2.1 : la caractéristique statique et le gain du capteur de niveau.....26

Liste des figures

Figure1.1 : la station CE117.....	12
Figure 1.2 : Réponse réelle et identifiée du système pompe-débit en boucle ouverte.....	14
Figure 1.3 : Réponse réelle et identifiée du système pompe-niveau en boucle ouverte.....	16
Figure1.4 : Schéma qui présente le circuit fluide procédé pour le réglage de la pression.....	16
Figure 1.5 : Réponse réelle et identifiée du système pompe-pression en boucle ouverte....	17
Figure 2.1 : Schéma d'un système de commande bouclé.....	21
Figure 2.2 : La forme standard du problème H_{∞}	22
Figure 2.3 : choix des signaux pour le problème standard.....	23
Figure 2.4 : La réponse du système pompe-débit sous le WinCC.....	24
Figure 2.5 : La réponse du système pompe-débit avec perturbation sous le WinCC.....	25
Figure 2.6 : La réponse du système pompe-débit avec variations paramétrique sous le WinCC.....	25
Figure2.7 : Les fonctions d'appartenance de la variable floue.....	27
Figure2.8 : La réponse du système en boucle ouverte après la résolution de problème de non linéarité du capteur sous WinCC.....	28
Figure 2.9 : La réponse du système pompe-niveau sous le WinCC.....	29
Figure 2.10 : La réponse du système pompe-niveau avec perturbation sous le WinCC.....	30
Figure 2.11 : La réponse du système pompe-niveau avec variations paramétrique sous le WinCC...	30
Figure 2.12 : le schéma fonctionnel du système en boucle fermé.....	31
Figure 2.13 : La forme standard du problème H_{∞}	31
Figure 2.14 : La mise sous la forme standard du problème H_{∞}	31
Figure 2.15 : La réponse du système pompe-niveau sous matlab	32
Figure 2.16 : La réponse du système pompe-niveau sous WinCC	33
Figure 2.17 : La réponse du système pompe-niveau avec perturbation et variations paramétriques sous WinCC	33
Figure 2.18 : La réponse du système pompe-niveau sous WinCC	34
Figure 2.19 : La réponse du système pompe-niveau sous WinCC.....	35
Figure 2.20 : La commande appliquée au système pompe-niveau sous WinCC.....	35
Figure 3.1 : La création d'un fichier source.....	38
Figure 3.2 : Le choix du modèle de bloc.....	38
Figure 3.3 : Programme sous le langage structuré dans un fichier source	39
Figure 3.4 : La fenêtre des erreurs de compilation.....	39
Figure 3.5 : Bloc de régulation appelé dans un OB1.....	40
Figure 3.6 : création du bloc de donné.....	40
Figure 3.7 : La fenêtre de choix des paramètres de réglage.....	41
Figure 3.8 : La vue principale de l'interface graphique IHM.....	42
Figure 3.9 : La vue d'un projet WinCC.....	43
Figure 3.10 : La vue des Archives.....	43
Figure 3.11 : La table des variables sous le WinCC.....	43
Figure 3.12 : La fenêtre de choix de l'archive.....	44
Figure 3.13 : La fenêtre de choix de cycle d'archivage.....	44

Introduction générale

En vue de l'avancement des systèmes d'automatisations, et de la place qu'ils occupent dans le monde industriel, le département d'automatique a proposé de commander la station CE117 par l'automate programmable S7 314 IFM dans le but de former des élèves ingénieurs automaticiens à maîtriser des systèmes d'automatisation, et à appliquer les aspects théoriques sur des systèmes réels qui ressemblent à ceux qui sont disponibles dans l'industrie.

Ce travail comporte des parties expérimentales, qui permettent non seulement de voir concrètement l'aboutissement des résultats et la finalité de l'étude, mais aussi de faire ressortir les problèmes cruciaux de la mise en œuvre. Ce travail permet aussi aux étudiants de se familiariser avec les systèmes industriels, notamment les systèmes hydrauliques et les API, d'appliquer les connaissances acquises dans l'informatique industrielle, l'identification, et la commande grâce aux différents systèmes de régulation qui se trouvent sur la station CE117, tel que le système de régulation de niveau, de débit, de pression, et de température.

Nous exposons dans le présent rapport trois chapitres décrivant les volets principaux de notre projet :

Le premier chapitre sera consacré à la description et l'identification des différents systèmes de la station CE117.

Le deuxième chapitre sera partagé en deux parties, la première partie est une partie consacrée à la description de la commande par backstepping et la commande H_∞ , dans la deuxième partie on synthétisera des différentes commandes pour chaque système de la station

Le troisième chapitre sera partagé aussi en deux parties dans la première partie on va présenter la méthode d'implémentation des commandes sur le Step7. Dans la deuxième partie on va voir la supervision et l'archivage sous le WinCC.

Chapitre 1 :

Description et identification de la station CE117

PARTIE A : Description de la station TecQuipment Process Trainer CE117

1.A.1. Description de la station TecQuipment Process Trainer CE117 :

La station CE117 Process Trainer est un appareil éducatif destiné à l'étude et le contrôle de plusieurs systèmes : contrôle de niveau, pression, température et en fin de débit, qui sont des systèmes rencontrés très souvent dans le monde industriel. La station est composée de trois parties commandées par un module de contrôle, chacune regroupe plusieurs équipements nécessaires pour le contrôle des différentes grandeurs:

- 1- Le circuit fluide procédé : réservoir, pompe P2, vanne By-pass, capteur de température avant refroidissement, refroidisseur, capteur de température après refroidissement, vanne proportionnelle, capteur transmetteur de débit.
- 2- Le circuit fluide chaud : cuve de chauffe, capteur de température de fluide de la cuve de chauffe, pompe P1, capteur de débit, capteur de température à l'entrée de la cuve de chauffe.
- 3- La cuve agitée : vanne de mise à l'atmosphère, capteur transmetteur de niveau et de pression de l'air comprimé dans la cuve, agitateur, capteur transmetteur de température de fluide, le serpentin de chauffe de fluide, vanne de drainage.

Les équipements de cet appareil permettent de contrôler le débit, le niveau, la pression et la température non seulement séparément mais aussi en interaction et en cascade.

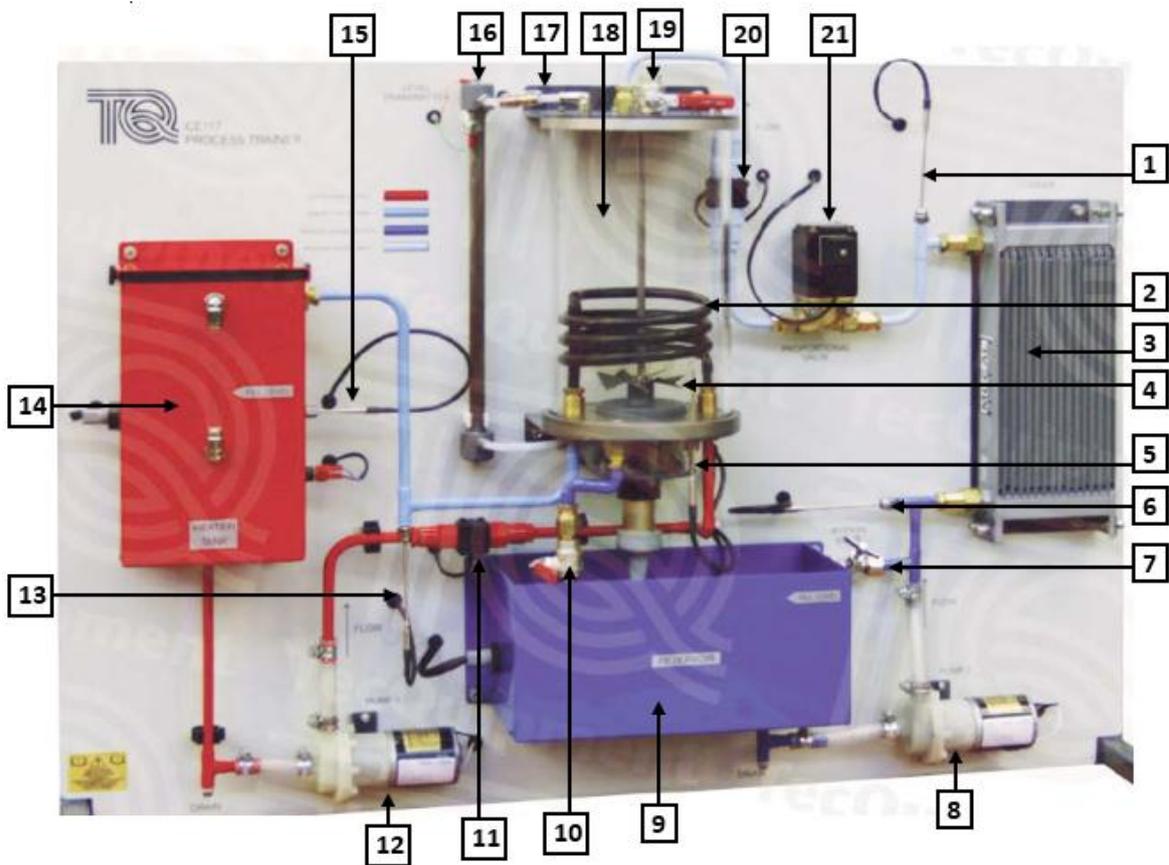


Figure1.1 : la station CE117

- | | |
|-------------------------------------|--------------------------------------|
| 1 - transmetteur de température TT4 | 12 - pompe 1 |
| 2 - serpentin de chauffe | 13 - transmetteur de température TT2 |
| 3 – refroidisseur | 14 - cuve de chauffe |
| 4 – Agitateur | 15 - transmetteur de température TT1 |
| 5 - transmetteur de température | 16 - transmetteur de niveau LT |
| 6 - transmetteur de température TT3 | 17 - transmetteur de pression PT |
| 7 - vanne By-pass | 18 – cuve agitée |
| 8 – pompe 2 | 19 – vanne vent |
| 9 - réservoir | 20 – transmetteur de débit FT2 |
| 10 – vanne de drainage | 21 – vanne proportionnelle |
| 11 – transmetteur de débit FT1 | |

1.A.2. Les contraintes des différents systèmes de la station CE117 :

- 1- La pompe possède une zone morte (si la tension de commande de la pompe descend au-dessous de 2V, elle ne délivre plus de débit).
- 2- Le capteur de niveau est non linéaire avec un gain variable.
- 3- La pompe ne peut délivrer que 5.2 (L/min) comme débit maximal.
- 4- La perturbation sur le débit provoqué par la vanne By-Pass peut atteindre 0.7 (L/min), ce qui va réduire le maximum de la référence de débit à 4.5 (L/min).
- 5- Si la pompe délivre un débit avec une pression inférieure à celle de l'air comprimé dans la cuve, le débit s'annule et la pression se stabilise à sa valeur malgré qu'il y ait une tension à l'entrée de la pompe.
- 6- L'automate programmable S7 314 IFM ne dispose que d'une seule sortie analogique, ce qui n'est pas suffisant pour la commande de notre système.

1.A.3. Les caractéristiques techniques des différents éléments de la station CE117 :

Le tableau suivant résume les caractéristiques techniques des différents éléments de la station CE117 :

article		Signal Analogique	Détail de la conversion	
PRT Température Transmetteurs (résistance platineum Thermomètre)	TT1	Sortie 0-10V linéaire	10°C par Volt 0V= 0°C 10V= 100°C	
	TT2			
	TT3			
	TT4			
	TT5			
Transmetteur de débit		FT1 FT2	Sortie 0-10V linéaire	1L/min par Volt 0V=pas de débit
Transmetteur de niveau		LT	Sortie 0-10V non linéaire	0V=réceptant vide 10V=Niveau Maximum
Transmetteur de pression		PT	Sortie 0-10V linéaire	100mbar par volt 0V=0mbar (jauge)
chauffage électrique			Sortie 0-10V	75W par volt 0V=chauffage OFF 10V=750W puissance max(Nominale)
Vanne Proportionnelle		S	Sortie 0-10V	0V=fermée 10V=ouverte
Pompe 1 Pompe 2			Sortie 0-10V	0V=pas de débit 10V=débit Maximal

Tableau 1.1 : les caractéristiques techniques des équipements de la station CE117

PARTIE B : Identification des différents systèmes de réglage commandés par la pompe P2 :

En utilisant la pompe P2 on peut régler le débit d'écoulement de l'eau dans la canalisation de circuit fluide procédé, le niveau d'eau dans la cuve agitée, et la pression de l'air comprimé dans la cuve.

1.B.1. Identification du système pompe - débit :

Pour l'identification de ce système, nous nous sommes basés sur l'étude de la réponse indicielle en boucle ouverte. On envoie un échelon à la pompe d'amplitude 80%, et à l'aide du logiciel CE2000, on relève la réponse qui est la tension image de débit selon la plage de conversion du capteur de débit.

