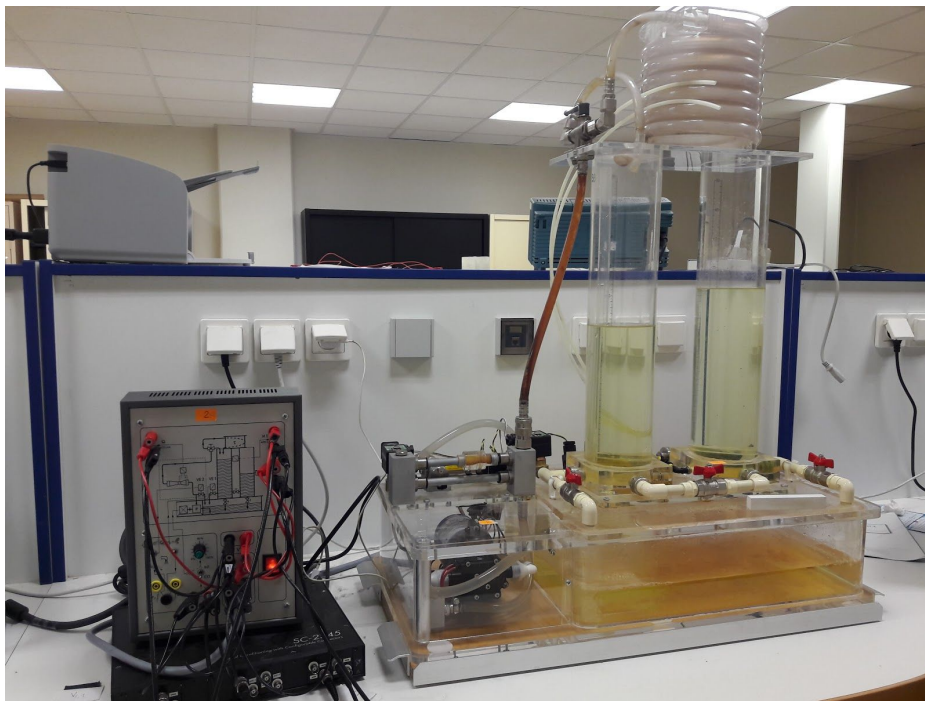


Claire VANDAMME
Alexandra VILLA LUCERO

IMA4
2017/2018

Rapport de projet P60 : Commande de niveau d'eau



Responsable : Pekpe Komi MIDZODZI

Remerciements

Tout d'abord, nous tenons à remercier l'équipe pédagogique de notre formation Informatique, Microélectronique et Automatique (IMA) qui nous a enseigné les savoirs et connaissances utiles et nécessaires pour la réalisation de ce projet.

Ensuite, nous souhaitons remercier Monsieur Pekpe Komi MIDZODZI d'avoir proposé ce sujet ainsi que son aide et conseils tout au long du projet.

Enfin, un merci particulier à Monsieur Laurent ENGELS pour le temps qu'il nous a accordé pour la réalisation de notre vidéo.

Sommaire

Remerciements	1
Introduction	3
1. Présentation du projet	4
1.1. Présentation générale du système	4
1.2. Éléments du système	5
1.3. Choix techniques : matériel et logiciel	6
1.4. Etapes du projet	6
2. Simulink : Modélisation, filtrage et étalonnage	7
2.1. Modélisation	7
2.2. Filtrage du signal de sortie	8
2.3. Etalonnage	8
3. Identification du système	10
3.1. Fonction de transfert	10
3.2. Modèle d'état	11
4. Observateur et modèle hors ligne	12
4.1. Création de l'observateur	12
4.2. Modèle hors ligne	13
5. Problèmes rencontrés	15
Conclusion	16
Références	16

Introduction

Pour notre 4^{ème} année de formation ingénieur à Polytech Lille, en Informatique Microélectronique et Automatique, nous devons réaliser un projet tout au long du semestre 8. Nous avons choisi le sujet sur la commande de niveau d'eau qui utilise un système de deux cuves.

Le principal objectif de notre projet est de créer un observateur, c'est-à-dire reconstruire des valeurs dont on ne dispose pas directement (débit de sortie, niveau d'eau, ...) à partir de valeurs mesurées et de celles dont on dispose à partir du modèle du système qui sera identifié.

Cet observateur va nous permettre de détecter s'il y a des anomalies (exemple : fuites) dans le système lors d'une simulation grâce à un capteur logiciel. Avec celui-ci nous allons pouvoir d'ailleurs, voir la différence des valeurs entre le capteur physique et le capteur logiciel obtenu par l'observateur pour savoir si cela est cohérent. Et pouvoir alors gérer les pannes de capteurs s'il y en a.

Tout d'abord dans ce rapport, nous ferons une présentation détaillée du projet, où nous réaliserons la présentation du système avec les éléments qui le compose pour savoir quel rôle joue chaque composant, nos choix techniques et les étapes pour identifier l'observateur. Ensuite, nous continuerons avec la manière dont nous avons implémenté le système sous Simulink afin de le commander, en détaillant le filtrage et étalonnage dans cet étape.

Par la suite, nous développerons l'identification du système par la fonction de transfert et le modèle d'état.

