

## **Projet IMA4: Système hydraulique**



École Polytechnique universitaire de Lille

Qiaomu LI & Abdel-hakim MAHIR

Encadrant: Belkacem Ould-Bouamama

--Année scolaire 2012/2013--

# Remerciements

Toutes les personnes nous ayant permis de mener à bien ce travail sont assurées de notre gratitude.

Nous adressons tout d'abord nos profonds remerciements à Monsieur Belkacem OULD BOUAMAMA qui a été notre encadrant et nous a suivi avec beaucoup d'attention, ayant dirigé notre travail de façon efficace tout au long de notre projet.

Nous tenons aussi à remercier des thésards de l'école centrale avec qui nous avons été amenés à travailler, pour l'aide qu'ils nous ont donné, leurs précieux conseils ainsi que leur disponibilité.

Enfin, nous remercions également les membres du Jury qui nous font l'honneur de participer à la soutenance.

# Sommaire

Introduction .....	04
I-Présentation du projet.....	05
II-Modélisation avec Bond Graph.....	06
III-Réalisation du modèle réel.....	09
IV-Étude de la boîte à outil 'ident'.....	12
V-Contrôle du niveau d'eau avec PID.....	14
VI-Régulation cascade.....	16
VII-Commande par retour d'état.....	18
Conclusion .....	21
Annexes.....	22

# Introduction

Les systèmes hydrauliques sont utilisés dans beaucoup de domaine industriel tel que le domaine énergétique avec les centrales, le domaine mécanique avec les vérins ou encore le domaine du stockage avec les réservoirs. Ces domaines sont les plus utilisés et les plus demandés car ils continuent à augmenter et à s'améliorer. Il est donc nécessaire pour nous de comprendre leur fonctionnement et savoir les utiliser au niveau contrôle.

Dans le cadre de notre projet d'étude de quatrième année d'ingénieur en spécialité IMA (Informatique, microélectronique et automatique), nous devons :

- Elaborer le modèle dynamique du système hydraulique réel présent dans la salle C008
- Valider en temps réel le modèle dynamique Bond Graph
- Utiliser une boîte à outil temps réel de Matlab RTW pour l'identification des paramètres du modèle
- Contrôler le niveau d'eau des réservoirs
- Implémenter en temps réel une commande cascade
- Réaliser une interface de supervision assez agréable à l'œil et qui soit compréhensible facilement

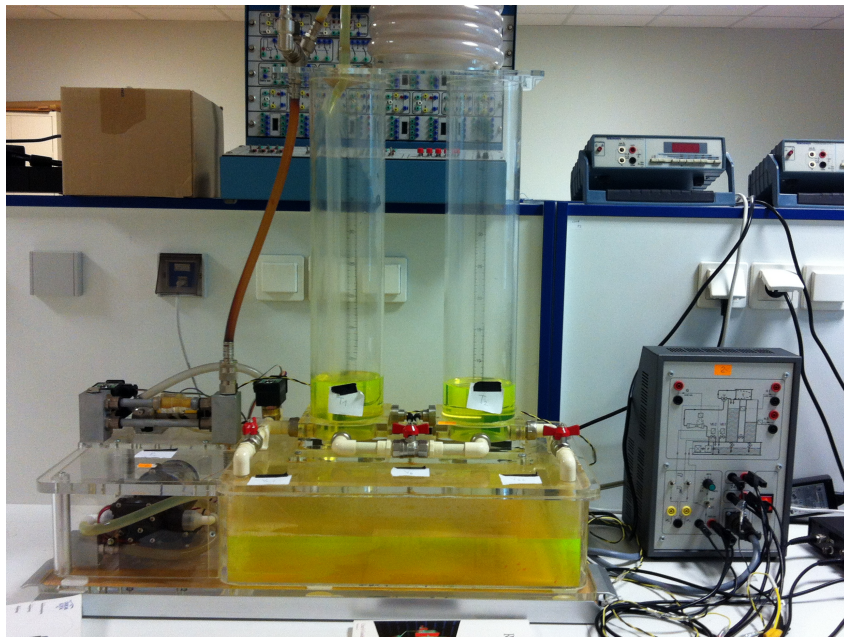
# I-Présentation du projet

L'objectif de notre projet est donc dans un premier temps de valider en temps réel un modèle dynamique 'Bond Graph' pour l'implémentation en temps réel d'un ensemble de commandes simples et avancées. On utilisera une interface de supervision.

Le projet est réalisé dans le cadre des applications temps réel des algorithmes de commandes avancées étudiés en 4ème année. Le système réel est un système hydraulique à deux réservoirs alimentés par une pompe commandée à vitesse variable. Le but est de mettre en application les algorithmes de commande et de modélisation par 'Bond Graph'.

Nous avons à notre disposition ce système hydraulique qui peut être commandé de manière manuelle ou, de manière automatique à l'aide d'un logiciel pouvant créer une interface ou encore un schéma bloc servant à asservir le système.

Le système est composé de deux réservoirs, réservoir 1 et réservoir 2, de trois vannes, d'un serpentín de retard, d'une pompe et d'un panneau de commande qui vous sont montrés ci-dessous :



Il y a deux capteurs de niveau (capteur 1 et capteur 2) pour le réservoir 1 et le réservoir 2.

La configuration est telle que la vanne 1 (la sortie du réservoir 1) est ouverte, la vanne b (vanne entre réservoir 1 et réservoir 2) est ouverte et la vanne 2 (la sortie du réservoir 2) est fermée.