En utilisant la méthode de broïda, on a assimilé la réponse de ce système à celui d'un premier ordre:

$$H(s) = \frac{0.6072}{2.9612s+1} \quad (1.1)$$

Donc le modèle d'état est le suivant :

$$\begin{cases} \dot{x} = -0.3377x + 0.205u \\ y = x \end{cases} \quad (1.2)$$

La figure ci-dessus représente les résultats de l'identification :

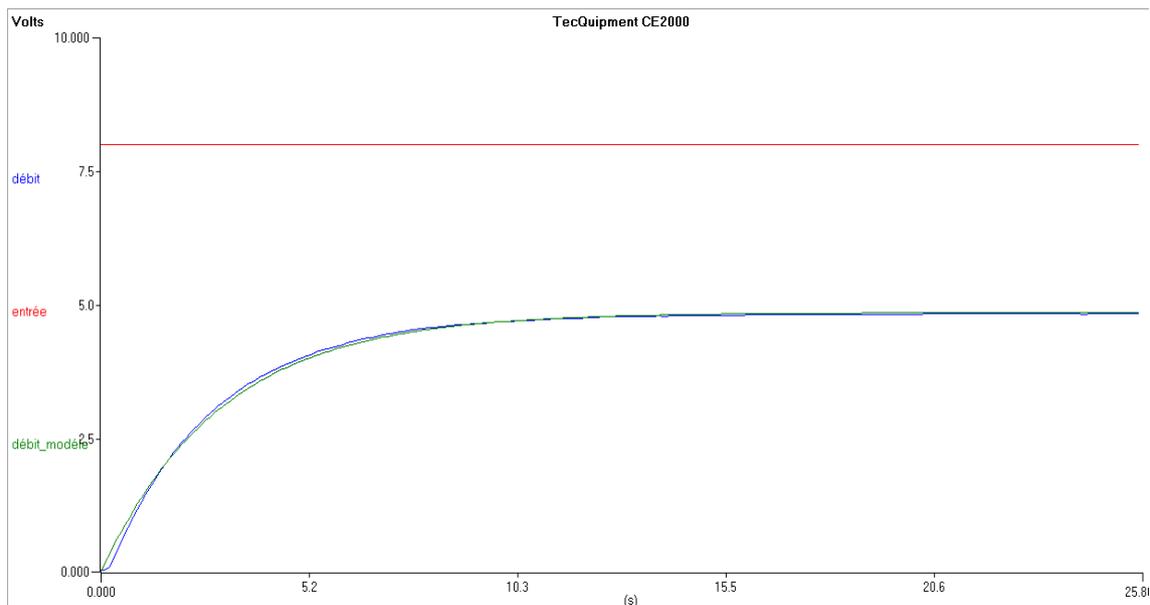


Figure 1.2: Réponse réelle et identifiée du système pompe-débit en boucle ouverte

1.B.2. Identification du système pompe-niveau :

Dans cette partie on garde la vanne proportionnelle entièrement ouverte et on régule le niveau par la pompe, cette dernière utilisée en mode analogique assure la variation de niveau de la cuve.

Pour l'identification de ce système, nous nous sommes basés sur l'étude de la réponse indicielle en boucle ouverte. On envoie un échelon à la pompe d'amplitude 80% et à l'aide du logiciel CE2000, on relève la réponse qui est la tension image du niveau.

En utilisant la méthode d'identification d'un système naturellement instable, on a assimilé la réponse de ce système à celui d'un intégrateur avec un retard pur:

$$H(s) = \frac{Ke^{-Ts}}{s} \quad (1.3)$$

Après avoir calculé les différentes constantes du modèle, par identification on trouve la fonction de transfert du système pompe-niveau suivante:

$$H(s) = \frac{0.0626e^{-0.64s}}{s} \quad (1.4)$$

Pour l'obtention du modèle d'état du système on a fait deux approximations :

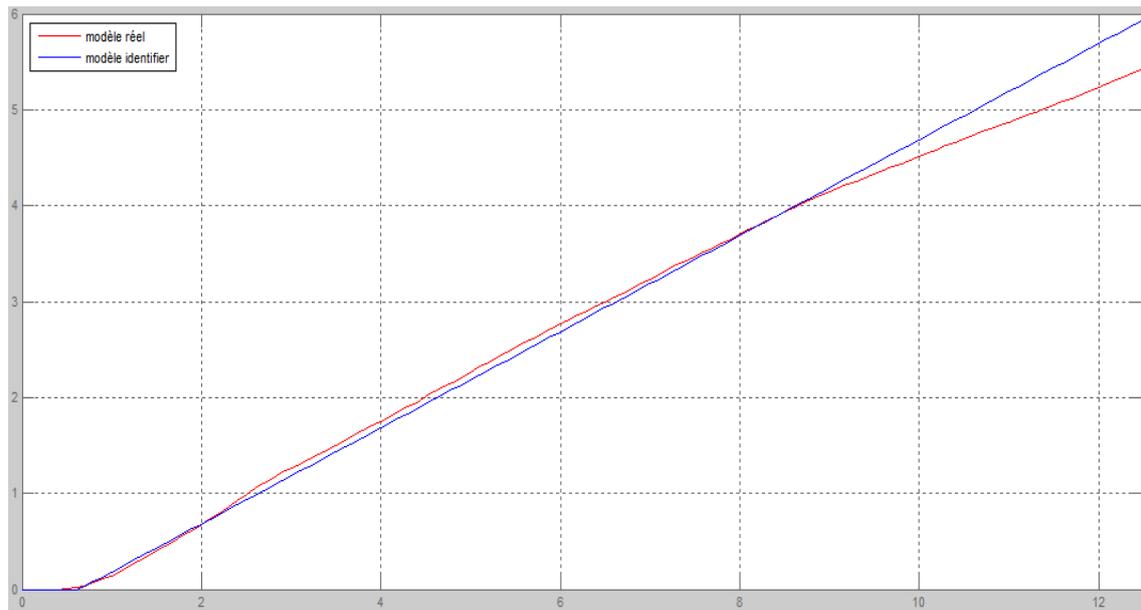
1. En négligeant le retard pur, on obtient le modèle suivant :

$$\begin{cases} \dot{x} = 0.0626u \\ y = x \end{cases} \quad (1.5)$$

2. En approximant le retard pur par un premier ordre, on obtient le modèle suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -1.5625x_2 + 0.0978u \\ y = x_1 \end{cases} \quad (1.6)$$

Afin de valider le modèle trouvé on importe les résultats (réponse réel) ainsi le temps échantillonné sous forme un fichier « Bloc note » à partir de logicielle CE2000 et on simule la réponse des deux modèles réel et identifier sur la même figure.



Figure

1.3 : Réponse réelle et identifiée du système pompe-niveau en boucle ouverte

1.B.3. Identification du système pompe-pression:

L'objectif derrière le réglage de la pression de l'air comprimé dans la cuve, est le réglage de la pression de fluide qui sort à travers la vanne de drainage comme montre la figure suivante.

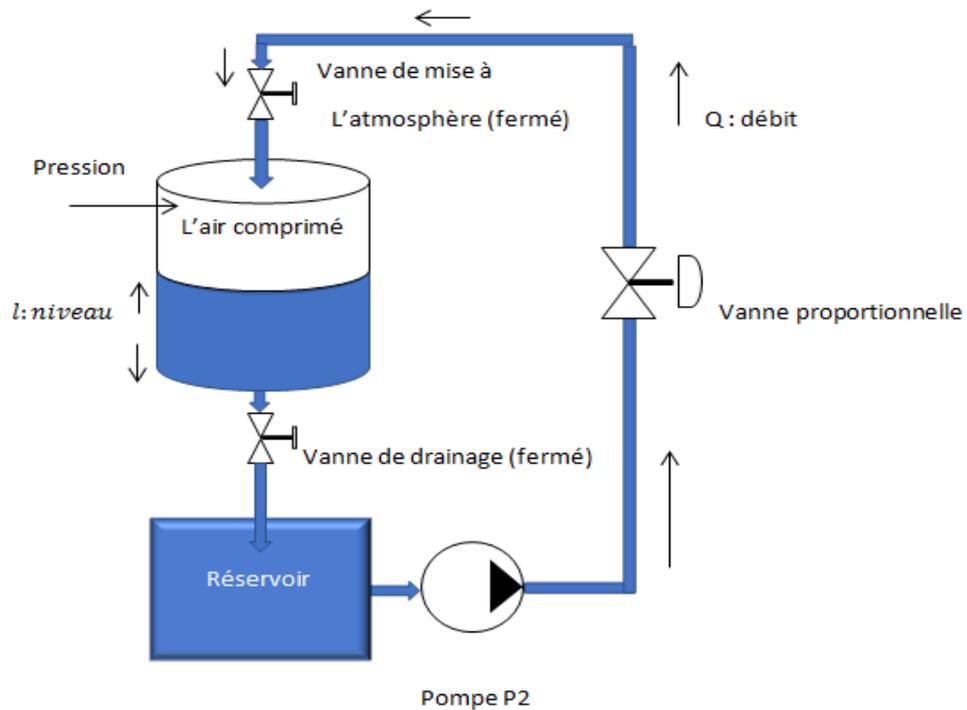


Figure1.4 : Schéma qui présente le circuit fluide procédé pour le réglage de la pression

Pour identifier ce système, on garde la vanne proportionnelle entièrement ouverte et on remplit la cuve jusqu'à 5 cm puis on ferme la vanne de la mise à l'atmosphère et on régule la pression par la pompe.

Pour l'identification de ce système, nous nous sommes basés sur l'étude de la réponse indicielle en boucle ouverte. On envoie un échelon à la pompe d'amplitude 8V et à l'aide du logiciel CE2000, on relève la réponse qui est la tension image de la pression.

En utilisant la méthode de broïda, on a assimilé la réponse de ce système à celui d'un premier ordre :

$$H(s) = \frac{0.6125}{12s+1} \quad (1.7)$$

Le modèle d'état de ce système est donné par :

$$\begin{cases} \dot{x} = -0.0833x + 5.1041u \\ y = x \end{cases} \quad (1.8)$$

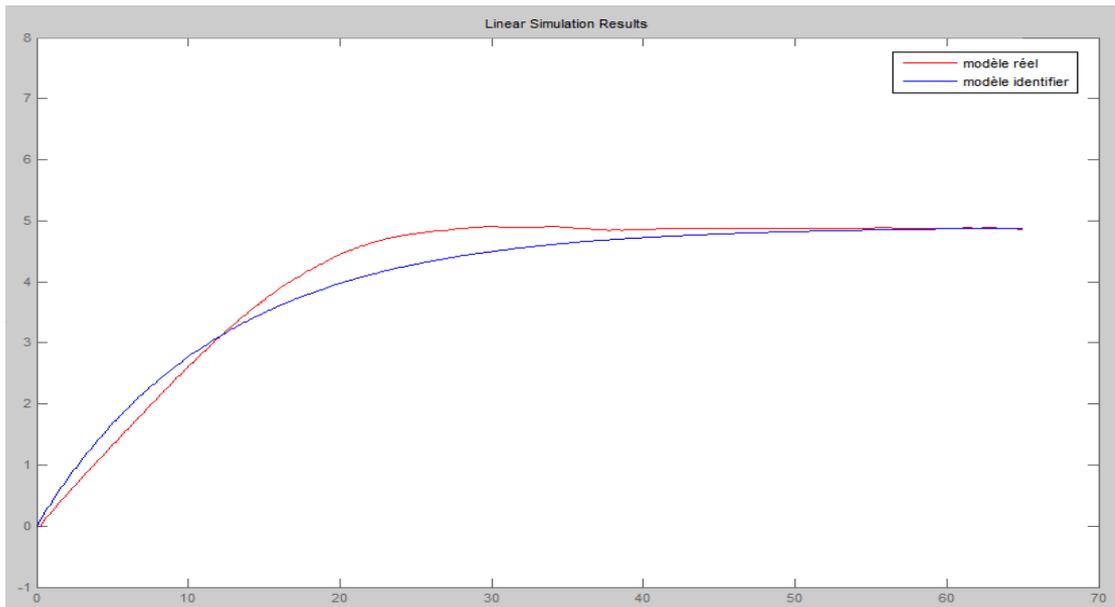


Figure 1.5 : Réponse réelle et identifiée du système pompe-pression en boucle ouverte

Chapitre 2 :

**Description et synthèse de la
commande par backstepping et H_∞**

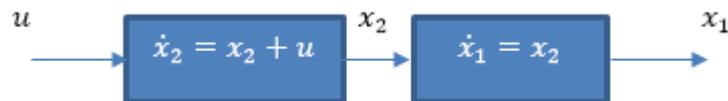
Introduction :

Ce chapitre est partagé en deux parties : Dans la première partie, on va voir la description des commandes utilisées pour le contrôle des différents systèmes de la station en utilisant la pompe P2 comme actionneur, puis dans la deuxième partie on va synthétiser la commande par backstepping pour les différents systèmes et la commande H_∞ pour le système pompe-niveau.