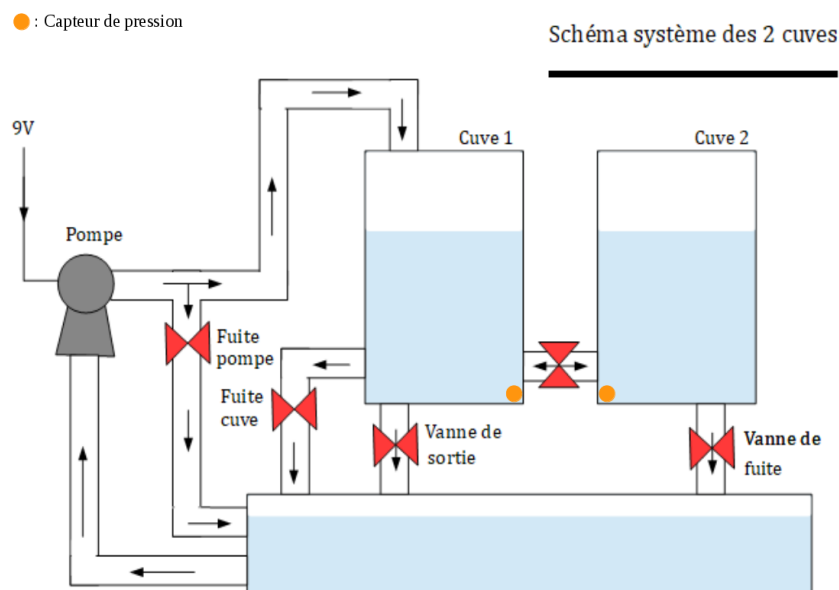
Enfin, nous parlerons de la création de l'observateur et du modèle hors-ligne qui permettent de répondre au but de notre projet. Ainsi qu'un point sur tous les problèmes rencontrés lors de ce semestre. Nous terminerons ce rapport par une petite conclusion et ressenti sur le projet.

1. Présentation du projet

1.1. Présentation générale du système

Pour ce projet, nous avons eu à disposition la maquette modélisant un système de 2 réservoirs, présent en C008. Celui-ci est composé de deux cuves, six vannes, d'un serpentin de retard, d'une pompe et d'un panneau de commande.

Ce système est commandé par un ordinateur équipé d'une carte d'acquisition National Instrument, et du noyau Temps Réel dans le logiciel Matlab.



Le système est composé de 2 cuves, dont la première est remplie en eau par le biais d'une pompe, elle-même alimentée par un moteur dont on choisira la tension. La pompe est reliée à une grosse cuve qui sert en quelque sorte de réservoir afin de récupérer l'eau qui sera ensuite vidée dans ce même réservoir lorsqu'on ouvre les différentes vannes.

Il y a une sortie principale du système : Vanne de sortie. Le reste des sorties ne simulent que les fuites, comme "Vanne de fuite" et "Fuite cuve". Nous avons également "Fuite pompe", qui génère aussi une fuite. "Fuite pompe" et "Vanne fuite" sont toutes les deux des électrovannes à contrôler par le tableau de commande, alors que "Vanne de fuite" est une vanne manuelle. Il reste une vanne manuelle, qui relie les 2 cuves ensemble, elle peut être fermée si on ne veut remplir que la première cuve (pour travailler sur un premier ordre) ou ouverte pour remplir les deux cuves.

Le système dispose également de deux capteurs de pression (un dans chaque cuve) permettant d'avoir des données afin de connaître le niveau d'eau de chaque cuve.

1.2. Éléments du système

Notre système possède des éléments "internes" comme les capteurs de pression mais également des éléments "externes" permettant de faire le lien entre interface/commande logiciel et le système en lui-même.

Les principaux composants que nous allons détailler sont :

- La carte d'acquisition PCI 6024-E
- La série SC-2345
- Les capteurs de pression

Carte d'acquisition PCI 6024-E :

Le système utilise la carte d'acquisition de données de la série NI SC-2345 de National Instruments. La carte prend la mesure à haute tension à partir de capteurs (dans notre cas Pressure Transmitters P28) avec le conditionnement des signaux du NI SCC. Elle arrive à obtenir un taux d'échantillonnage jusqu'à 200k éch./s, avec une résolution de 12 bits sur 16 entrées analogiques asymétriques. En fonction de votre disque dur, le débit de transfert en continu de la carte PC 6024 peut atteindre 200k éch./s.

Série SC-2345 :

Le SC-2345 est doté de connecteurs configurables qui acceptent des mini-panneaux de connexion d'E/S, pour une connectivité directe aux signaux/capteurs. Ces boîtiers pour modules de conditionnement de signaux SCC se connectent directement aux matériels d'acquisition de données. Sa fonction est de conditionner des signaux analogiques et numériques de notre dispositif d'acquisition de données (PCI 6024-E)

Capteurs de pression :

Le système utilise un capteur de type "Transmetteur de pression PCE-28" pour chaque cuve. Ce capteur est capable de mesurer la pression, sous-pression et pression absolue d'un liquide et de donner une sortie analogique.