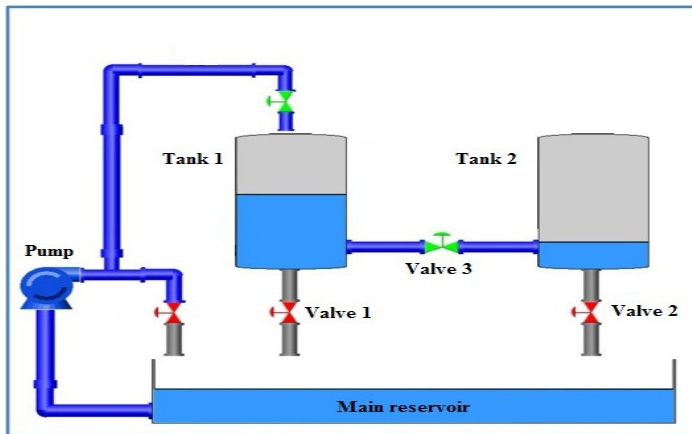
Pour ce projet, nous devons donc tout d'abord faire un modèle théorique du système puis le valider. Ensuite nous devons contrôler le niveau d'eau dans les deux réservoirs.

## II-Modélisation Bond Graph

L'outil Bond Graph est un langage graphique unifié pour tous les domaines des sciences de l'ingénieur. La modélisation par Bond Graph ne nécessite pas l'écriture des lois générales de conservation. Elle repose essentiellement sur la caractérisation d'échange d'énergie au sein du système.

Pour réaliser la modélisation Bond Graph, on a donc tout d'abord analysé le système.

Le débit d'entrée  $Q_e$  est le débit de la pompe, c'est le débit qui entre dans le réservoir 1. On suppose que le débit qui sort du réservoir 1 par la vanne 1 est  $V's_1$ , par la vanne b est  $V'b$ . Le débit qui entre dans le réservoir 2 est  $V'c_2$ .

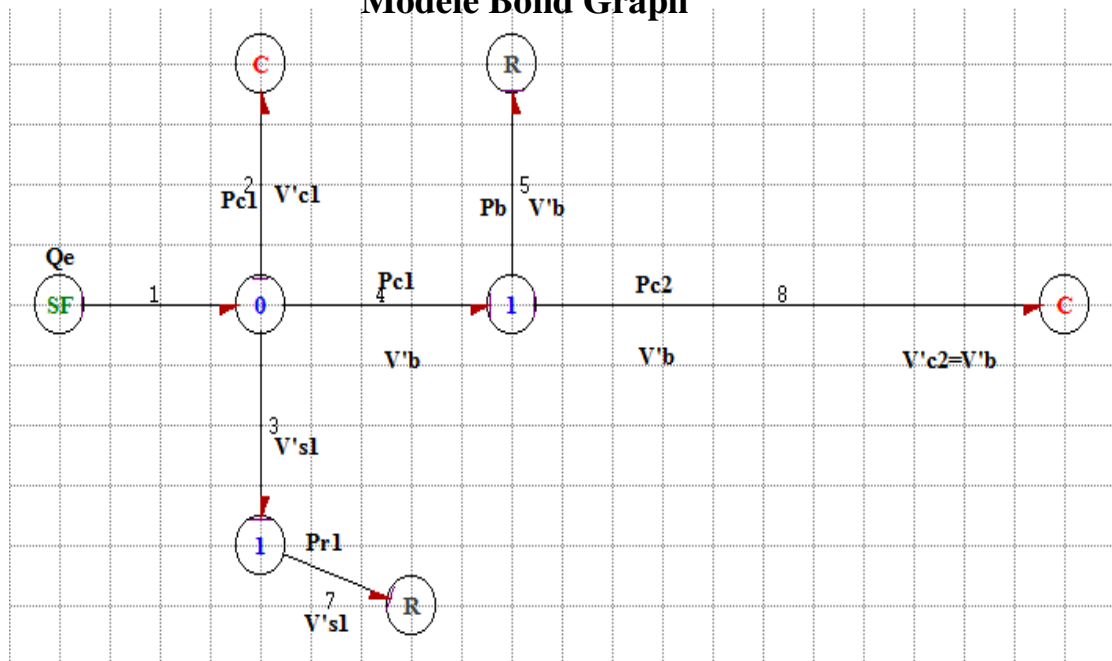


Etant donné que la vanne 2 est fermée, il n'y a pas de débit qui sort du réservoir 2, donc  $V's_2=0$  et  $V'c_2=V'b$ . ( $V'b=V'3$ )

Rappelons que dans un système physique, il y a toujours trois types de phénomènes : résistives (résistance hydraulique), d'inertie (masse en mouvement) et capacitifs.

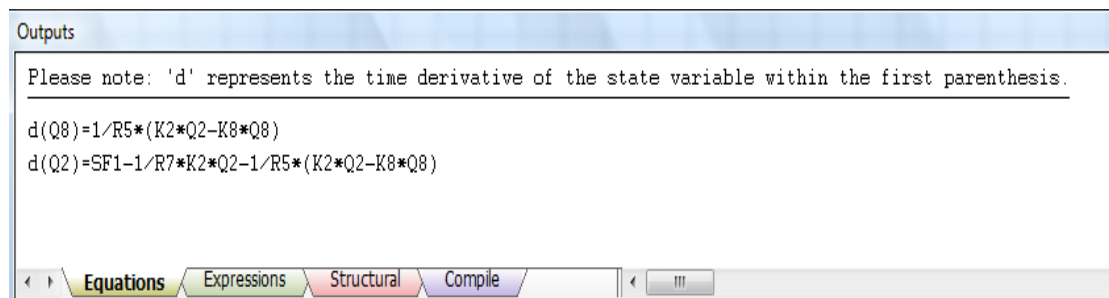
Nous réalisons donc notre schéma Bond Graph sur le logiciel Symbols :

### Modèle Bond Graph



R et C représentent les résistances hydrauliques et les capacités.