PARTIE A: Description des différentes commandes utilisées pour le contrôle de la station :

2.A.1. La commande de Backstepping :

La méthodologie du *Backstepping* peut se définir comme une façon d'organiser un système en plusieurs sous-systèmes en cascade. L'exploitation de la méthodologie de conception sur un plan général, aboutit à la mise en place d'une loi de commande par rétroaction associée systématiquement à une fonction de Lyapunov.



2.A.1.1. Théorie de Lyapunov :

On annonce dans cette partie la deuxième méthode de Lyapunov qui est venue pour remédier aux inconvénients de la première méthode, cette méthode est basée sur la recherche d'une fonction appelé fonction de Lyapunov, cette fonction possède une relation avec la norme de la variable d'état du système qui permet de décider sur la stabilité du système globale et non seulement locale.

En appliquant la théorie de Lyapunov (2^{ème} méthode) on peut déduire sur toutes les types de stabilité : stabilité exponentielle - stabilité asymptotique - stabilité au sens de Lyapunov- instabilité.

Stabilité asymptotique est le type de stabilité que nous tâcherons d'obtenir dans notre conception de commande.

2.A.1.2. Stabilité asymptotique :

S'il est possible de trouver une fonction $V(x)$ de signe définie positive «FDP» (avec $V(0) = 0$) dans un domaine ID comprenant la position d'équilibre, et dont la dérivée totale par rapport au temps soit définie négative «FDN» sur le même domaine ID, le point d'équilibre est dite asymptotiquement stable dans ce domaine.

Pour éviter les problèmes et faciliter la recherche de la fonction de Lyapunov, on va utiliser la méthodologie de Backstepping qui permet de synthétiser une commande bien définie et qui assure la stabilité asymptotique.

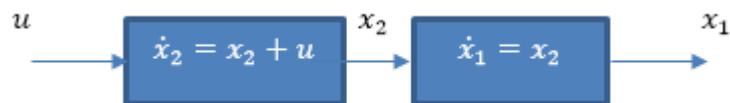
2.A.1.3. Exemple d'application :

Soit le système suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_2 + u \end{cases}$$

Pour ce système, on veut que notre variable d'état x converge vers 0

La synthèse de la commande de ce système s'effectue par deux étapes :



Etape 01: commandé x_1 par la commande fictif x_2

$$v_1 = \frac{1}{2}x_1^2 \quad (\text{FDP sur } \mathbb{R})$$

$$\dot{v}_1 = x_1 \dot{x}_1 = -\lambda_1 x_1^2$$

$$x_2 = -\lambda_1 x_1 = \varphi_1 \quad \lambda_1 > 0$$

Etape 02: commandé x_2 par la commande u

$$v_2 = \frac{1}{2}x_1^2 + \frac{1}{2}(x_2 - \varphi_1)^2 \quad (\text{FDP sur } \mathbb{R}^2)$$

$$\dot{v}_2 = x_1 \dot{x}_1 + (x_2 - \varphi_1)(\dot{x}_2 - \dot{\varphi}_1) = x_1 \varphi_1 + (x_2 - \varphi_1)(x_1 + \dot{x}_2 - \dot{\varphi}_1)$$

Pour que la dérivé \dot{v}_2 soit FDN, le terme multiplie fois x_2 doit être égale $-\lambda_2 x_2$ tel que $\lambda_2 > 0$

$$x_1 + x_2 + u - \dot{\varphi}_1 = -\lambda_2(x_2 - \varphi_1)$$

Et pour atteindre ce résultat, la commande u doit être égale :

$$u = -\lambda_2(x_2 - \varphi_1) - x_1 - x_2 + \dot{\varphi}_1$$

Cette commande qu'on a obtenue assure la convergence de la variable d'état ainsi la stabilité du système.

2.A.2. La commande H_∞ :

La commande H_∞ fait partie des méthodes de synthèse fréquentielles, La commande apporte une solution au problème de contrôle avec un certain nombre de contraintes. Elle permet de prendre en compte des spécifications données par le cahier de charge qui peut contenir trois classes de spécifications :

- La poursuite de référence (consigne).
- Rejet / atténuation des signaux de perturbations.
- Élimination des effets des erreurs de modélisation sur la stabilité du système.

2.A.2.1. Théorème du petit gain (Small gain):

Le théorème du petit gain présente un résultat extrêmement important pour étudier la stabilité en boucle fermée, il se base sur la notion de norme des systèmes. C'est une méthode légère à mettre en œuvre, elle donne une condition suffisante facile à tester qui permet de conclure sur la stabilité.

Théorème :

Soient $G_1(s)$, $G_2(s)$ deux fonctions de transferts stables, le système en boucle fermée est stable si : $\|G_2G_1\|_\infty < 1$

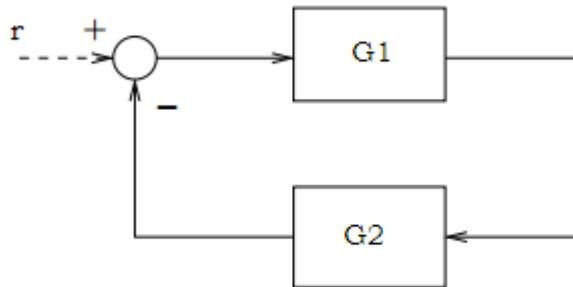


Figure 2.1 : Schéma d'un système de commande bouclé.

La puissance de ce théorème réside dans le fait qu'il simplifie l'évaluation de la robustesse des systèmes linéaires en cas de présence des erreurs de modélisation dont le comportement fréquentiel peut être majoré par une certaine bande fréquentielle $B(\omega)$.

2.A.2.2. Conditions de la stabilité robuste :

L'approche classique de la synthèse par marge de gain et marge de phase permet d'avoir une idée sur le degré de robustesse d'un système mais elle n'inclus pas sous une manière analytique les erreurs de modélisation. Les méthodes de synthèse robustes apportent une information plus large sur la réaction de la régulation avec les erreurs de modélisation. L'exploitation du théorème du petit gain permet de sortir par une condition suffisante pour la stabilité robuste.

2.A.2.3. Formulation du problème standard H_∞ :

L'approche utilisée dans le problème H_∞ consiste à une balance optimale selon un certain critère entre les performances et la robustesse. En premier temps le problème H_∞ est un problème de rejet de perturbation. Il consiste à minimiser l'effet d'une perturbation sur le comportement du système. Le signal v est supposé d'énergie finie. Enfin, on peut agir sur le système par une commande u et on dispose d'une observation e . Il s'agit donc de synthétiser une loi de commande $u = R(s).e$ qui minimise l'impact de v sur z . D'autre part, le problème H_∞ inclut aussi les performances en mettant dans le quantificateur de performance le signal qui représente l'erreur de poursuite et il s'agit aussi d'un problème de minimisation pondérée. Le problème H_∞ peut se schématiser comme suit :

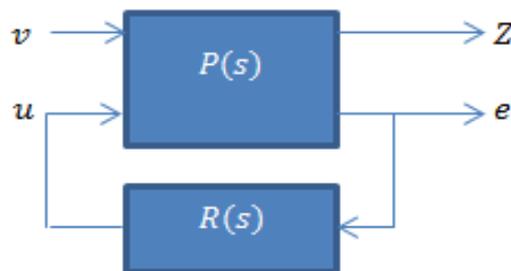


Figure 2.2 : La forme standard du problème H_∞ .

Le signal v représente les entrées extérieures, telles que les signaux de référence, les perturbations, les bruits... ; le vecteur u représente la commande, le vecteur quantificateur de performance z est choisi pour caractériser le bon fonctionnement de l'asservissement ; enfin e représente les mesures disponibles pour élaborer la commande (l'erreur de régulation).

La matrice de transfert $P(s)$ qui est supposé linéaire invariant dans le temps modélise les interactions dynamiques entre deux ensemble d'entrées et deux ensembles de sorties, tandis que $R(s)$ désigne le correcteur que l'on cherche à calculer. Le système admet l'écriture suivante :

$$\begin{pmatrix} Z \\ e \end{pmatrix} = \begin{pmatrix} P_{11} & P_{21} \\ P_{12} & P_{22} \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix}$$

Le transfert qu'on cherche à minimiser entre les entrées et les sorties commandées peut se mettre sous la forme suivante :

$$z = [P_{11} + P_{12}R(I - P_{22}R)^{-1}P_{21}] v = F_l(P, R) v$$

Cette expression, généralement notée $F_l(P, R)$, est appelée (Transformation Fractionnaire Linéaire) présente la relation de base pour la synthèse H_∞ car l'objectif de la commande est de minimiser la norme infinie ce qui revient à minimiser l'effet des signaux d'entrées sur le quantificateur de performance. En fait l'approche de synthèse utilise une démarche inverse, le principe c'est de poser la forme adéquate de $F_l(P, R)$ puis chercher la configuration associée à cette

forme. En générale la forme la plus utilisée est celle qui mettre ensemble la fonction sensibilité et la sensibilité complémentaire ce problème est appelé problème H_∞ de sensibilité mixte, donc le problème revient à minimiser le critère suivant :

$$R(s) = ? \quad \text{Min} \left\| \begin{matrix} w_1 S \\ w_2 T \end{matrix} \right\|_\infty$$

w_1 et w_2 sont des pondérations fréquentielles tel que w_1 pondère la fonction sensibilité S dans les basses fréquences et w_2 pondère la fonction sensibilité complémentaire T dans les hautes fréquences. Ce critère est une balance pour le compromis performance-robustesse, minimiser la sensibilité en basses fréquences revient à réduire l'erreur de poursuite, et minimiser la sensibilité complémentaire en hautes fréquences revient à réduire l'effet des erreurs de modélisation.

La résolution du problème fait appelle au problème "sous-optimal" qui stabilise le système et assure $\|F_l(P, K)\|_\infty < \gamma$, $\gamma > 0$. Les correcteurs satisfaisant ces deux conditions sont dites sous-optimum. Alors que le problème optimal assure le minimum de la norme de la transformation fractionnaire :

$$\text{Min}_{R(s)} \|F_l(P, K)\|_\infty$$

2.A.2.4. Synthèse de la commande

La commande H_∞ vise à minimiser le critère qui inclut les deux fonctions S et T , on donne le schéma suivant pour le choix des signaux v , u et Z , e :

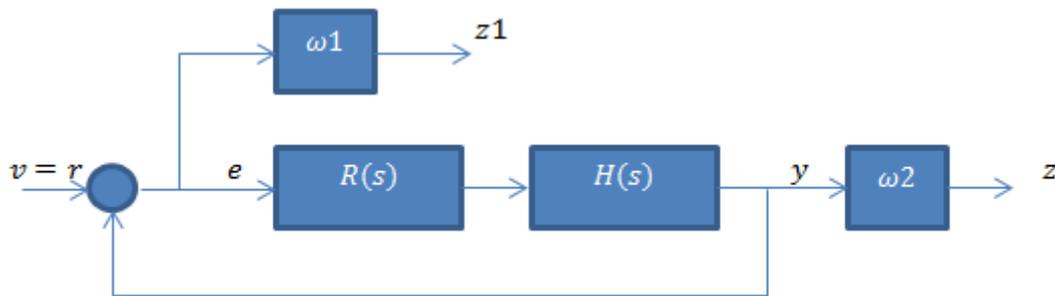


Figure 2.3 : choix des signaux pour le problème standard.

Avec la considération de $z_1 = w_1 S y e$, $z = w_2 T y y$, et $v=r$, on peut mettre le problème sous la forme standard ce qui donne la matrice $P(s)$:

$$P = \begin{pmatrix} w_1 & -w_1 G \\ 0 & w_2 G \\ I & -G \end{pmatrix}$$

Pour synthétiser la commande H_∞ il faut d'abord choisir les deux pondérations adéquates. En générale la première pondération w_1 est choisi de telle sorte qu'elle coupe à la bande passante désirée ce qui est l'équivalent d'un filtre passe bas, w_2 est choisie de telle sorte qu'il pondère la

fonction de sensibilité complémentaire dans les hautes fréquences ce qui a pour effet de sélectionner la plage de fréquence dans laquelle on minimise l'effet des erreurs de modélisation sur le système.