Les 2 capteurs sont reliés au panneau de commande (SC-2345) comme suit :

- Bornes "+" à l'alimentation 24 VCC
- Capteur H1 à l'entrée E2 (ach3)
- Capteur H2 à l'entrée E3 (ach9)
- Borne du signal d'entrée du système à l'entrée E1 (ach0)
- L'entrée S au moteur de la pompe (DAC0)

1.3. Choix techniques : matériel et logiciel

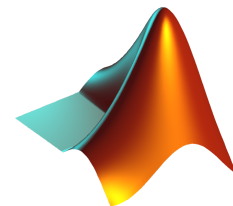
Au niveau du matériel, nous n'avons besoin que de la maquette du système des 2 cuves, avec tous ses éléments afin de l'asservir via l'ordinateur.

Au niveau du logiciel, comme nous utilisons le matériel de National Instruments, il nous est possible d'avoir un environnement de développement d'application et un choix d'applications logicielles souples. Nous avons donc utilisé MATLAB (« matrix laboratory »), qui est un langage de programmation de quatrième génération émulé par un environnement de développement du même nom ; il est utilisé à des fins de calcul numérique. Matlab peut s'utiliser seul ou bien avec des toolbox (« boîte à outils »). Dans notre projet, nous utilisons Simulink, qui est un logiciel de modélisation système multi-physique intégré à MATLAB, ainsi que le noyau Temps Réel.

Ce qui nous a permis de développer un système pour les applications de test, de mesure et de contrôle/commande, permettant d'accéder rapidement au matériel et aux informations sur les données, de plus, on peut afficher les signaux d'E/S.

La fenêtre "Command Window", qui est un des éléments du bureau MATLAB, nous permet d'exécuter des commandes simples.

Des séquences de commandes peuvent être aussi sauvegardées dans un fichier texte, sous la forme d'un « script » ou encapsulées dans une fonction pour obtenir des calculs.



1.4. Etapes du projet

1ère étape :

Modélisation mathématique du système à l'aide des lois physiques puis estimation des paramètres du modèle à l'aide des données (débit d'entrée et hauteurs mesurées)

2ème étape :

Conception de l'observateur du système (à l'aide du modèle d'état et de la méthode de l'espace de parité) et son implémentation sous Matlab Simulink.

3ème étape :

Génération des signaux indicateurs de défauts appelés résidus à partir des mesures recueillies directement sur le système physique et données générées par l'observateur.

4ème étape (facultatif) :

Mise en place d'un test statistique pour générer des alarmes

2. Simulink : modélisation, filtrage, étalonnage

2.1. Modélisation

Pour modéliser notre système sur Simulink, nous allons procéder étape par étape en identifiant chaque élément physique en un bloc correspondant sur le fichier Simulink.

Tout d'abord pour activer la pompe, il nous faut un signal d'entrée.

Celui-ci peut être simulé par un step que l'on va initialiser à 5V arbitrairement, ce qui est suffisant afin de faire tourner le moteur. Pour que ce signal soit reçu par le système réel, on va y mettre un bloc "Analog Output" en précisant en paramètre sur quel "channel" les données seront envoyés.

On va donc pouvoir envoyer la donnée d'entrée au moteur de la pompe qui elle même alimentera en eau la première cuve.

On veut récupérer en sortie les valeurs que nous envoient les deux capteurs de pressions.

Pour cela, on va ajouter deux blocs "Analog Input" avec encore une fois en paramètre le "channel" correspondant sur le panneau de commande.

Nos entrées/sorties sont donc maintenant modélisées.

En plaçant un scope à chaque sortie, on remarque que l'on a un signal très bruité et en volts. On a besoin de sorties en cm pour le niveau d'eau afin de faire le traitement nécessaire par la suite.

On va donc devoir filtrer le signal afin de minimiser le bruit et réaliser un étalonnage pour avoir des cm en sortie et non des volts.

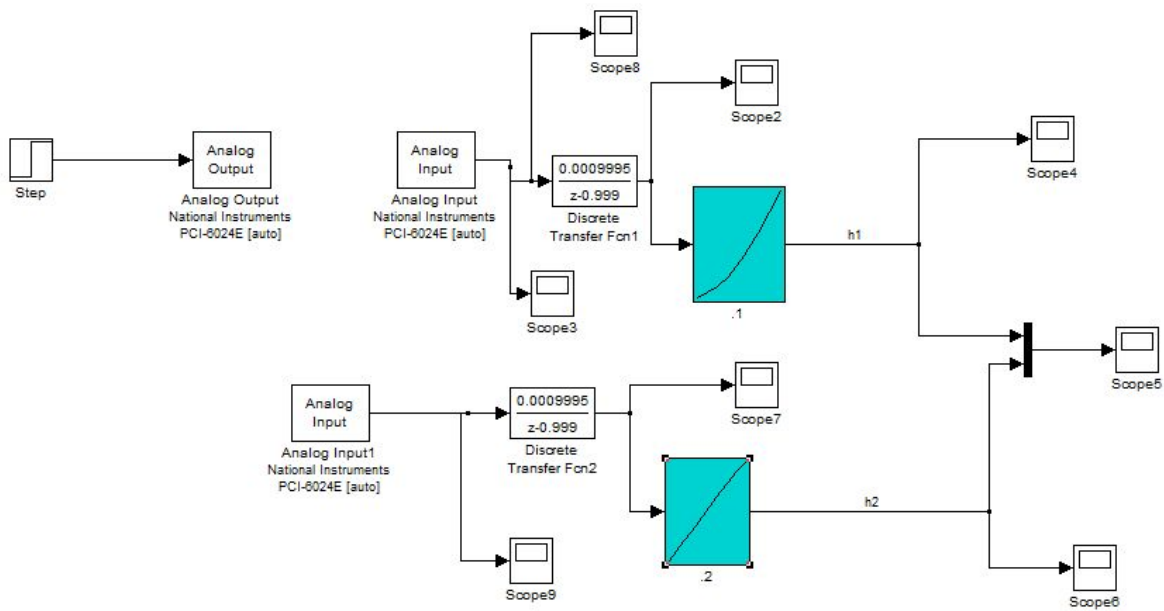
Pour minimiser le bruit, on ajoutera le bloc "Discrete Transfer Function", qui prend en paramètres les coefficients du numérateur et du dénominateur de la fonction de transfert.