A partir du schéma de Bond Graph, on obtient les équations d'état du modèle :



```
Outputs
Please note: 'd' represents the time derivative of the state variable within the first parenthesis.
d(Q8)=1/R5*(K2*Q2-K8*Q8)
d(Q2)=SF1-1/R7*K2*Q2-1/R5*(K2*Q2-K8*Q8)
```

Le schéma Bond Graph nous permet également de déterminer plus simplement les équations correspondant au modèle du système :

$$\begin{aligned} V'c1 &= Qe - (V'b + V's1), \\ V'c2 &= V'b \\ Pr1 &= R1 * V's1, \\ Pc1 &= (1/C1) * (\int Qc1 dt), \quad Pc2 = (1/C2) * (\int Qc2 dt) \\ Pc1 - Pc2 &= Rb * V'b \end{aligned}$$

Ayant trouvé les équations régissant le modèle, on a alors déterminé la valeur de certains paramètres. On a donc tout d'abord déterminé l'aire des réservoirs 1 et 2 :  $A1 = 0.007625 \text{ m}^2$ ,  $A2 = 0.0079 \text{ m}^2$

On a ensuite consulté une documentation pour trouver la valeur des résistances hydrauliques  $R1$  et  $R2$  et des capacités  $C1$  et  $C2$ .  
D'après la documentation :  $\beta1 = 2.25 * 10^{-4}$ ,  $\beta2 = 2.09 * 10^{-4}$

On utilise alors les relations suivantes :

$$\begin{aligned} 1/R1 &= \beta1 / \rho g, \quad 1/R2 = \beta2 / \rho g, \\ C1 &= A1 / \rho g, \quad C2 = A2 / \rho g \end{aligned}$$

Après le calcul, on trouve les valeurs suivantes:

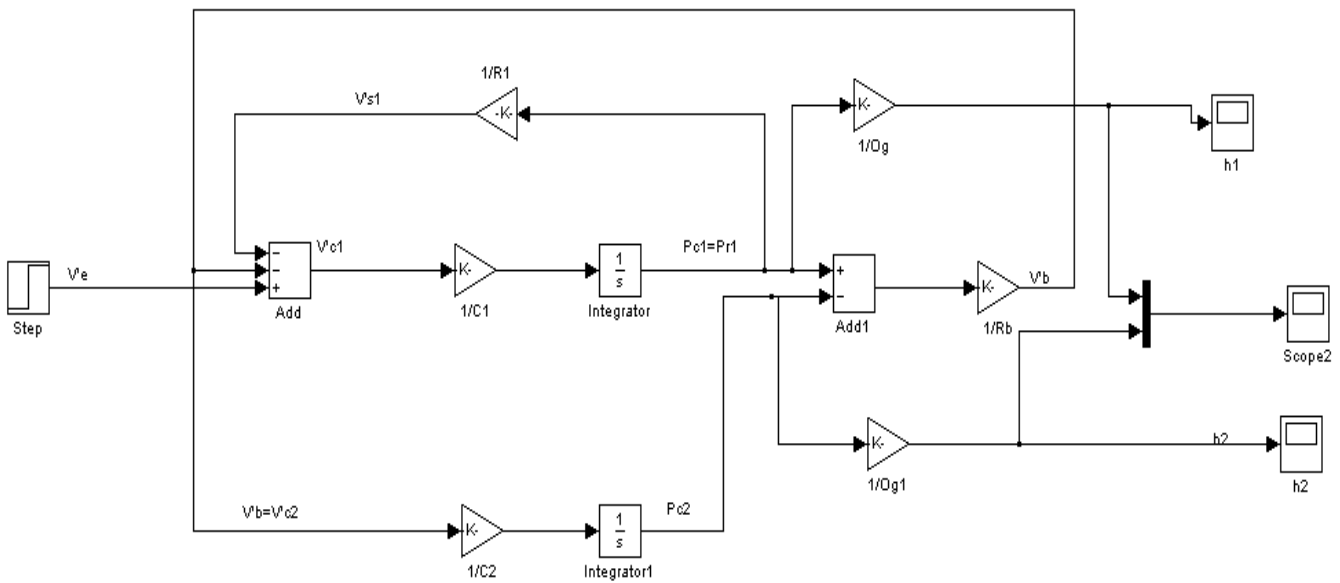
$$\begin{aligned} 1/R1 &= 2.25 * 10^{-8} \\ 1/R2 &= 2.09 * 10^{-8} \end{aligned}$$

....

Avec toutes les équations et les valeurs des paramètres, on a pu réaliser le modèle sur Simulink. Dans ce modèle, le signal d'entrée du système est le débit  $Qe$  et il y a 2 sorties qui sont les hauteurs  $h1$  et  $h2$  du niveau d'eau des réservoirs 1 et 2.

Voici notre modèle théorique sur Simulink :

### Modèle théorique sur Simulink



Avant de pouvoir simuler ce modèle, il nous faut configurer certains paramètres. Pour cela, il faut aller dans le menu de Simulink **simulation** -> **Configuration Parameters**.

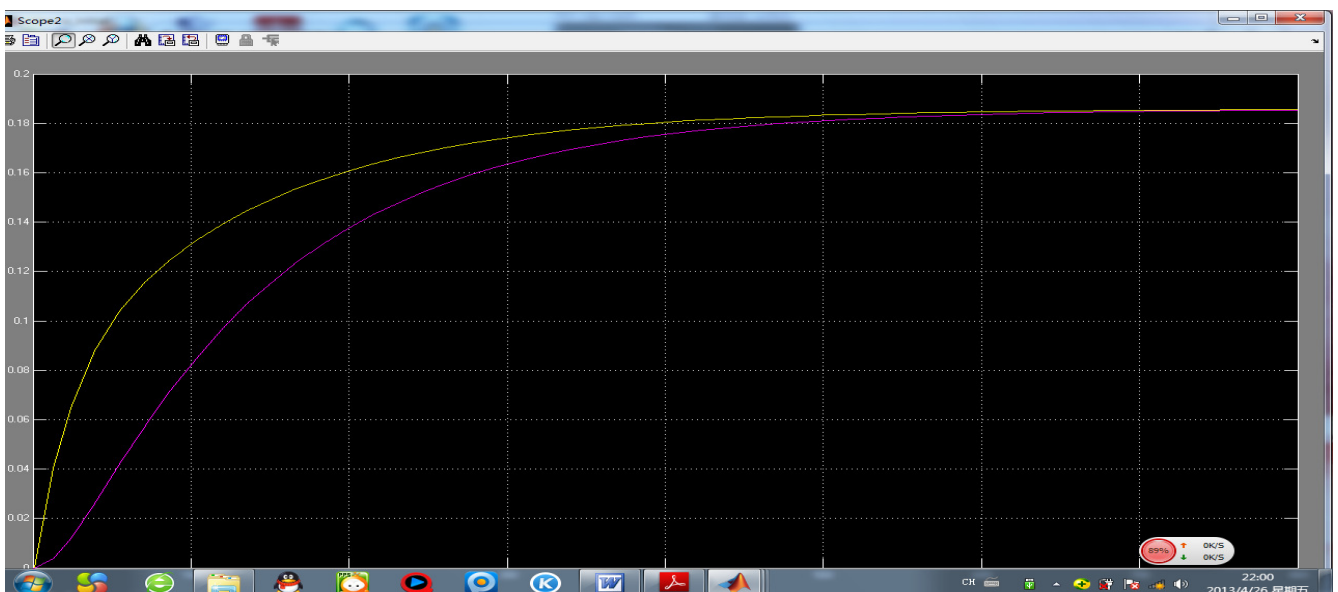
Pour ce projet, on a configuré de cette façon:

Type: Fixed-Step / Solver: discret / Stop time:3000 /Fixed-step size :0.01 / Tasking mode for periodic sample time: auto.

On peut alors tester ce modèle. Si on prend un débit d'entrée  $Q_e=4.06 \cdot 10^{-5} \text{ m}^3/\text{s}$ , on constate que  $h_1$  et  $h_2$  tendent vers une hauteur qui vaut 18 cm.

Voici les courbes de  $h_1$  et  $h_2$  qu'on obtient avec notre modèle théorique :

### Courbes $h_1$ et $h_2$ pour un débit d'entrée $Q_e=4,06 \cdot 10^{-5} \text{ m}^3/\text{s}$

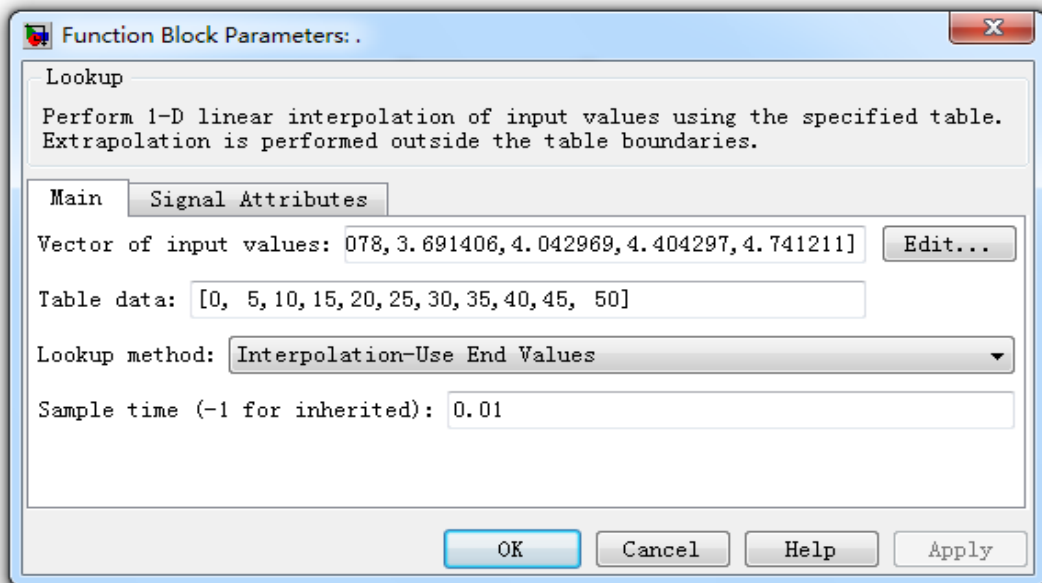




## III-Réalisation du modèle réel

Afin de pouvoir faire le contrôle du système réel, il faut récupérer les données des capteurs. On récupère les données sous forme de tension et on doit faire correspondre ces tensions à une hauteur. Pour convertir une tension en une hauteur, il faut utiliser un bloc d'étalonnage sous Matlab. Il nous sert à avoir des données transmises par les capteurs sous forme de tension et de les convertir.

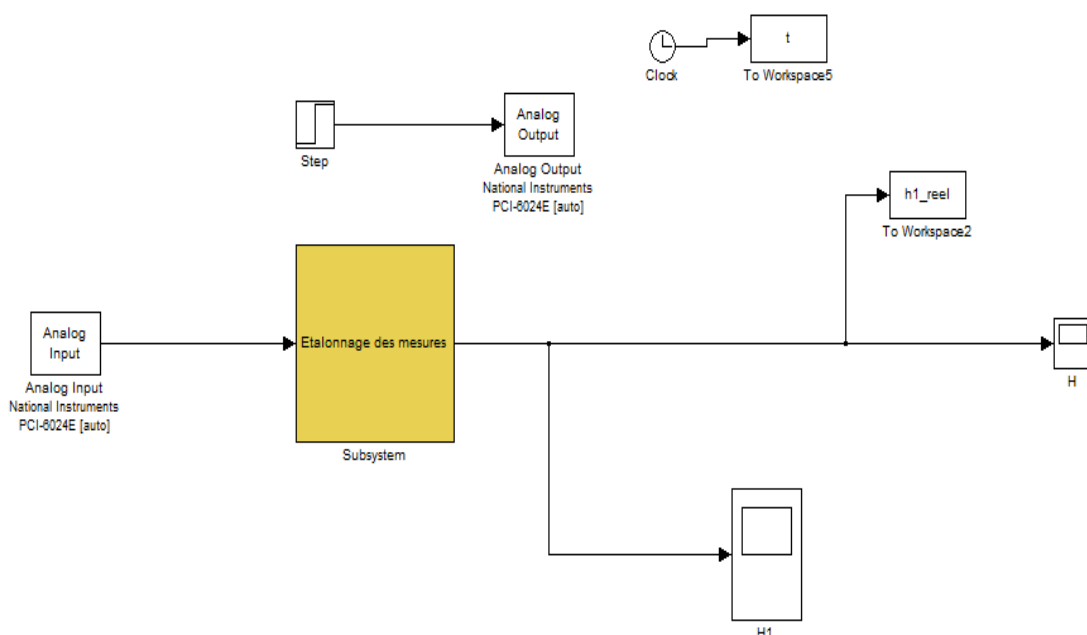
Comme le bloc d'étalonnage ne fait pas partie des objectifs de notre projet, nous avons utilisé un bloc d'étalonnage qui avait déjà été fait.