PARTIE B : Synthèse des commandes

2.B.1. Système Pompe-débit :

2.B.1.1. Synthèse de la commande de backstepping :

Pour synthétiser le régulateur par la méthode de backstepping, on va utiliser le modèle d'état suivant :

$$\begin{cases} \dot{x} = -0.3377x + 0.2050u \\ y = x \end{cases} \quad (2.1)$$

On prend $e = x - x_r$

L'objectif est que $e \rightarrow 0$

Soit la fonction de Lyapunov : $v = \frac{1}{2}e^2$ FDP sur IR

$$v = \frac{1}{2}e^2 \Rightarrow \dot{v} = e\dot{e} = (x - x_r)(\dot{x} - \dot{x}_r) = (x - x_r)(-0.3377x + 0.2050u)$$

$$\dot{v} \text{ Est FDN sur IR si } -0.3377x + 0.2050u = -\lambda(x - x_r) \quad (2.2)$$

(2.2) \Rightarrow

$$u = \frac{1}{0.2050} (0.3377x - \lambda(x - x_r)) \quad (2.3)$$

La figure suivante montre la réponse du système sous le WinCC :

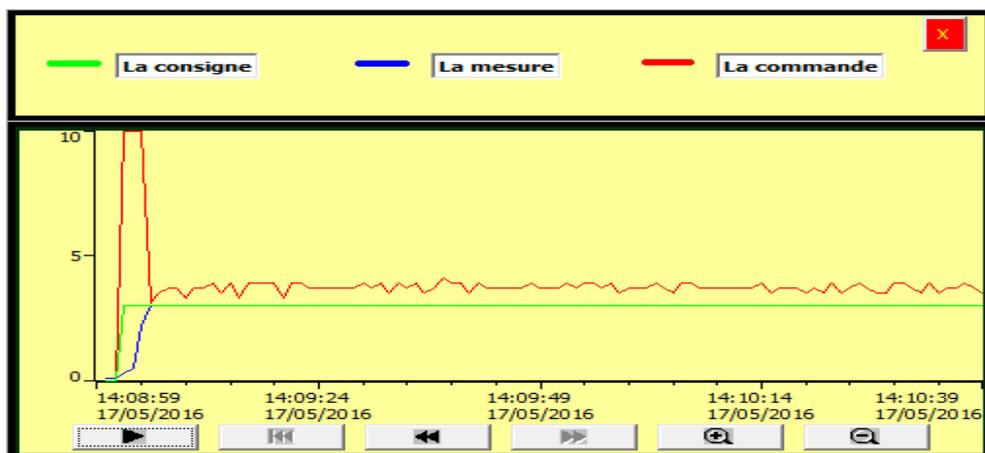


Figure 2.4 : La réponse du système pompe-débit sous le WinCC

2.B.1.2. Test de robustesse et de performances :

1- La poursuite et le rejet de perturbation :

Pour tester le rejet de perturbation, on va introduire une perturbation sur le débit par la vanne Bypass. D'après la figure ci-dessous on remarque bien qu'il y a une bonne poursuite de la référence et un bon rejet de perturbation, donc les performances sont assurées par la commande de backstepping.

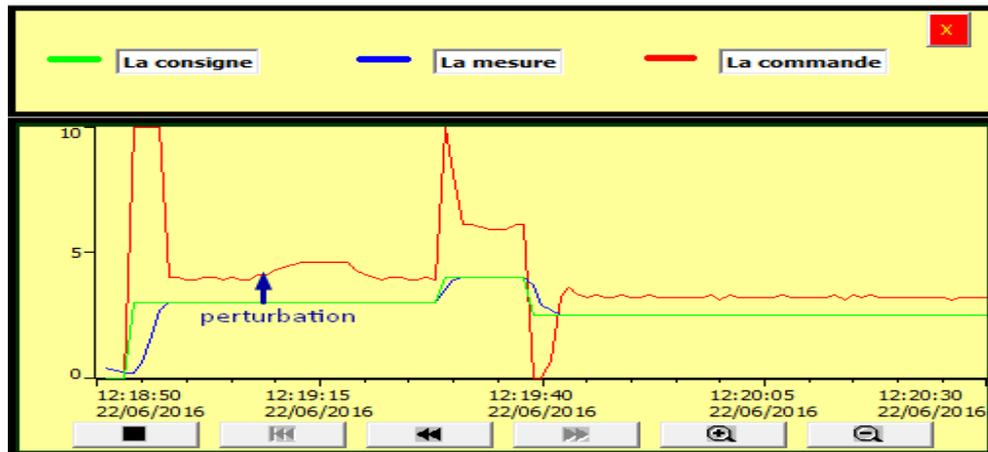


Figure 2.5 : La réponse du système pompe-débit avec perturbation sous le WinCC

2- La robustesse :

Pour tester la robustesse, on va changer l'ouverture de la vanne proportionnelle de 100% à 50%. D'après la figure 2.6 on remarque que malgré les variations paramétriques le système reste robuste.

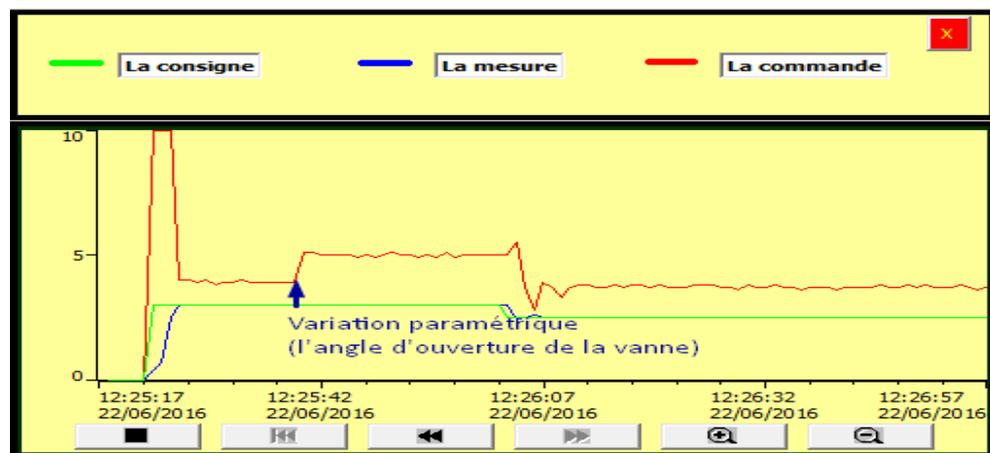


Figure 2.6 : La réponse du système pompe-débit avec variations paramétrique sous le WinCC

2.B.2. Système Pompe-niveau:

- **problème de non linéarité du capteur de niveau :**

Tension (V)	Niveau (Cm)	Gain Kc
1	1	1
2	2	1
3	3	1
4	4	1
5	5.2	0.96
6	6.6	0.90
7	8.4	0.83
8	10	0.80
9	12.2	0.73
10	14.8	0.67

Tableau 2.1 : la caractéristique statique et le gain du capteur de niveau

- **Résolution du problème :**

On va faire une adaptation par logique floue du gain de capteur en suivant ces étapes :

1- Fuzzification:



y : variable réelle « le niveau du capteur (V) »

y_F : La variable floue

Les valeurs floues de $y_F : \{TP, P, MP, M, MG, G, TG\}$, chaque valeur est associé d'une fonction d'appartenance

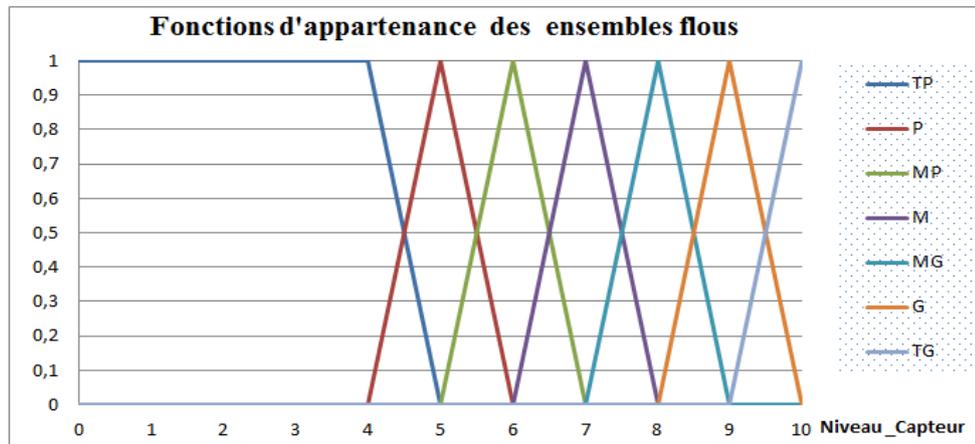


Figure 2.7 : Les fonctions d'appartenance de la variable floue

2- On a établi les règles sous l'approche de TSK comme suivant:

Règle 1 : si y_F est TP alors $K_1 = 1 \rightarrow \alpha_1 = \mu_{TP}(y_F)$

Règle 2 : si y_F est P alors $K_2 = 0.96 \rightarrow \alpha_2 = \mu_P(y_F)$

Règle 3 : si y_F est MP alors $K_3 = 0.9 \rightarrow \alpha_3 = \mu_{MP}(y_F)$

Règle 4 : si y_F est M alors $K_4 = 0.83 \rightarrow \alpha_4 = \mu_M(y_F)$

Règle 5 : si y_F est MG alors $K_5 = 0.8 \rightarrow \alpha_5 = \mu_{MG}(y_F)$

Règle 6 : si y_F est G alors $K_6 = 0.73 \rightarrow \alpha_6 = \mu_G(y_F)$

Règle 7 : si y_F est TG alors $K_7 = 0.67 \rightarrow \alpha_7 = \mu_{TG}(y_F)$

D'après l'approche de TSK, le gain réel du capteur est donné par l'expression suivante :

$$K = \frac{\sum \alpha_i K_i}{\sum \alpha_i}$$

Donc pour avoir le niveau réel de la cuve, on a divisé y par K

$$\text{Niveau-réel} = y / (\text{Niveau-capteur} / K) = y \cdot K / \text{Niveau-capteur}$$

Après la correction de gain du capteur le système se comporte exactement comme un intégrateur pur :

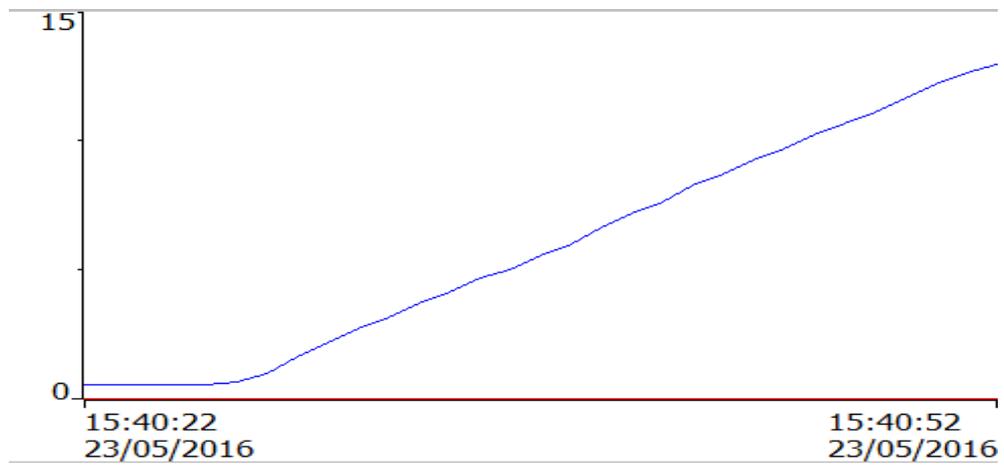


Figure2.8 : La réponse du système en boucle ouverte après la résolution de problème de non linéarité du capteur sous WinCC

Pour le réglage de ce système on va utiliser deux types de commandes :

- a- Le backstepping
- b- La commande H_∞

2.B.2.1. Commande par backstepping :

Pour synthétiser le régulateur par la méthode de backstepping, on va utiliser le modèle d'état suivant dans lequel on a négligé le retard pur :

$$\begin{cases} \dot{x} = 0.0626u \\ y = x \end{cases} \quad (2.4)$$

On prend $e = x - x_r$. L'objectif est que $e \rightarrow 0$

Soit la fonction de Lyapunov : $v = \frac{1}{2}e^2$ FDP sur IR

$$v = \frac{1}{2}e^2 \Rightarrow \dot{v} = e\dot{e} = (x - x_r)(\dot{x} - \dot{x}_r) = (x - x_r)(0.0626u)$$

\dot{v} Est FDN sur IR si $0.0626u = -\lambda(x - x_r)$ (2.5)

$$u = \frac{1}{0.0626}(-\lambda(x - x_r))$$

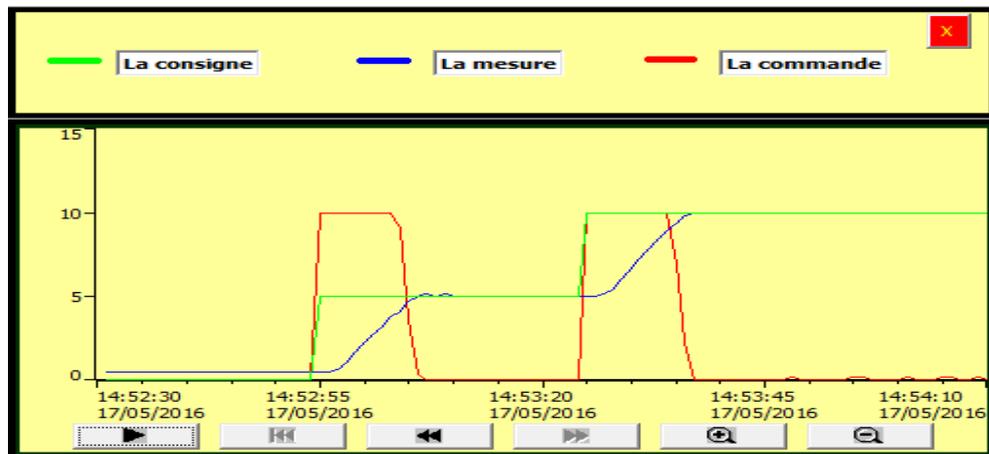


Figure 2.9: La réponse du système pompe-niveau sous le WinCC

2.B.2.2. Test de robustesse et de performances :

1- La poursuite et le rejet de perturbation :

Pour tester le rejet de perturbation de ce système, on va introduire une perturbation sur le niveau par la vanne de drainage. D'après la figure 2.10 on remarque bien qu'il y a une bonne poursuite de la référence et un bon rejet de perturbation, donc les performances sont assurées par la commande de backstepping.