Enfin, afin de réaliser l'étalonnage du signal maintenant filtré, on va implémenter un bloc "Function Block Parameters", qui utilise la méthode d'interpolation.

Ces 2 blocs seront plus détaillés dans les deux prochaines parties.

On a donc maintenant en sortie, deux signaux correspondant au deux niveau d'eau en cm de chacune des cuves.

Tous ces blocs seront regroupés en un sous système nommé "Modèle Boîte Noire" afin de rendre plus lisible le fichier Simulink global.



2.2. Filtrage du signal de sortie

Pour résoudre ce problème de bruit sur nos signaux de sortie, nous réalisons un filtre à l'aide de la fonction de transfert.

On va alors utiliser la fonction $tf()$ de MATLAB afin de calculer la fonction de transfert pour déterminer les paramètres du filtre.

Puis, on utilise la commande $c2d()$ pour passer d'un résultat à temps continu à un résultat à temps discret car c'est le cas de notre système.

Une fois ceci calculé, on insère les valeurs dans les blocs correspondants.

Bien sur, il faut faire ces opérations pour les 2 sorties, mais on a pu remarquer que l'on trouvait au final les mêmes valeurs pour les fonctions.

2.3. Etalonnage

L'étalonnage va permettre de donner une relation de valeurs entre la tension reçue et la hauteur en cm de la cuve.

Ceci est essentiel pour la suite du projet car la sortie est pour l'instant en volt et pour créer le modèle d'état on aura besoin de la sortie en cm.

Projet P60 : "Commande de niveau d'eau"

Pour réaliser cet étalonnage, on procède de la manière suivante :

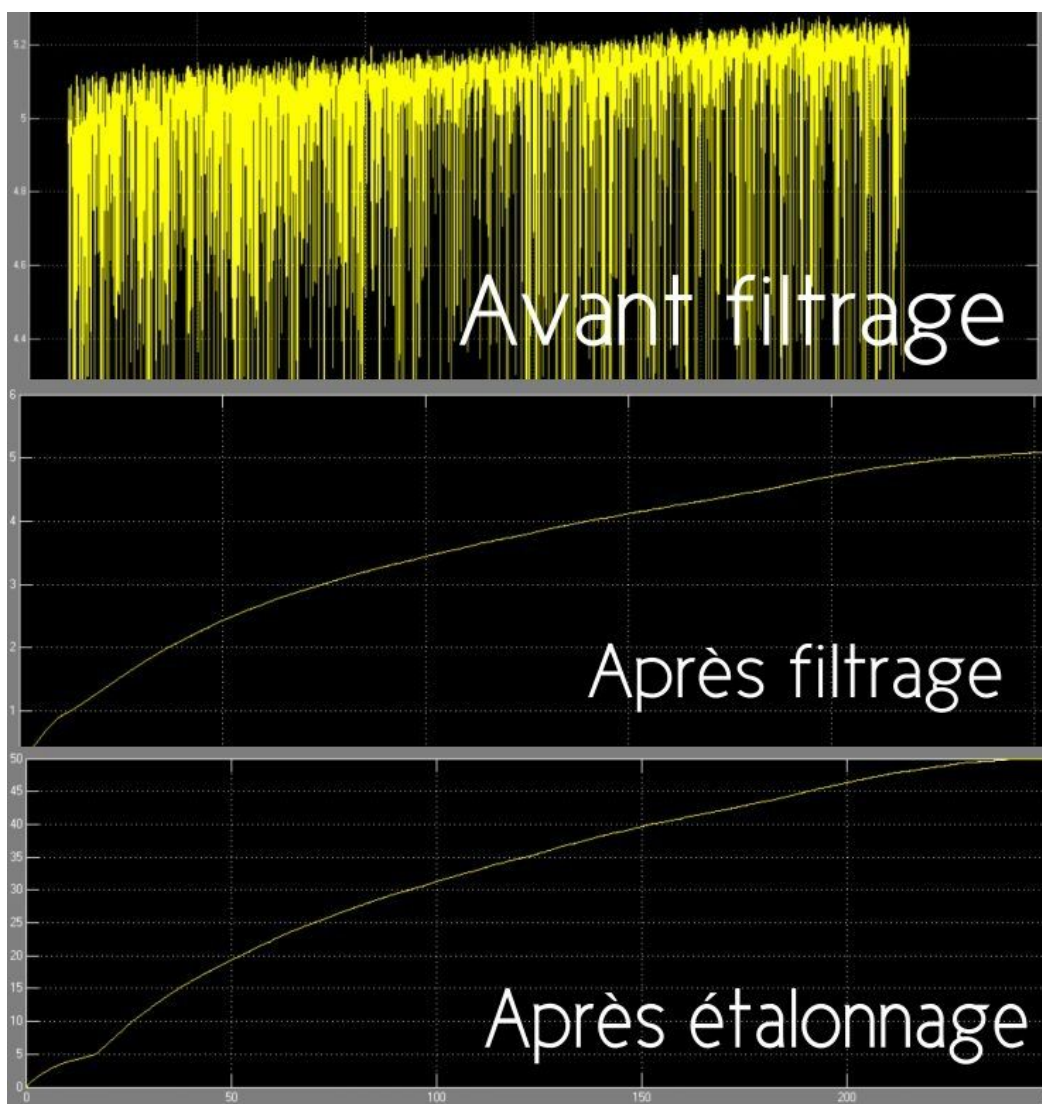
Tout d'abord, on place deux scopes pour visualiser les données envoyées par chacun des capteurs.