Le modèle réel sur Matlab est donc composé d'un bloc d'étalonnage, d'un signal d'entrée (analog output) et d'un capteur (analog input).

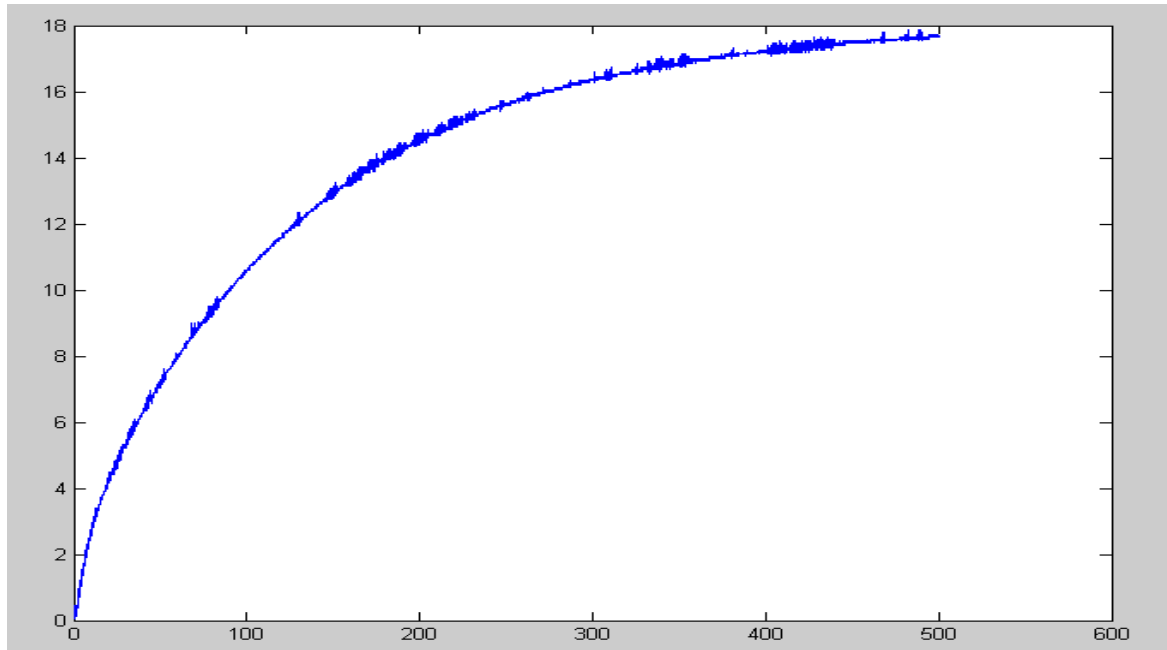
Voici le modèle :

### Modèle réel sur Simulink



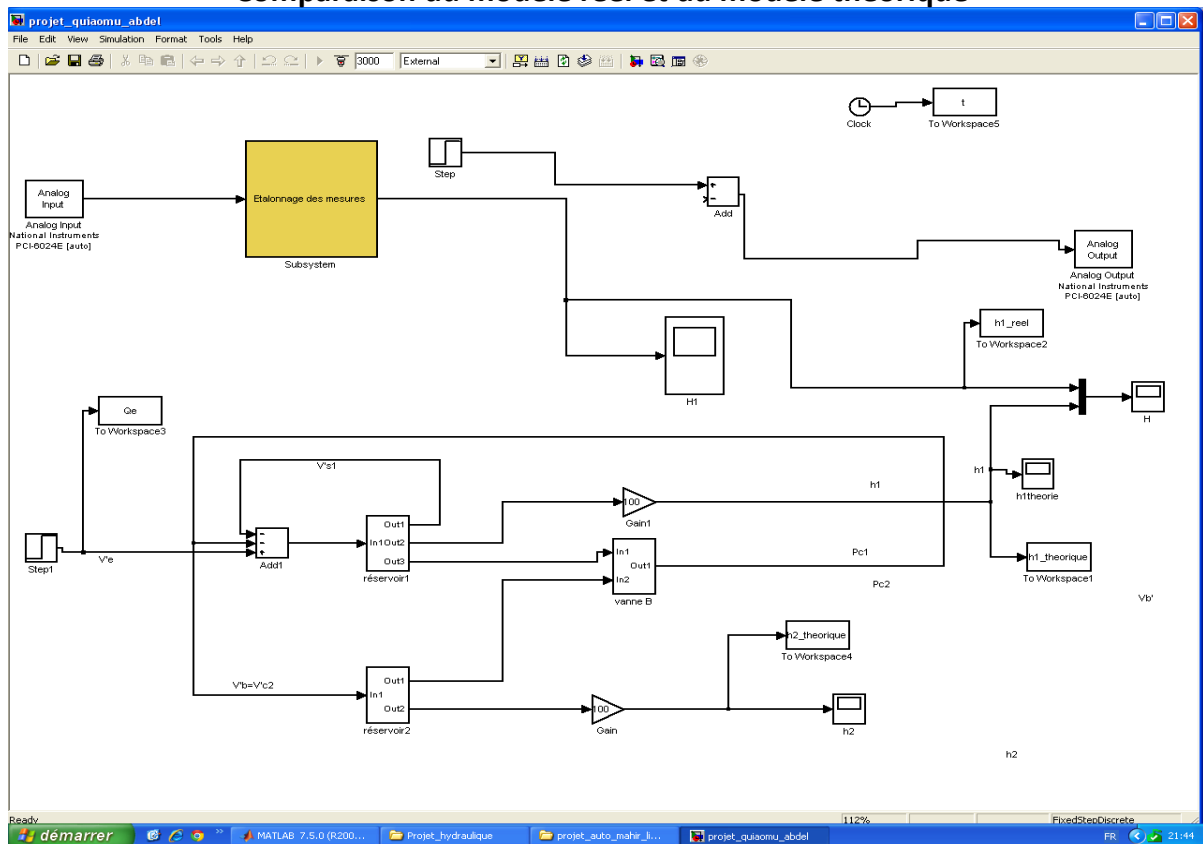
Pour un signal d'entrée de 5Volts (ce qui correspond à un débit  $Q_e=4.06 \cdot 10^{-5} \text{ m}^3/\text{s}$ ), nous observons la courbe h1 réel grâce au capteur 1:

**Evolution de h1\_réel (cm) pour un débit d'entrée  $Q_e=4,06 \cdot 10^{-5} \text{ m}^3/\text{s}$**



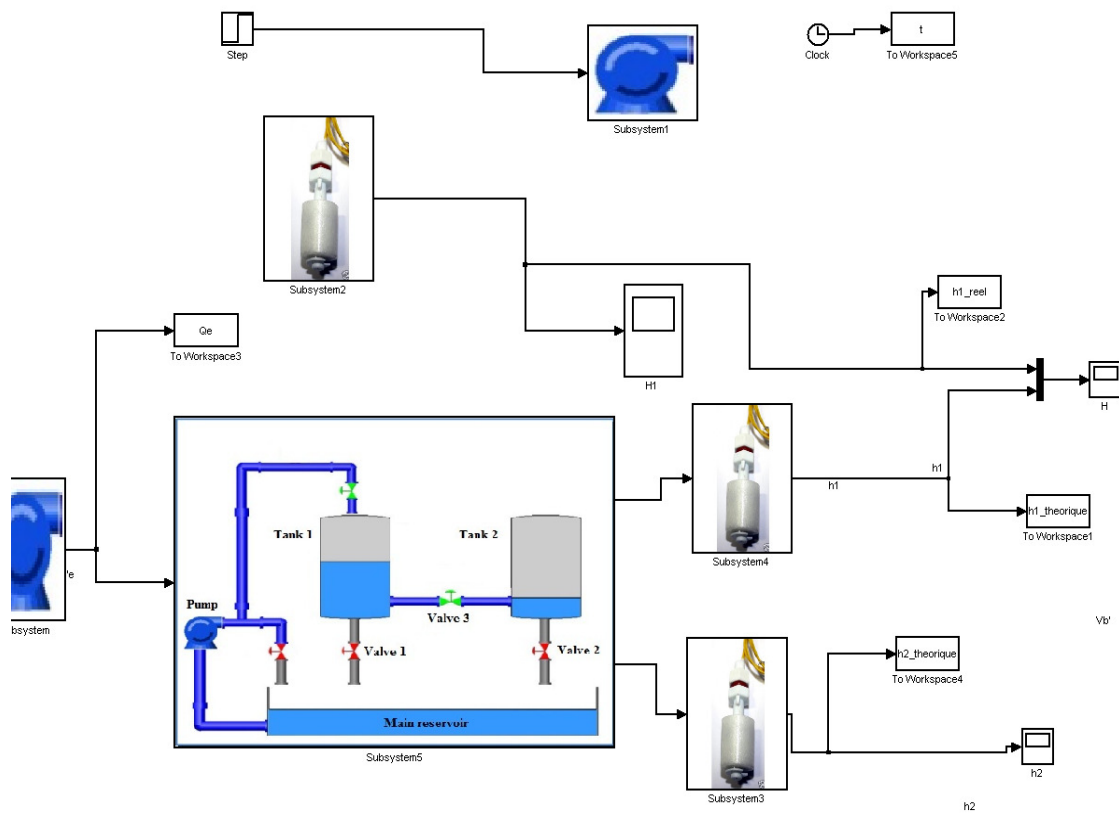
On compare alors le modèle réel au modèle théorique pour un même débit d'entrée  $Q_e=4.06 \cdot 10^{-5} \text{ m}^3/\text{s}$ :