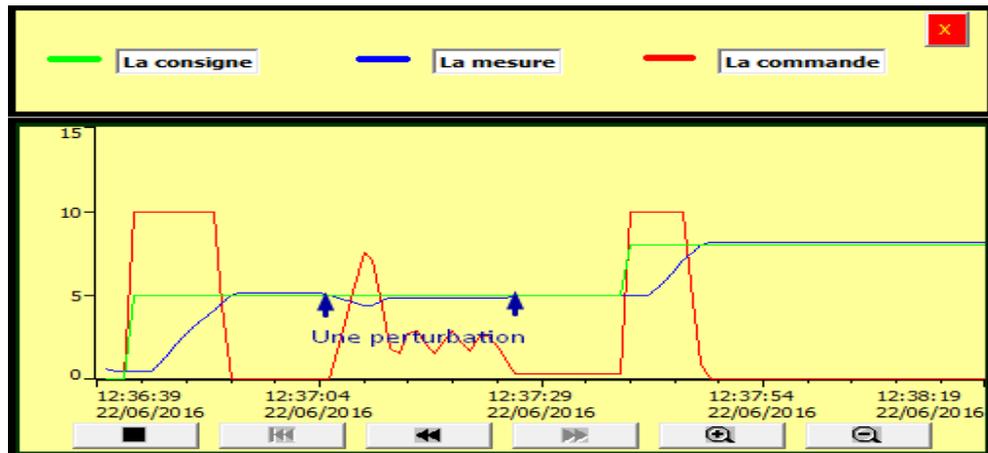


Figure 2.10: La réponse du système pompe-niveau avec perturbation sous le WinCC

2- La robustesse :

Pour tester la robustesse, on va changer l'ouverture de la vanne proportionnelle de 100% à 50%. D'après la figure 2.11 on remarque que malgré les variations paramétriques le système reste robuste.

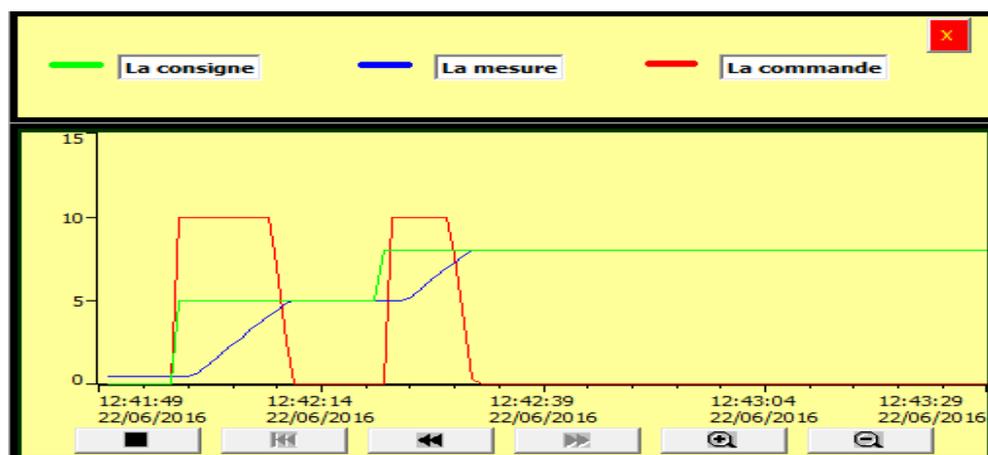


Figure 2.11 : La réponse du système pompe-niveau avec variations paramétrique sous le WinCC

2.B.2.3. La commande H_∞ :

La fonction de transfert en boucle ouverte du système pompe-niveau :

$$H(s) = \frac{0.0626}{(1 + 0.64s)s}$$

Le système en boucle fermé est donné par le schéma fonctionnel suivant :

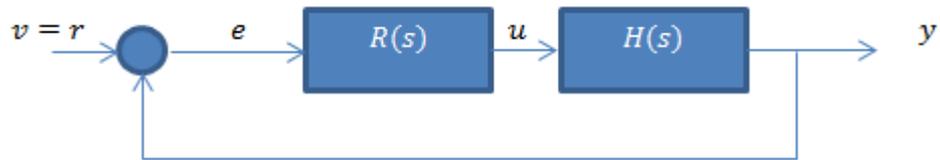


Figure 2.12 : le schéma fonctionnel du système en boucle fermé

1- Formulation du problème standard H_∞ :

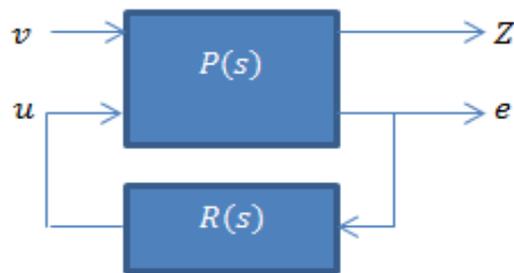


Figure 2.13 : La forme standard du problème H_∞

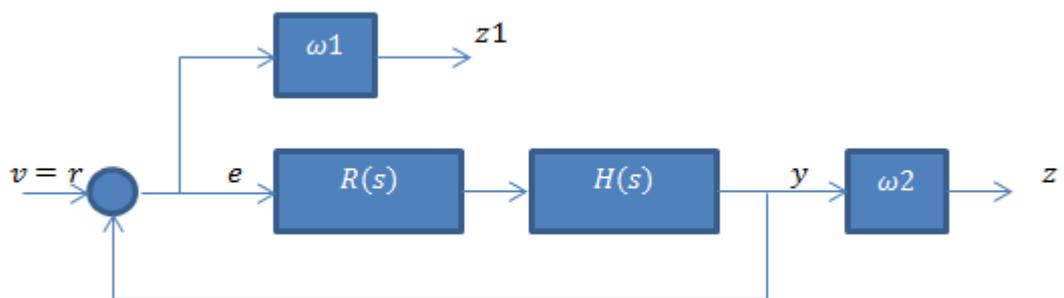


Figure 2.14: La mise sous la forme standard du problème H_∞

2- Le choix des fonctions de pondérations :

2.1. Fonction de pondération des performances :

$$\omega_1 = \frac{100}{(1 + 1000s)}$$

2.2. Fonction de pondération de robustesse :

$$\omega_2 = \frac{10s}{(1 + 0.1s)}$$

En utilisant la fonction « mixsyn » sur Matlab sous la syntaxe suivante :

$$R = \text{tf}(\text{mixsyn}(H, \omega_1, [], \omega_2))$$

On trouve le régulateur suivant:

$$R(s) = \frac{3.577 \times 10^4 s^3 + 4.136 \times 10^5 s^2 + 5.589 \times 10^5 s + 3.71 \times 10^{-5}}{(s^4 + 176.4 s^3 + 1.485 \times 10^4 s^2 + 3.506 \times 10^5 s + 348.9)} \quad (2.6)$$

Afin d'implémenter ce régulateur sur Step7 on va réduire l'ordre de régulateur, En utilisant la fonction « reduce » sur Matlab : $R = \text{zpk}(\text{reduce}(t, 3))$

$$R(s) = 35771 \frac{(s + 1.561)(s + 10)}{(s + 35.64)(s^2 + 140.7s + 9839)} \quad (2.7)$$

En appliquant le régulateur suivant sur le système sous matlab, on trouve la réponse suivante.

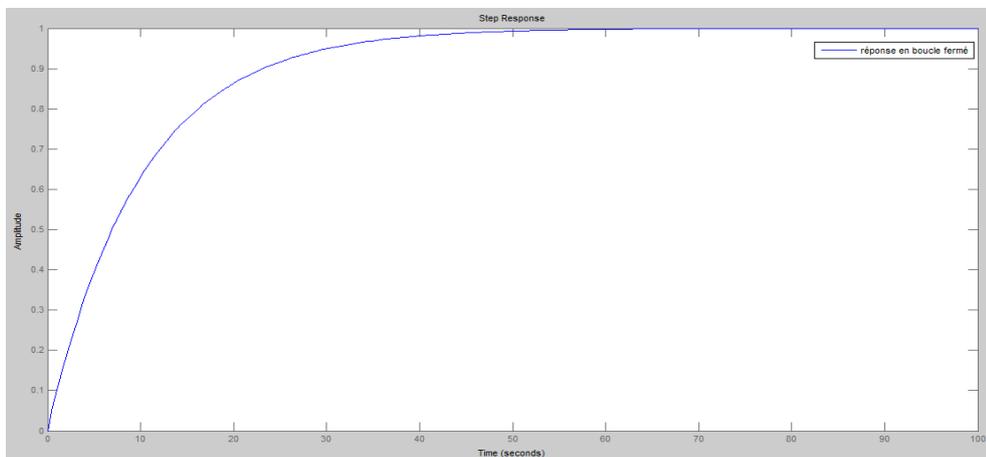


Figure 2.15: La réponse du système pompe-niveau sous matlab

Pour pouvoir appliqué ce régulateur en utilisant le Step7 on va approximer l'opérateur de dérivation dans l'espace de Laplace par une dérivé numérique.

$$s = \frac{1 - z^{-1}}{h}$$

h : Le temps d'échantionnage=100 ms.

z^{-1} : L'opérateur de retard dans le domaine temporel.

La réponse du système sous le WinCC :

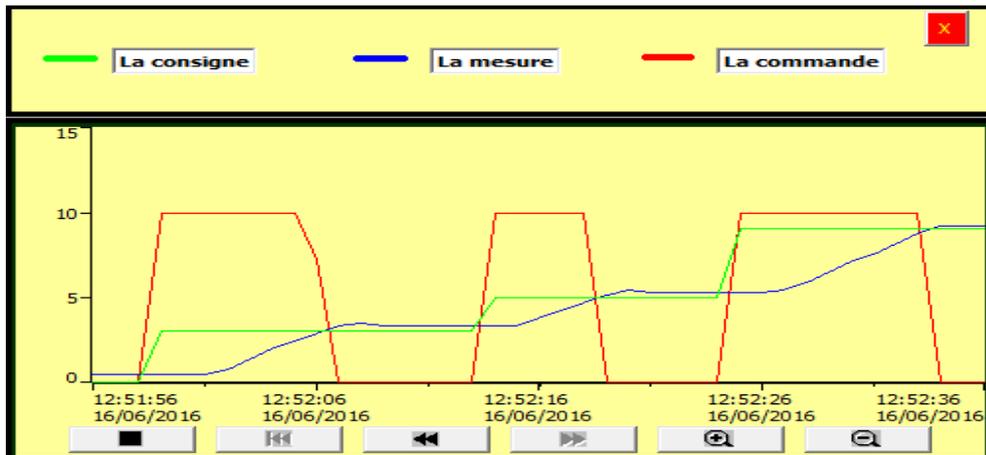


Figure 2.16: La réponse du système pompe-niveau sous WinCC

2.B.2.4. Test des performances et de robustesse :

Malgré les variations paramétriques et la perturbation le système reste stable et assure une bonne poursuite.