On lance le système à une tension constante et tous les 5 cm on récupère la valeur en tension de chacun des capteurs. Pour plus de précision, on arrête le système à chaque mesure, sinon nous n'avons pas le temps de prendre la mesure correctement sur le scope placé en sortie.

Nous avons dû réaliser cette étape à multiples reprises suite à divers problèmes que l'on expliquera plus tard dans le rapport.

Au final, on a réalisé l'étalonnage avec une entrée de 6V car c'est la valeur maximale que l'on utilise dans le projet.

Voici alors un exemple de courbes où l'on peut voir la différence du signal après filtrage, puis après étalonnage :



3. Identification du système

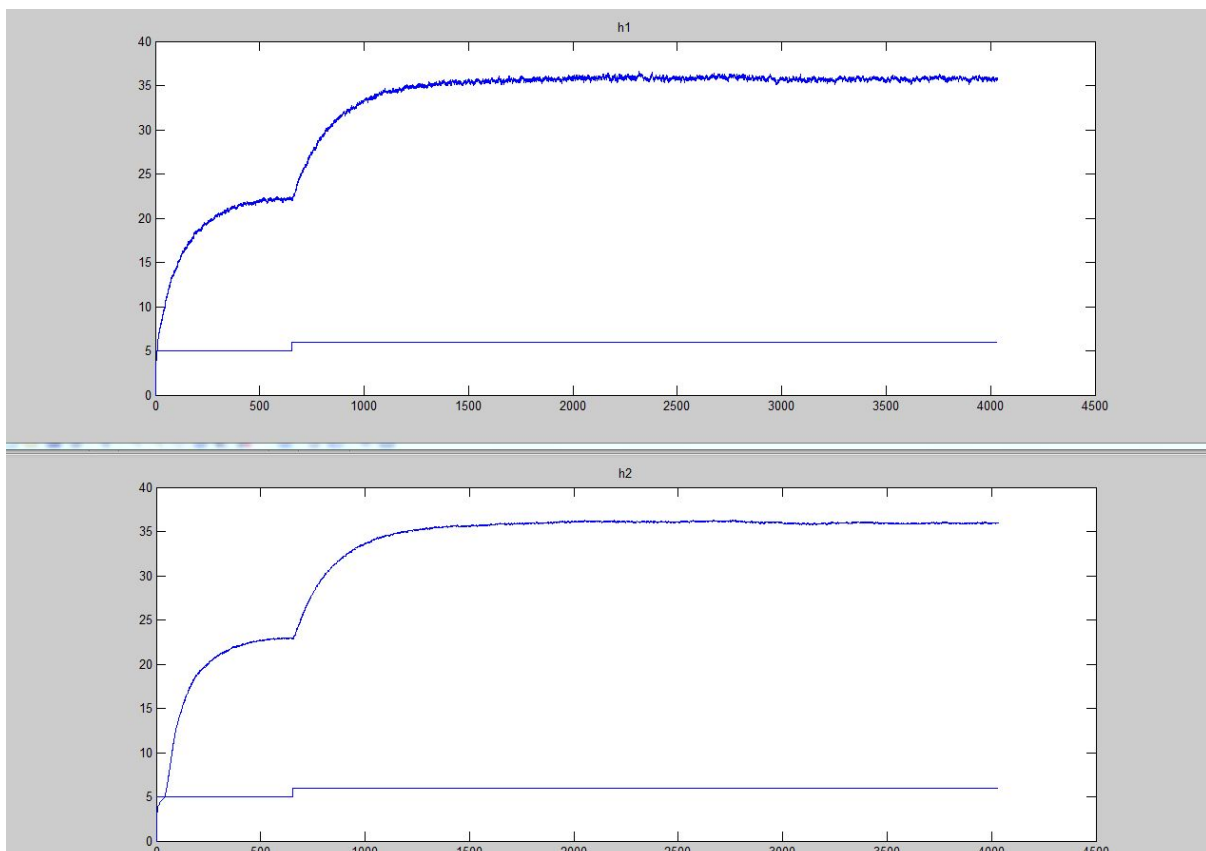
3.1. Fonction de transfert

Pour déterminer la fonction de transfert, cela dépend du type de système que l'on a. Ici, on a un système non linéaire, on ne peut donc pas directement trouver la fonction de transfert. On va alors prendre une partie linéaire du modèle, avec comme origine le système se stabilisant pour une tension d'entrée de 5V et on travaillera au voisinage de cette valeur.