**Comparaison du modèle réel et du modèle théorique**

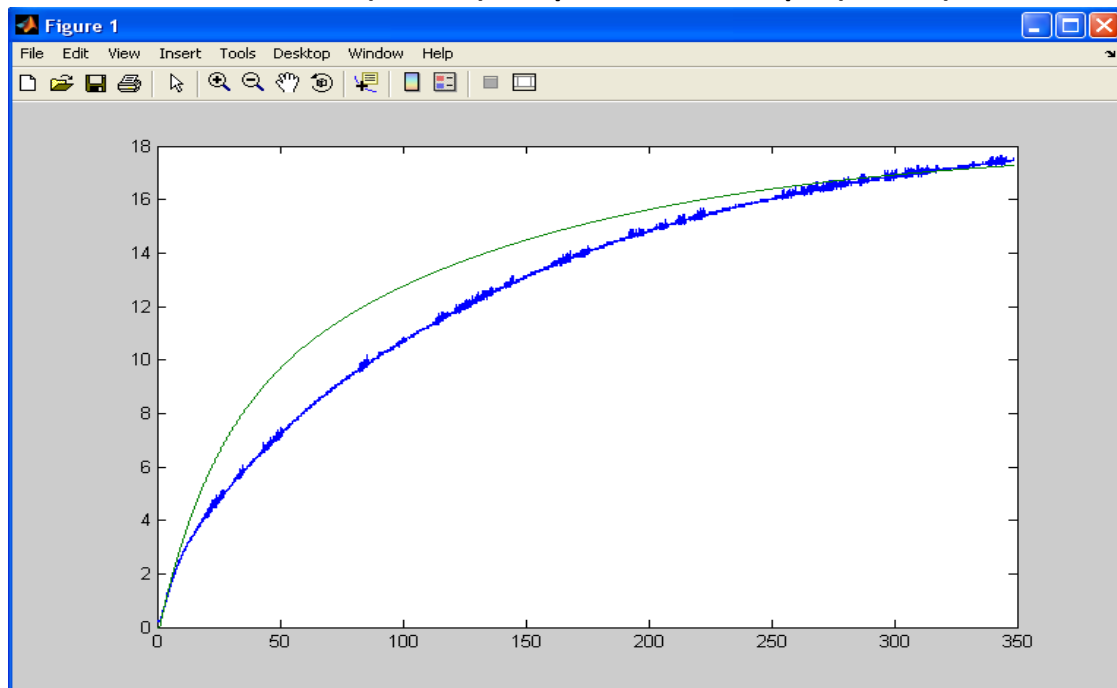


Pour que le modèle soit compris facilement, nous avons modifié l'interface avec des images représentant la pompe et les capteurs de niveau:

### Comparaison des deux modèles avec interface modifiée



**Courbe h1 réel (en bleu) comparé à h1 théorique (en vert)**



Les allures des deux courbes sont assez semblables mais la courbe théorique suit une croissance exponentielle plus rapide.

## IV-Etude de la boîte à outil 'ident'

Ident est une boîte à outil temps réel de Matlab RTW. Cet outil permet de déterminer le modèle mathématique d'un système sur la base des observations expérimentales entrées-sorties. Ainsi, en utilisant la boîte à outil 'ident' on va pouvoir simplifier notre modèle en utilisant une seule fonction de transfert.

La base de l'identification est le traitement de données (data) en vue d'obtenir un modèle. L'interface graphique de la boîte à outil est composée de deux parties (data et models).

Ainsi on exécute notre modèle réel en envoyant chaque seconde la valeur de la hauteur h1 atteinte et la valeur du débit d'entrée sur Workspace. Cela permet de sauvegarder les valeurs de Qe et h1-réel au cours du temps.

On peut alors estimer les paramètres du modèle dans le menu (Estimate->Parametric Models).

Ici, on choisit d'obtenir une fonction de transfert d'ordre 2. Afin d'obtenir la fonction de transfert directement, on fait glisser ce modèle vers 'To Work Space'.

### Boîte à outil 'ident'

The screenshot displays the 'System Identification Tool' interface. The main workspace shows a 'Dryer' model being processed. The 'Data/model Info: P2' window on the right provides the following information:

Model name: P2  
Color: [0,0.75,0.75]

Process model with transfer function  
K  
 $G(s) = \frac{K}{(1+Tp1*s)(1+Tp2*s)}$   
with K = 4.4141e+005  
Tp1 = 1064  
Tp2 = 55.208  
Estimated using PEM from data set Dryer  
Loss function 0.00389709 and FPE 0.00389736

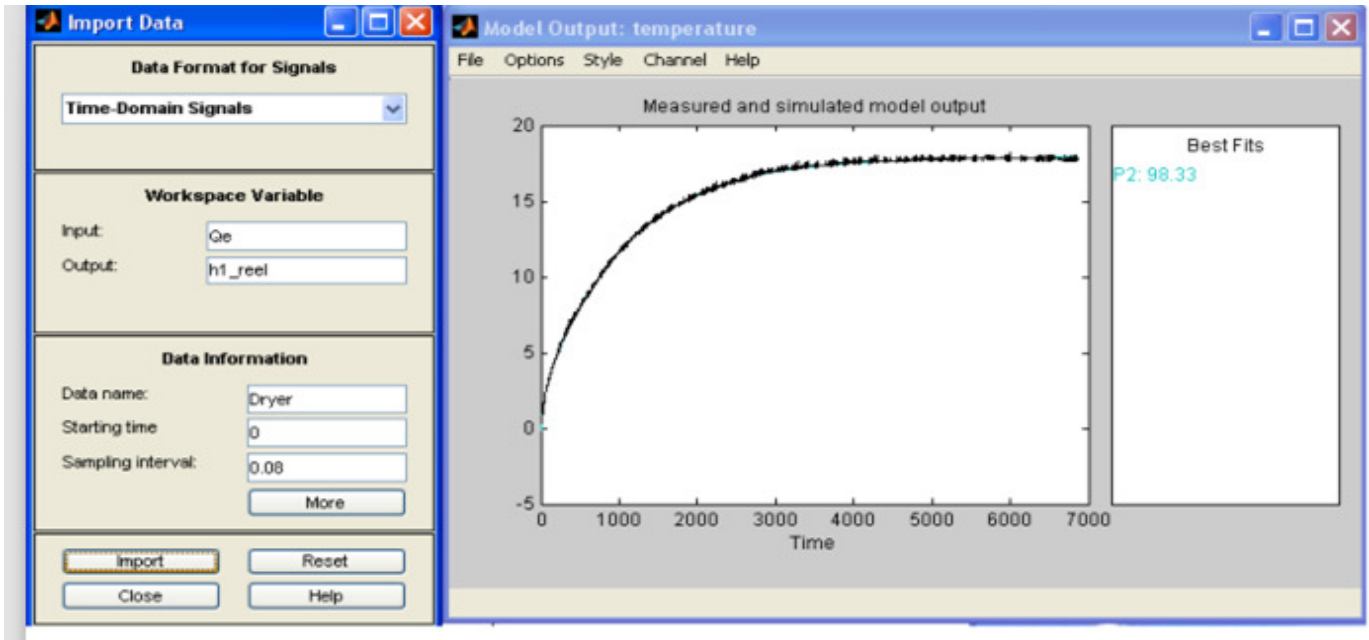
Diary And Notes  
% This is the 'Hair Dryer' data set. The  
% input is the electric power and the  
% output is the outlet air temperature.  
load dryer2  
% import Dryer  
P2 = pem(Dryer,P2);