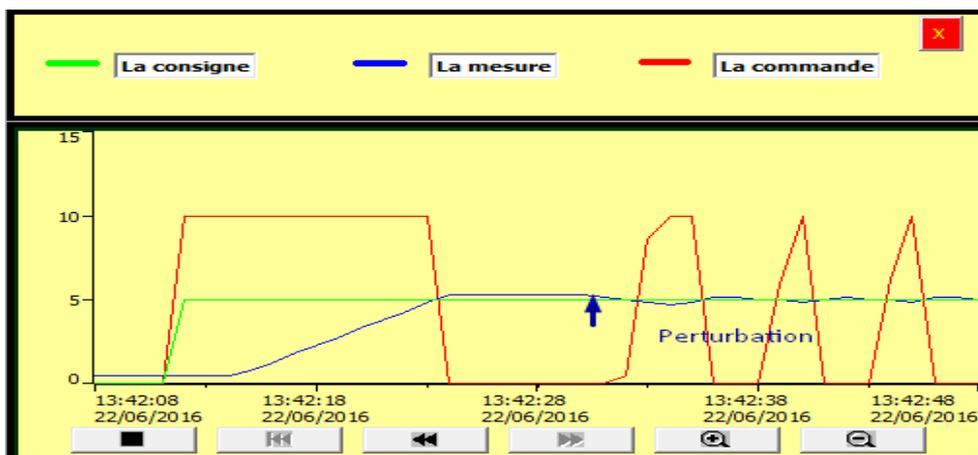


Figure 2.17: La réponse du système pompe-niveau avec perturbation et variations paramétriques sous WinCC

2.B.3. Système Pompe-pression:

2.B.3.1. Synthèse de la commande par backstepping :

Pour synthétiser le régulateur par la méthode de backstepping, on va utiliser le modèle d'état suivant :

$$\begin{aligned}\dot{x} &= -0.0833x + 5.1041u \\ y &= x\end{aligned}\quad (2.8)$$

On prend $e = x - x_r$

L'objectif est que $e \rightarrow 0$

Soit la fonction de Lyapunov : $v = \frac{1}{2}e^2$ FDP sur IR

$$v = \frac{1}{2}e^2 \Rightarrow \dot{v} = e\dot{e} = (x - x_r)(\dot{x} - \dot{x}_r) = (x - x_r)(-0.0833x + 5.1041u)$$

$$\dot{v} \text{ Est FDN sur IR si } -0.0833x + 5.1041u = -\lambda(x - x_r) \quad (2.9)$$

(2.9) =>

$$u = \frac{1}{5.1041} (0.0833x - \lambda(x - x_r))$$

(2.10)

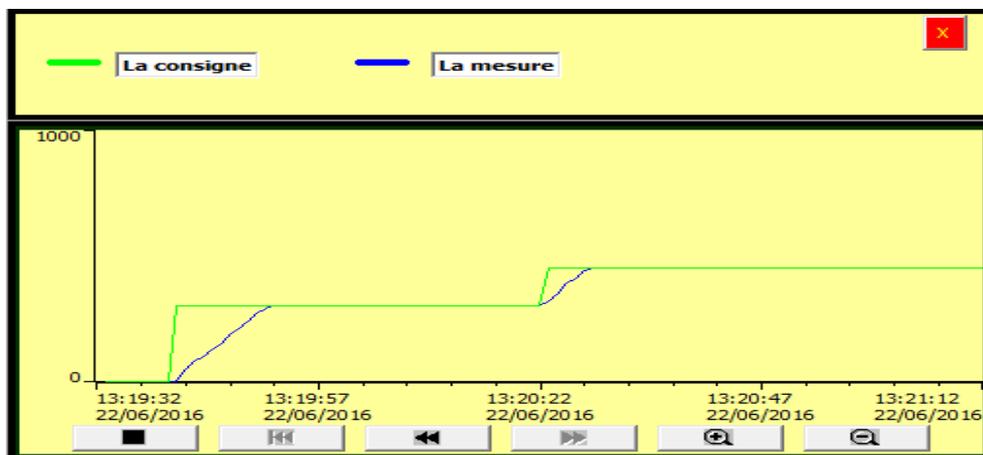


Figure 2.18: La réponse du système pompe-niveau sous WinCC

2.B.3.2. Test de robustesse et de performances :

- **Poursuite et rejet de perturbation :**

Pour tester le rejet de perturbation de ce système, on va introduire une perturbation sur la pression par la vanne de drainage. D'après la figure 2.19 on remarque bien qu'il y a une bonne poursuite de la référence et un bon rejet de perturbation, donc les performances sont assurées par la commande de backstepping.

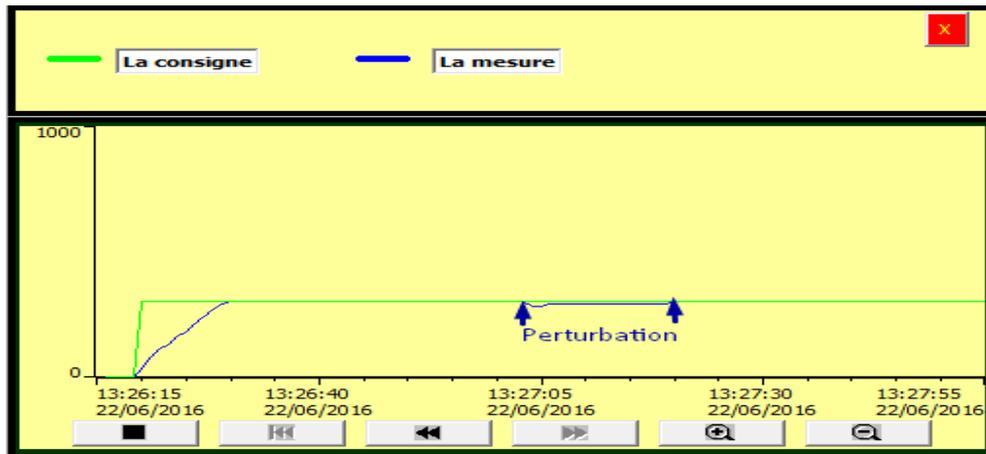


Figure 2.19: La réponse du système pompe-niveau sous WinCC

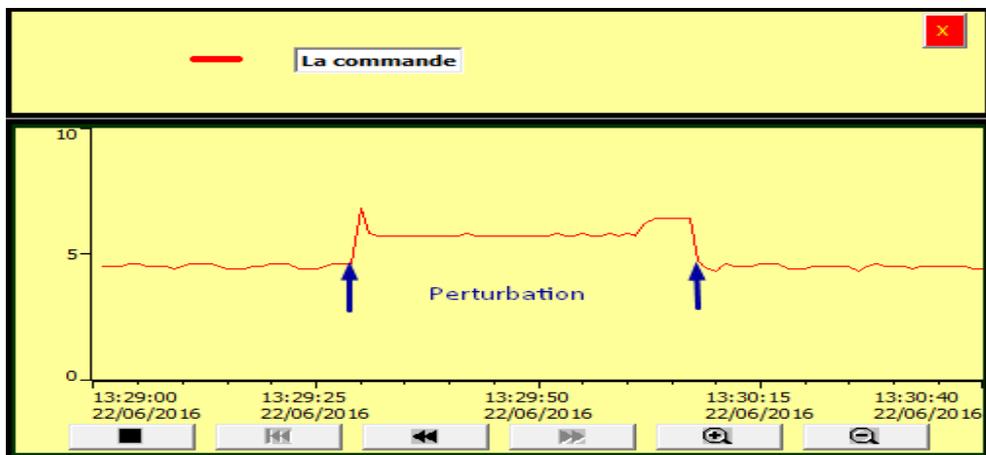


Figure 2.20: La commande appliquée au système pompe-niveau sous WinCC

Conclusion :

Dans ce chapitre, on a vu la synthèse de la commande par backstepping et la commande H_∞ pour les différents systèmes de la station, on a vu aussi le teste de robustesse et de performances de ces deux commande.

Chapitre 3:

**Implémentation des commandes sur
le Step7 et supervision par le WinCC**

PARTIE A : Implémentation des commandes sur le Step7

3.A.1. Implémentation des commandes sur le Step7 :

Pour pouvoir régler tous les systèmes identifiés, il faut d'abord lire les valeurs réelles des grandeurs à régler (Débit, Niveau, Pression) à travers les capteurs, puis après le calcul de la commande dans l'automate on envoie la commande directement vers l'actionneur à travers le module analogique.

Pour faire ceci il faut faire un adressage pour chaque entrée/sortie au niveau de la table des mnémonique et faire une mise à l'échelle des entrées et des sorties au niveau du programme principale OB1.

Adressage des entrées/sorties :

PEW	128	Débits
PEW	130	Niveau
PEW	132	Pression
PAW	128	L'entrée de l'actionneur

La mise à l'échelle :

Pour la mise à l'échelle des entrées qui viennent des capteurs et la sortie qui se dirige vers les actionneurs, on a utilisé les deux fonctions qui se trouvent dans la bibliothèque du Step7 : FC105 (SCALE) et FC106 (UNSCALE).

Pour l'implémentation des différentes commandes qu'on a synthétisées, on a créé des blocs fonctionnels à partir des fichiers sources en utilisant le langage structuré, et par la suite on a fait appel à ces blocs dans le programme principale OB1.

- Les étapes de l'implémentation des commandes:

1- Création d'un fichier source:

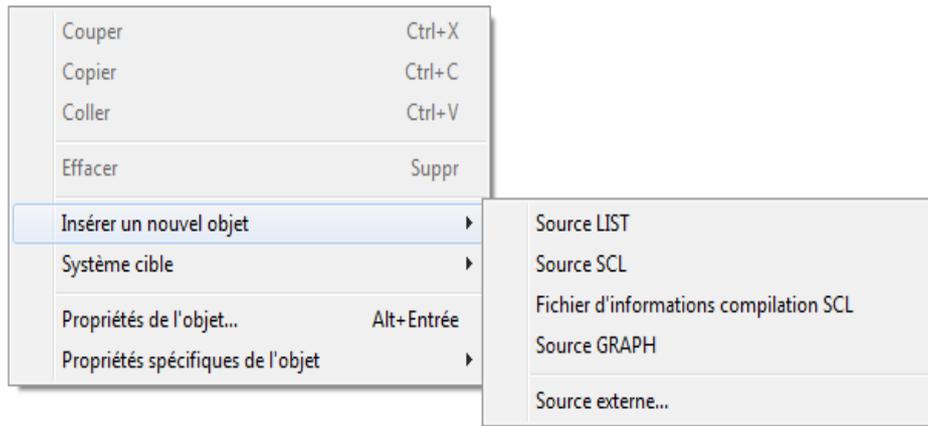


Figure 3.1: La création d'un fichier source

2- Le choix du modèle de bloc :

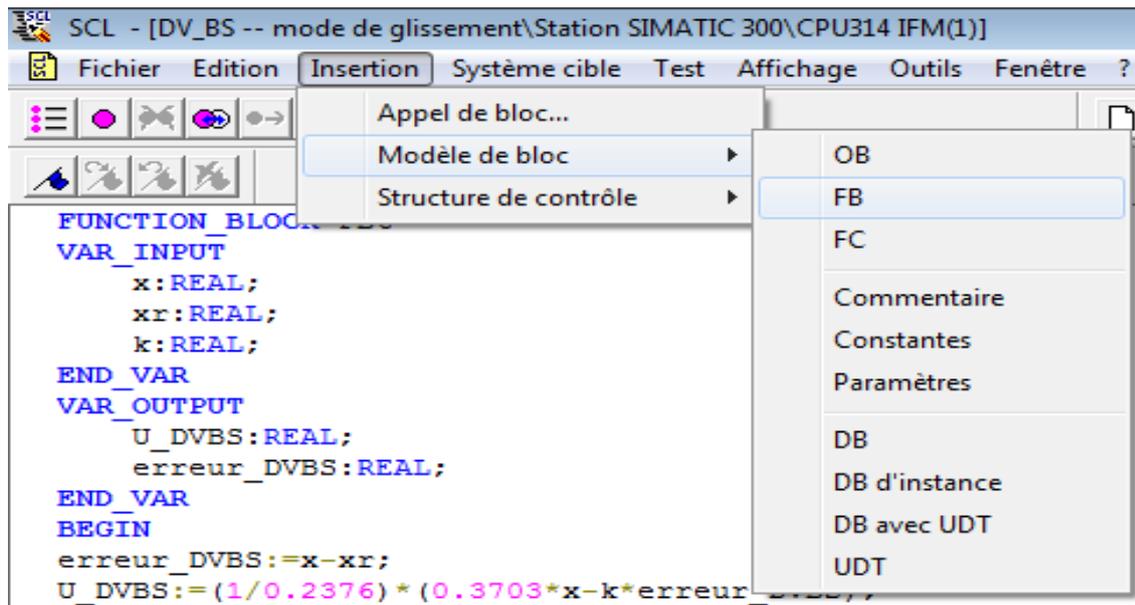


Figure 3.2: Le choix du modèle de bloc

3- La déclaration des variables et la saisie du programme:

Pour la programmation des blocs fonctionnels on a utilisé le langage structuré comme montre la figure:

```
FUNCTION_BLOCK FB3
VAR_INPUT
  x:REAL;
  xr:REAL;
  k:REAL;
END_VAR
VAR_OUTPUT
  U_DPBS:REAL;
  erreur_DPBS:REAL;
END_VAR
BEGIN
  erreur_DPBS:=x-xr;
  U_DPBS:=(1/0.2050)*(0.3377*x-k*erreur_DPBS);
END_FUNCTION_BLOCK
```

Figure 3.3: Programme sous le langage structuré dans un fichier source

4- La compilation du programme:

La compilation sert à détecter les erreurs, et à générer le bloc fonctionnel qui correspond au programme compilé.