Ensuite, on a un système de deux cuves qui symbolisent deux éléments de stockage d'énergie, donc il s'agit d'un système dit du deuxième ordre.

Pour déterminer les paramètres d'une telle fonction de transfert, on lance une simulation pendant 10mn à 5V afin de stabiliser le système puis on change le signal à 6V et c'est la partie qui va nous intéresser pour les paramètres de la fonction de transfert.

On obtient la courbe suivante :



On observe que la courbe, une fois que l'on est passé à 6V, est plus proche d'un premier ordre que d'un second ordre.

On va donc utiliser les formules pour un premier ordre.

On détermine alors K, le gain correspondant à l'amplitude du signal et τ , le temps de réponse à 63%.

On trouve donc comme fonction de transfert (l'application numérique est faite ici pour la première cuve) :

$$y(t) = k \left(1 - e^{-\frac{t}{\tau}} \right) = 14 \left(1 - e^{-1/200} \right)$$

3.2. Modèle d'état

En automatique, une représentation d'état permet de modéliser un système dynamique en utilisant des variables d'état. Cette représentation, qui peut être linéaire ou non, continue ou discrète, permet de déterminer l'état du système à n'importe quel instant futur si l'on connaît l'état à l'instant initial et le comportement des variables exogènes qui influent sur le système. La représentation d'état du système permet de connaître son comportement "interne" et pas seulement son comportement "externe" comme c'est le cas avec sa fonction de transfert.

La représentation d'état de ces systèmes, s'écrit de la manière suivante :

$$\begin{cases} \dot{x} = A x + B u \\ y = C x + D u \end{cases}$$

Comme on a en donnée que la fonction de transfert de chaque cuves, on va l'utiliser afin d'identifier les variables de la fonction de transfert avec les variables du modèle d'état.

On a alors comme formule théorique pour un système du premier ordre :

$$\frac{Y(s)}{U(s)} = \frac{k}{1 + \tau p} = \frac{\frac{k}{\tau}}{p + \frac{1}{\tau}} = \frac{b_0}{a_1 p + a_0}$$

Avec cette formule, on peut maintenant identifier les variables du modèle d'état avec :

- A qui prend la valeur de a_0
- B qui prend la valeur de a_1
- C qui prend la valeur de b_0
- D qui prend la valeur de b_1 (qui ici est nulle)

Ce qui va créer 2 modèles d'état, un pour chaque cuve, et l'idée maintenant et de regrouper ces 2 modèles en un en posant :

$$X = \begin{bmatrix} X1 \\ X2 \end{bmatrix} \quad Y = \begin{bmatrix} Y1 \\ Y2 \end{bmatrix} = \begin{bmatrix} H1 \\ H2 \end{bmatrix}$$

Et on se trouve au final avec :

$$\dot{X} = \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} A1 & 0 \\ 0 & A2 \end{bmatrix} \begin{bmatrix} X1 \\ X2 \end{bmatrix} + \begin{bmatrix} B1 \\ B2 \end{bmatrix} u = A x + B u$$

$$Y = \begin{bmatrix} Y1 \\ Y2 \end{bmatrix} = \begin{bmatrix} C1 & 0 \\ 0 & C2 \end{bmatrix} \begin{bmatrix} X1 \\ X2 \end{bmatrix} = C x$$

Ce modèle d'état se trouvera dans le sous-système du modèle boîte noire.

4. Observateur et modèle hors ligne

4.1. Création de l'observateur

Pour créer l'observateur, on va utiliser la méthode d'espace de parité.

Celle-ci se traduit théoriquement par les équation suivantes :

$$X = AX + BU$$

$$Y = CX + DU \Rightarrow \dot{Y} = C\dot{X} + B\dot{U} = C(AX + BU) + D\dot{U} = CAX + CBU + D\dot{U}$$

$$\begin{aligned} \ddot{Y} &= CA\dot{X} + CB\dot{U} + D\ddot{U} \\ &= CA(AX + BU) + CB\dot{U} + D\ddot{U} \\ &= CA^2X + CABU + CB\dot{U} + D\ddot{U} \\ &= CA^2X + [CAB \quad CB \quad D] \begin{bmatrix} U \\ \dot{U} \\ \ddot{U} \end{bmatrix} \end{aligned}$$

$$Y^3 = \dots$$

$$Y^n = \dots$$

Comme on travaille sur un système que l'on simplifie, du premier ordre, n=1, donc on ne dérive qu'une fois l'équation.

$$\begin{bmatrix} Y \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} C \\ CA \end{bmatrix} X + \begin{bmatrix} D & 0 \\ CB & D \end{bmatrix} \begin{bmatrix} U \\ \dot{U} \end{bmatrix}$$

$$\Leftrightarrow Y = \gamma X + HU$$

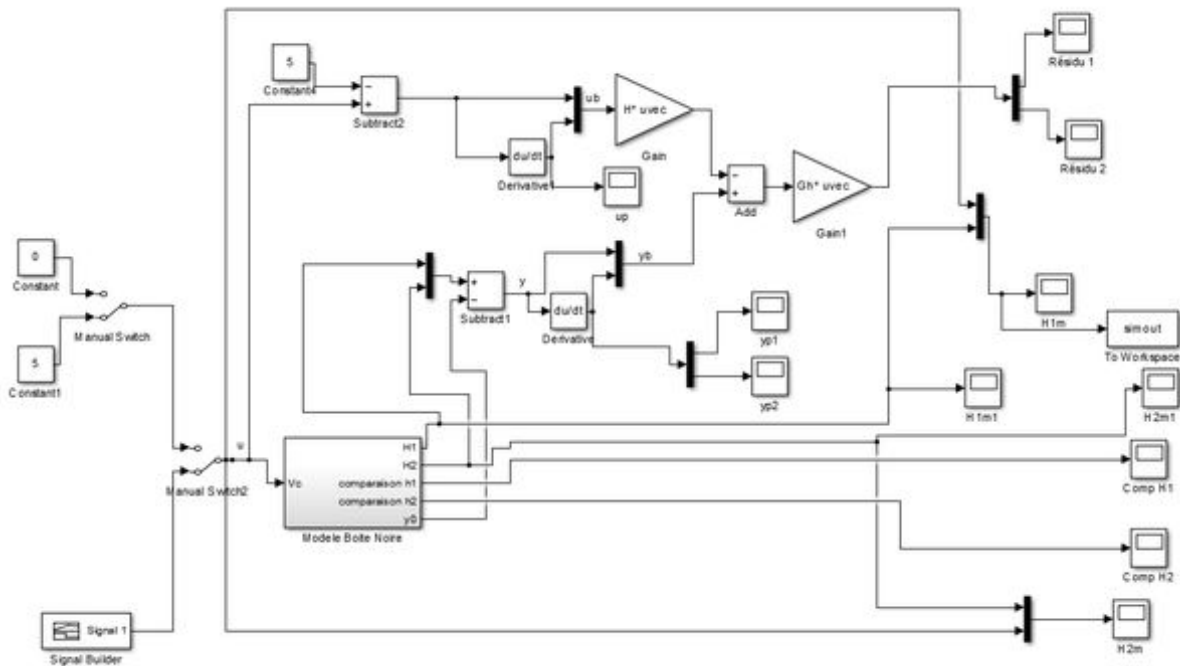
C'est à partir de cette équation que l'on va trouver l'équation de l'erreur.

Après quelques calculs (détaillés dans le wiki), et l'ajout d'un défaut, on trouve l'équation suivante :

$$E = \gamma^t f = \gamma^t Y - \gamma^t HU$$

où f représente le défaut.

Maintenant que l'on a cette équation, on a pu l'ajouter directement dans notre fichier Simulink, de la manière suivante :



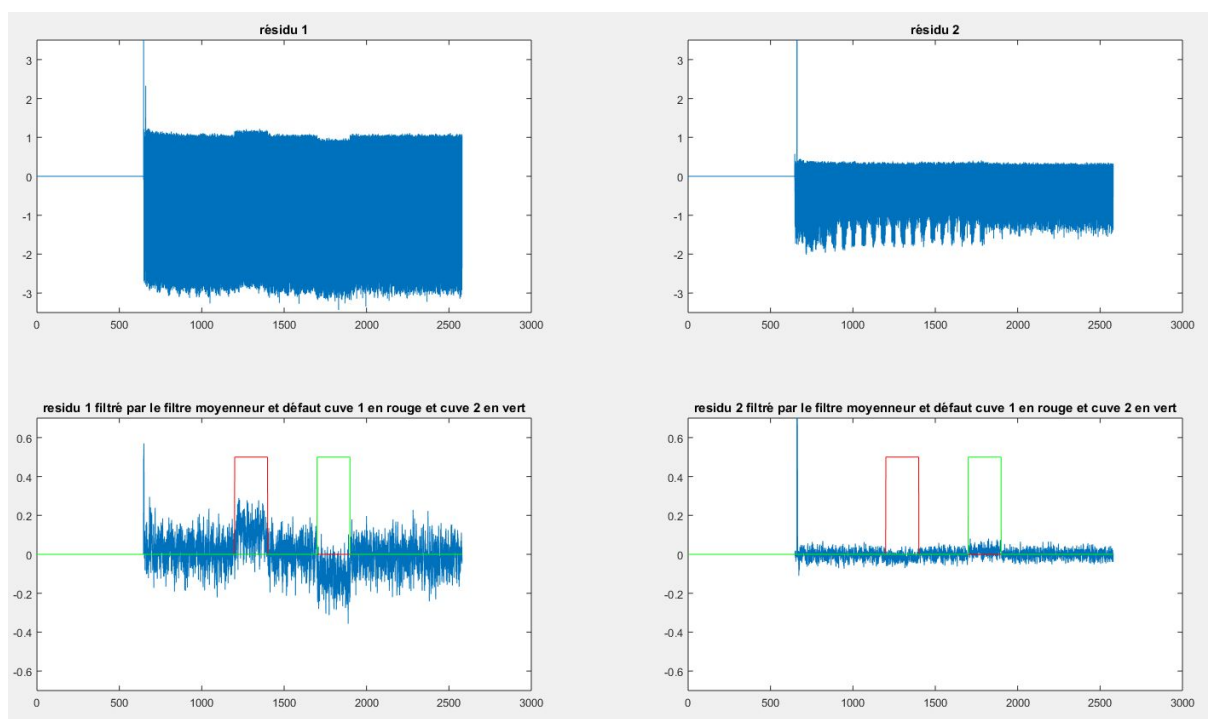
4.2. Modèle hors-ligne

Afin de pouvoir utiliser notre modèle sans forcément avoir le système nous avons réalisé un script Matlab pour pouvoir effectuer le même traitement que précédemment (modèle d'état + observateur) en chargeant des données préalablement enregistrées grâce au système réel.