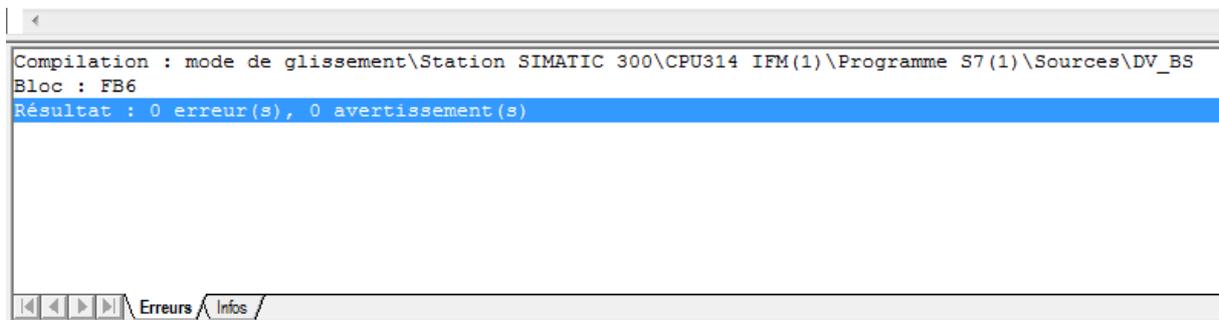


Figure 3.4: La fenêtre des erreurs de compilation

5- L'appel du bloc dans le programme principal OB1:

Pour appeler ce bloc dans le OB1 il suffit d'aller sur l'icône FB et de choisir le bloc fonctionnel et de le glisser à l'intérieure du réseau.

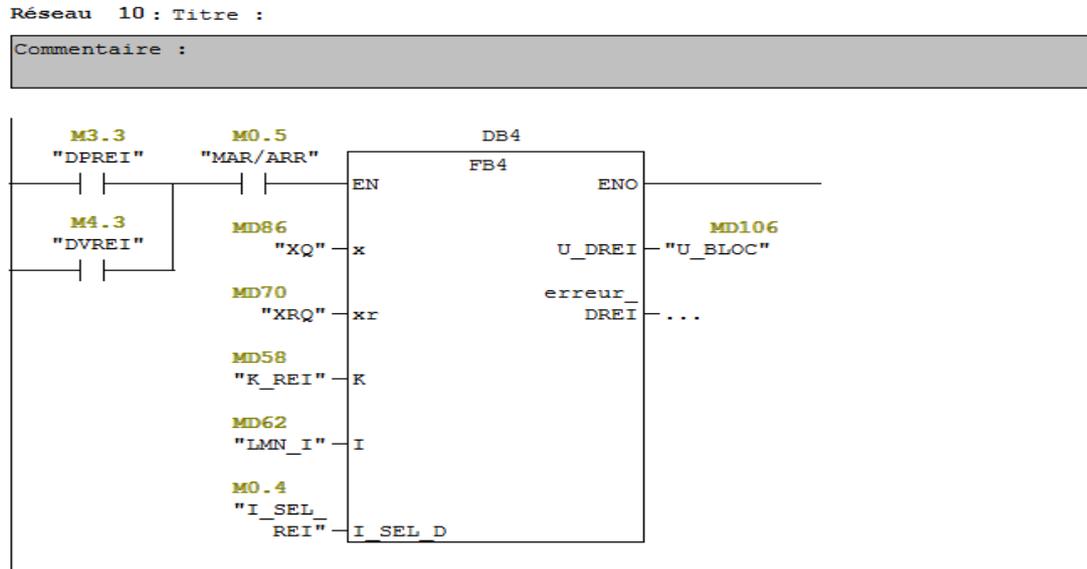


Figure 3.5: Bloc de régulation appelé dans un OB1

6- La création d'un bloc de donné qui correspond au bloc fonctionnel:

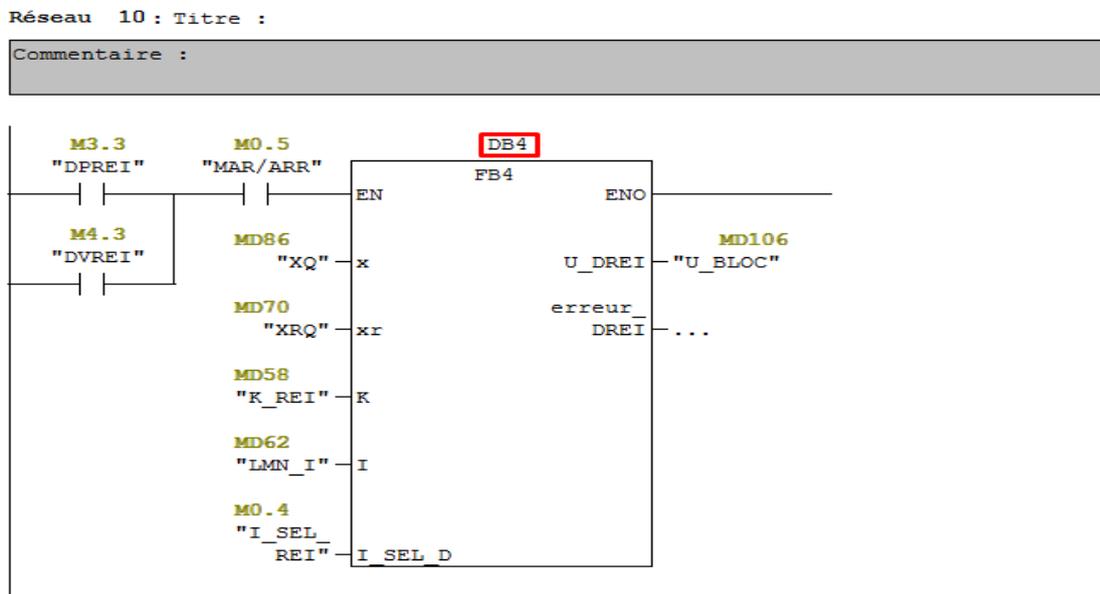


Figure 3.6: création du bloc de donné

Certains blocs qui contiennent une dérivation ou une intégration numérique (nécessitent un temps d'échantionnage précis) comme le FB41, on les appelle dans des OB d'alarme cyclique. La CPU S7 314 IFM met à notre disposition l'OB35 qui interrompt le traitement de programme cyclique à intervalles précis. Dans la configuration matérielle, il est possible de choisir la fréquence d'exécution de l'OB35 en allant dans les propriétés de la CPU sous l'onglet Alarmes cycliques. Ce temps ne doit pas être trop court. On doit s'assurer que le reste du programme et notamment l'OB1 ait le temps de s'exécuter entre deux appels de l'OB35.

Les alarmes cycliques sont déclenchées à des intervalles de temps précis. Le moment de déclenchement de la période est le passage de l'état de fonctionnement "Arrêt" (STOP) à l'état "Marche" (RUN). Le tableau suivant montre les périodes et classes de priorité prédéfinies des OB d'alarme cyclique. On peut modifier les périodes et classes de priorité par paramétrage.

PARIE B : Supervision et Archivage sous le WinCC

3.B.1. Supervision sous le WinCC

Pour une bonne visualisation et pour contrôler facilement les opérations d'automatisations, on a créé une interface graphique sur le WinCC.

Cette interface est composée de deux vue, la vue principale et la vue secondaire.

Dans la vue principale de l'interface graphique, on trouve :

- 1- un schéma représentatif de la station CE117
- 2- Une fenêtre partagée en 3 parties :
 - La première partie c'est pour l'état du système (Marche / Arrêt)
 - La deuxième partie c'est pour le choix des paramètres de réglage

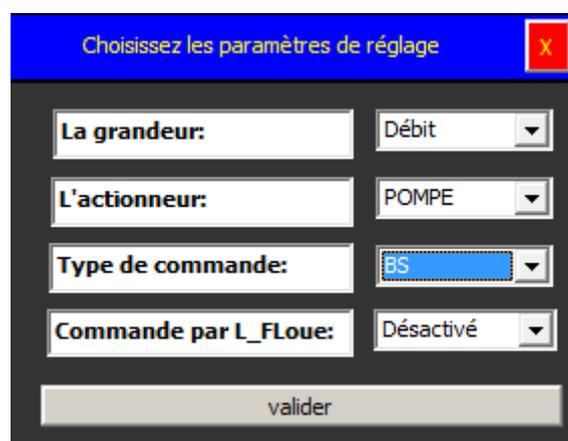


Figure 3.7 : La fenêtre de choix des paramètres de réglage

- La troisième partie est consacrée au contrôle de système

Dans la deuxième partie, on a les possibilités suivantes :

- Le choix de système de réglage : la grandeur, l'actionneur et le type de commande.
- La saisie des paramètres de commande
- La visualisation de la courbe de réglage
- 3- Un bouton qui permet le choix entre la commande automatique et manuelle
- 4- Une fenêtre d'alarme pour l'affichage des alarmes et des avertissements
- 5- Un tableau d'affichage de système de réglage

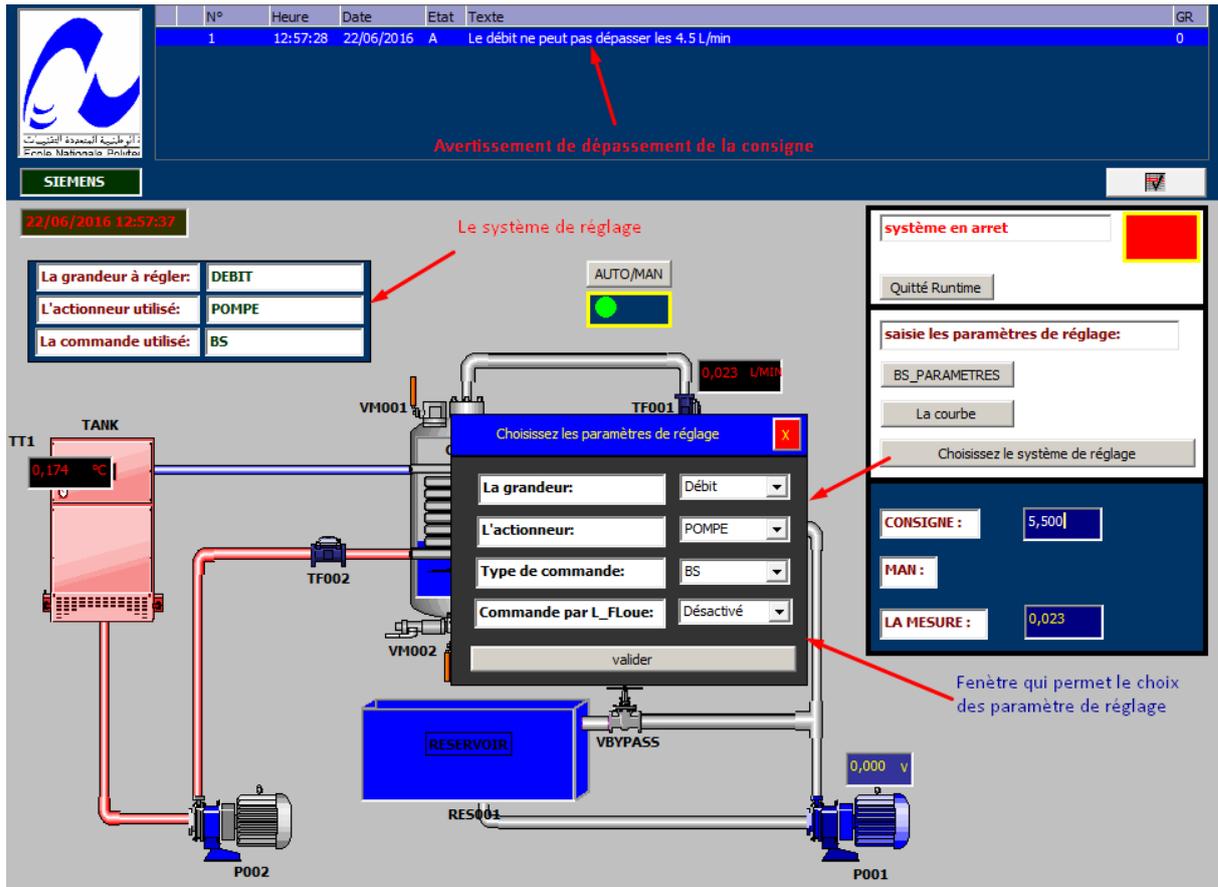


Figure 3.8: La vue principale de l'interface graphique IHM

Dans la vue secondaire de l'interface graphique, on trouve :

- La visualisation de la courbe de réglage
- Fenêtre de contrôle l'état de système (Marche / Arrêt)
- Fenêtre pour la saisie de la consigne et pour visualiser la mesure
- Fenêtre pour la saisie des paramètres de commande
- Une fenêtre d'alarme pour l'affichage des alarmes et des avertissements.

3.B.2. Archivage sous le WinCC :

Pour identifier n'importe quel système il faut d'abord avoir sa réponse sous forme une courbe si il s'agit des méthodes classiques ou bien sous forme un fichier de données pour le cas d'identification par les méthodes numériques. Sous le WinCC, cette étape est appelé l'archivage. L'archivage consiste à crée un fichier de données pour une variable donné, ce fichier peut être traité par d'autre programme telle que le logiciel matlab.

3.B.2.1. Les étapes d'archivage d'une variable :

- En allant sur Historique, choisissez Archives comme montre la figure ci-dessous.

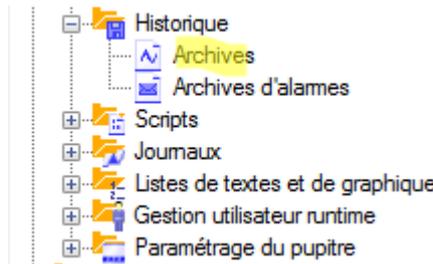


Figure 3.9: La vue d'un projet WinCC

- La fenêtre ci-dessous s'ouvre. Dans cette fenêtre, on a les possibilités suivantes :
 - 1- Le nom de l'archive.
 - 2- Le format de fichier de l'archive.
 - 3- L'emplacement, ou on veut enregistrer le fichier.

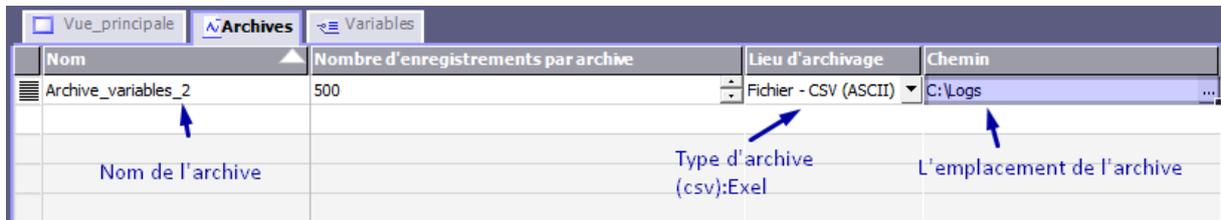


Figure 3.10: La vue des Archives

- Aller sur la table des variables, aller sur la colonne de la variable qu'on veut archiver.

Nom	Élément...	Cycle d'acqui...	Commentaire	Archive de données	Mode d'acquisition archivage	Cycle d'archivage
ER_VD	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
KI_REI	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
K_REI	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XN	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XRQ	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XRL	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XRP	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XRT	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XQ	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XL	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XP	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
XT	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
MW3	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
MW5	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>
MW7	1	100 ms		<indéfini>	Cyclique en continu	<indéfini>

Figure 3.11: La table des variables sous le WinCC

- En cliquant sur Archive de données, la fenêtre ci-dessous s'ouvre choisissez l'archive que vous avez créé.

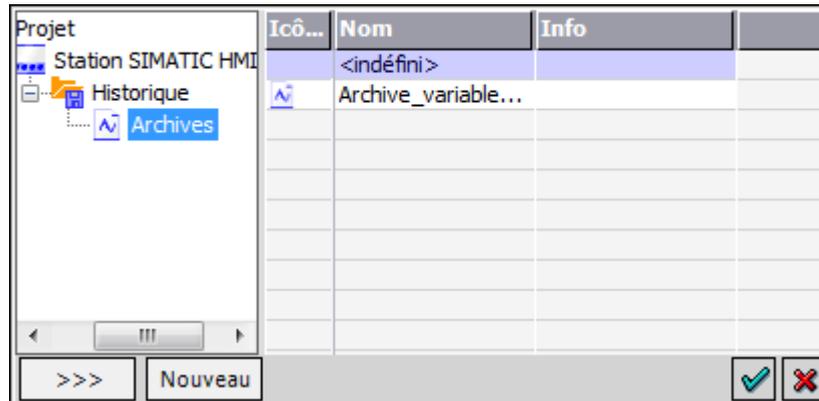


Figure 3.12: La fenêtre de choix de l'archive

- La dernière étape est le choix de cycle d'archivage.

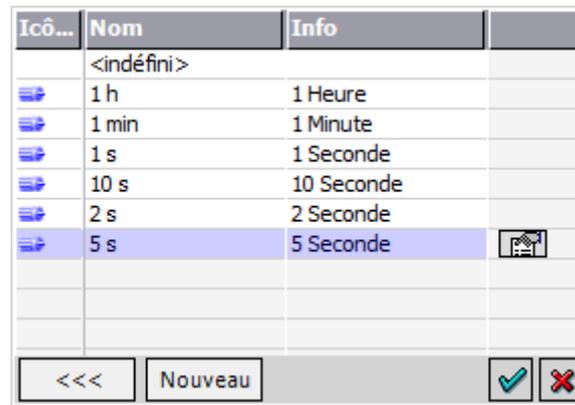


Figure 3.13: La fenêtre de choix de cycle d'archivage

3.B.2.2. Application sur le système pompe-débit :

Dans cette application on va suivre les démarches citées précédemment. On a abouti à ce fichier de données « Bloc-notes ».

"VarName"	"TimeString"	"Varvalue"	"validity"	"Time_ms"
"XQ"	"2016-06-22 14:08:53"	0	1	42543589496.6898
"XQ"	"2016-06-22 14:08:54"	0	1	42543589508.2639
"XQ"	"2016-06-22 14:08:55"	0	1	42543589519.838
"XQ"	"2016-06-22 14:08:56"	0	1	42543589531.412
"XQ"	"2016-06-22 14:08:57"	1.736111E-02	1	42543589542.9861
"XQ"	"2016-06-22 14:08:58"	2.314815E-02	1	42543589554.5718
"XQ"	"2016-06-22 14:08:59"	2.314815E-02	1	42543589566.1458
"XQ"	"2016-06-22 14:09:00"	2.314815E-02	1	42543589577.7199
"XQ"	"2016-06-22 14:09:01"	1.736111E-02	1	42543589589.294
"XQ"	"2016-06-22 14:09:02"	1.736111E-02	1	42543589600.8681
"XQ"	"2016-06-22 14:09:03"	1.736111E-02	1	42543589612.4421
"XQ"	"2016-06-22 14:09:04"	1.736111E-02	1	42543589624.0162
"XQ"	"2016-06-22 14:09:05"	0.1736111	1	42543589635.5903
"XQ"	"2016-06-22 14:09:06"	0.4340278	1	42543589647.1644
"XQ"	"2016-06-22 14:09:07"	1.388889	1	42543589658.7384
"XQ"	"2016-06-22 14:09:08"	2.372685	1	42543589670.3935
"XQ"	"2016-06-22 14:09:09"	3.038194	1	42543589681.9792
"XQ"	"2016-06-22 14:09:10"	3.559028	1	42543589693.5532
"XQ"	"2016-06-22 14:09:11"	3.825232	1	42543589705.1389
"XQ"	"2016-06-22 14:09:12"	4.074074	1	42543589716.7245
"XQ"	"2016-06-22 14:09:13"	4.230324	1	42543589728.2986
"XQ"	"2016-06-22 14:09:14"	4.322917	1	42543589739.8727
"XQ"	"2016-06-22 14:09:15"	4.386574	1	42543589751.4468
"XQ"	"2016-06-22 14:09:16"	4.43287	1	42543589763.0208
"XQ"	"2016-06-22 14:09:17"	4.467592	1	42543589774.5949
"XQ"	"2016-06-22 14:09:18"	4.508102	1	42543589786.169
"XQ"	"2016-06-22 14:09:19"	4.537037	1	42543589797.7431
"XQ"	"2016-06-22 14:09:20"	4.548611	1	42543589809.3982
"XQ"	"2016-06-22 14:09:21"	4.560185	1	42543589820.9722
"XQ"	"2016-06-22 14:09:22"	4.577546	1	42543589832.5463
"XQ"	"2016-06-22 14:09:23"	4.583333	1	42543589844.1204
"XQ"	"2016-06-22 14:09:24"	4.600695	1	42543589855.6944

Figure 3.14: Le fichier de données qui résulte de l'archivage de débit

Dans ce fichier on trouve :

- 1- Le nom de la variable.
- 2- La date et l'heure pour chaque échantion.
- 3- La valeur débit à chaque seconde après le lancement de « *Run time* ».

Conclusion générale

Nous avons vu dans ce mémoire les différentes étapes de régulations du système hydraulique CE117, équipé par un automate programmable S7 314 IFM qui nous a donné la possibilité de l'implémentation et la supervision de différentes commandes.

Notre travail était partagé en trois chapitres :

Le premier chapitre nous a permis d'avoir une bonne connaissance sur les différents systèmes qui existe sur la station CE117 : les contraintes pour chaque système, et finalement après l'identification on a pu connaitre le modèle approprié pour chaque système, ce qui nous a aidés par la suite à la synthèse des régulateurs.

Dans le deuxième chapitre nous avons pu voir l'application de la commande par backstepping et la commande H_∞ .

Dans le troisième chapitre nous avons vu l'implémentation à l'aide des outils de programmation offerts par STEP7, Nous avons vu aussi la supervision des processus à travers l'interface graphique créé sous le WinCC. Et finalement la méthode d'archivage des données sous le WinCC.

Ce projet comporte de nombreux volets et touche à plusieurs disciplines en même temps, c'était une occasion formidable pour nous d'appliquer les connaissances acquises le long de notre formation sur des systèmes réels qui ressemble à ceux qui sont disponible dans l'industrie, et de confronter à des problèmes qu'un ingénieur affronte dans le monde industriel.

En fin, nous espérons que notre travail sera utile aux étudiants d'automatique, pour appliquer ce qu'ils apprennent en cours de l'informatique industrielle et d'Automatique avancé et pourquoi pas d'identification, et qu'il participe à forger des ingénieurs automaticiens avec beaucoup d'expérience qui leurs permettre de passer vers une nouvelle phase où ils seront obligés à manipuler des systèmes réels.

A la lumière des observations et des résultats présentés dans ce mémoire des perspectives intéressantes pouvant contribuer à l'amélioration du fonctionnement de notre dispositif expérimental :

- Programmer une interface pour l'identification des différents systèmes par différentes méthodes de la station sous le WinCC.
- Faire un archivage pour pouvoir identifier les différents systèmes avec des cycles inférieurs à 1 s en utilisant un script sur le WinCC.

Bibliographie

- [1] Niedercorn LT « la Briquerie » 57100 THIONVILLE-Automates programmables-présentation.
- [2] Slim BEN SAOUD : LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API).
- [3] SIEMENS, Notions de base sur les bus de terrain avec SIMATIC S7-300, Annexe IV, Edition 05/2004
- [4] SIEMENS SIMATIC : Programmer avec STEP 7 Manuel
- [5] WinCC flexible 2008 SP2, WinCC flexible Information System.
- [6] www.siemens.com
- [7] : Automatique Contrôle et Régulation / Patrick Prouvost
- [8] FAHED ESHBAIR ‘ ‘ MODÉLISATION ET COMMANDE D'UN SYSTÈME MULTI-MOTEUR PAR LA TECHNIQUE DE COMMANDE *BACKSTEPPING* ’ ’ EDITION AOÛT 2005
- [9] M.A.A.FOUKA, B.FERHAOUI, «REALISATION D'UNE MAQUETTE DE STATION DE POMPAGE A BASE D'AUTOMATE PROGRAMMABLE SIEMEN », projet de fin d'étude, Ecole Nationale Polytechnique 2015.
- [10] B.A.SAHAR, A.SOUCHANE «Commande et supervision de la station FESTO PCS_COMPACT à l'aide de STEP7 et WinCC», projet de fin d'étude, Ecole Nationale Polytechnique 2009.
- [11] M.BOUALBANI «Commande et supervision de la station CE117 Process Trainer avec l'automate s7 300», projet de fin d'étude, Ecole Nationale Polytechnique 2015.
- [12] A.ABRICHE, « Réalisation et gestion d'un prototype de station de pompage à base d'automates programmables industriels SIEMENS », projet de fin d'étude, Ecole Nationale Polytechnique 2007.