Ce modèle a été réalisé en décomposant ce que l'on faisait dans le fichier Simulink bloc par bloc, et en le traduisant en équation, boucle for, ... afin d'en faire un script Matlab.

Nous avons ajouté dans le modèle hors-ligne (que l'on a pas dans le fichier Simulink) un filtre moyenneur pour minimiser les bruits concernant les signaux des résidus.

On obtient alors les courbes suivantes :



On remarque donc que la courbe augmente ou diminue selon si on ajoute des défauts ou non.

Cela prouve que nous pouvons créer un avertissement lorsqu'il y a détection de défaut comme ici une fuite.

5. Problèmes rencontrés

Au début du projet, nous ne savions pas que le système de deux cuves n'était pas constant et précis à chaque fois que l'on faisait des tests.

A cause de ce problème les valeurs de la hauteur en cm varient pour une même valeur de tension en entrée. Nous avons donc décidé de faire plusieurs fois les tests pour arriver au résultat le plus précis et de choisir une valeur moyenne pour l'étalonnage.

La variation de la hauteur affecte aussi la valeur du gain de la fonction transfert.

Par conséquent, notre système ne sera bien identifié. Pour résoudre ce problème, on a mis plusieurs hauteurs possibles et pendant le test avant de passer au "Signal Builder", on regarde la bonne valeur en cm des deux cuves et on l'ajuste directement dans le fichier Simulink.

Les données des capteurs de pression varient si la force de l'eau varie et ces données ne sont pas prévisibles, donc il faut faire une mise à jour sur quelques variables dans la modélisation du fichier Simulink.

De plus, le signal d'entrée peut varier de 0 à 9 volts, mais la pression de l'eau varie aussi. Pour cette raison il faut choisir une valeur moyenne.

Comme le système prend du temps pour se stabiliser alors il faut attendre 10 minutes à 5 volts pour que le signal de sortie soit constant et après on choisit un signal sinusoïdal de 5-6 volts qui sera le même pendant tout le test.

On pense également que les multiples problèmes de précision (comme la stabilisation qui ne se fait pas à la même hauteur) sont dûs aux composants qui commencent à être un peu vieux et donc qu'ils ne seraient plus calibrés correctement, comme cela doit être le cas de la pompe et des capteurs.

En passant à un matériel un peu plus moderne, cela permettrait d'avoir une meilleure précision et un gain de temps, car nous n'aurions pas à refaire les tests plusieurs fois.

Enfin, le projet nous a pris plus de temps que prévu lors des simulations, car l'ordinateur de la salle C008 ne contient pas assez mémoire pour travailler avec Matlab et avec une autre application (par exemple Internet). Il saturait et plantait après parfois quelques minutes ou des dizaines de minutes.

Dans ces cas là, le PC s'éteignait brutalement et nous étions obligés de relancer toute la simulation qui pouvait parfois prendre 40mn.

De part son manque de mémoire, le PC pouvait être aussi très long ne serait-ce que pour charger les données.

N'ayant pas également une version de Matlab des plus récentes (2010), certaines fonctions n'était pas disponible, alors qu'elles auraient pu nous faciliter certaines étapes.

Conclusion

Pour conclure nous avons créer un observateur sur un système de deux cuves comme projet de 4ème année à Polytech Lille.

Le projet consistait à créer un observateur qui détecte des anomalies, dans notre cas des fuites, et déterminer l'erreur du signal de sortie grâce aux données fournies par les capteurs de pression et la modélisation dans Simulink pour commander le système et l'analyser.

Grâce à notre formation nous avons réussi à réaliser ce projet, et l'orientation de notre tuteur Mr. MIDZODZI nous a permis de comprendre comment on peut manipuler le système manuellement et comment on peut nous même le commander avec l'ordinateur spécifiquement dans Simulink.

Ce projet nous a également permis d'élargir et renforcer nos connaissances sur Matlab, qui était un outil important pour obtenir des valeurs plus rapidement, créer des scripts pour travailler avec les données obtenues et visualiser tous les signaux avant et après leur traitement grâce à l'oscilloscope.

Nous avons travaillé avec la détection de fuites pour ce projet, toutefois dans l'industrie, on pourra trouver plusieurs types de défauts, qu'il faut détecter à l'avance car il est très important de corriger tôt le problème avant que celui-ci ne cause plus de dégâts dans le système.

Finalement, nous avons fais face à des problèmes techniques comme théoriques au cours de ces semaines de travail. Nous recommandons que le système soit vérifié par des personnes qualifiées pour améliorer les résultats et optimiser le temps d'exécution des tests.

Références

<http://www.rampacek.fr/www/IMG/pdf/pjx.pdf>

<http://www.pumpagents.com/pdf/ShurfloPumps/2095-212-234.pdf>

<http://www.aplisens.com/pdf/produkty/PR-28.pdf>

https://www.asconumatics.eu/images/site/upload/_en/pdf1/00031gb.pdf

<http://www.ni.com/pdf/manuals/371064f.pdf>