On obtient directement la fonction de transfert :

$$G(s) = \frac{K}{(1+Tp1*s)(1+Tp2*s)} \quad \text{Avec } K=4.4141 \cdot 10^5, \quad Tp1=1064, \quad Tp2=55.608$$

On voit qu'avec cette fonction de transfert, la réponse est très précise (réponse h1 en bleu ciel ci-dessous), et en plus, on peut obtenir un modèle beaucoup plus simple. On garde la même valeur pour la consigne  $Q_e$  ( $Q_e=4,05 \cdot 10^{-5} \text{ m}^3/\text{s}$ ).

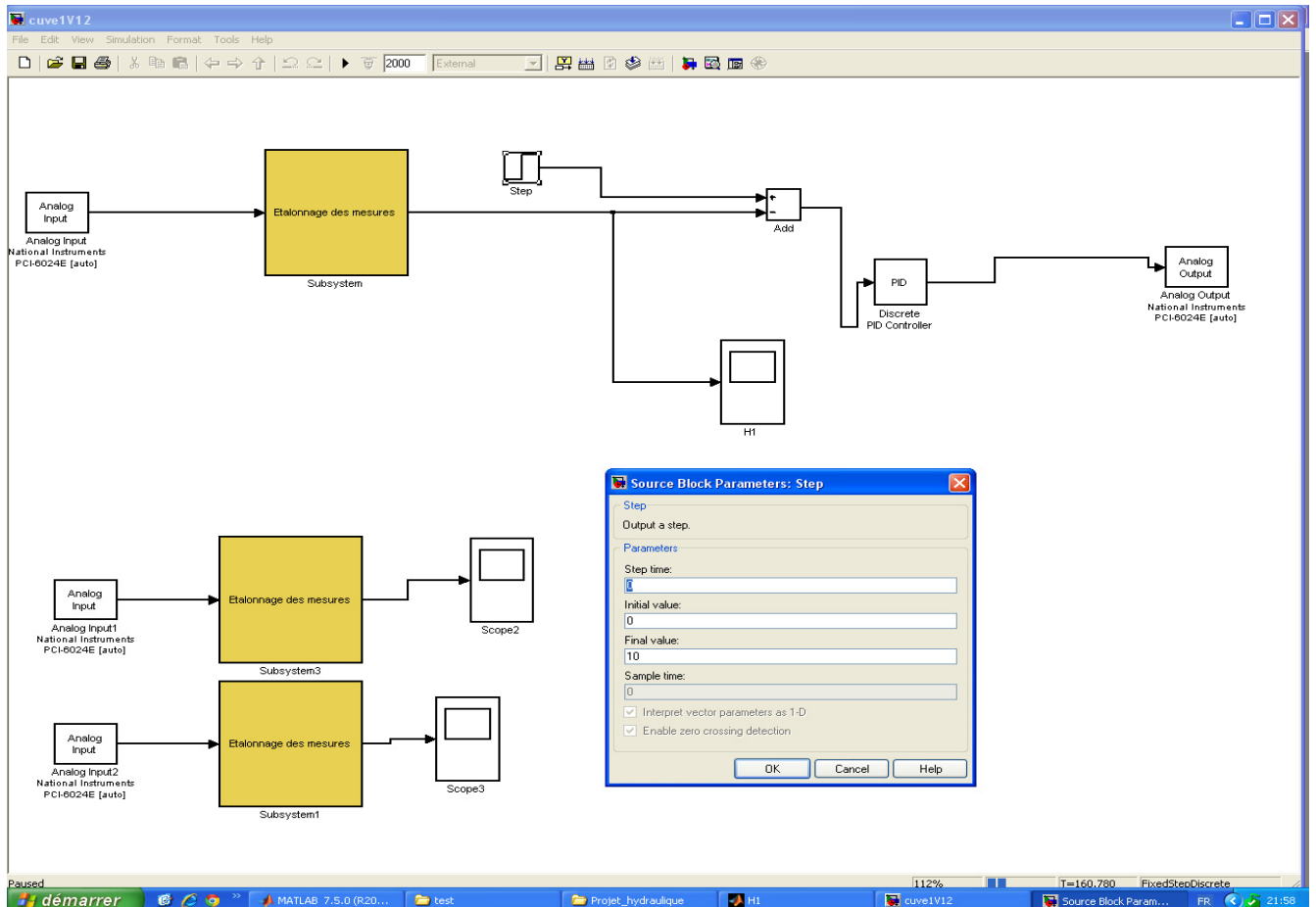
### Visualisation de la courbe h1 obtenue avec l'outil 'ident' (courbe en bleu ciel)



# V-Contrôle du niveau d'eau avec PID

Au départ, on étudiait le système pour un débit d'entrée  $Q_e$  et on atteignait une certaine valeur  $h_1$  et  $h_2$ . Maintenant on veut contrôler le niveau d'eau des réservoirs. C'est-à-dire que la consigne (signal d'entrée) est désormais une hauteur (en centimètre). Pour cela on ajoute un correcteur PID afin de corriger l'erreur ce qui va permettre d'atteindre la consigne.

Schéma Simulink du modèle réel avec correcteur PID



Pour déterminer les paramètres du PID, nous avons choisi la méthode de Ziegler-Nichols, car cette méthode de réglage est la plus utilisée en théorie en boucle ouverte et boucle fermée.

La méthode utilisée est de régler un correcteur à partir d'un essai en limite de pompage ('oscillations permanentes'). Pour obtenir la limite de pompage, on place un correcteur proportionnel dans la boucle fermée et on augmente doucement le gain de ce correcteur jusqu'à obtenir des oscillations auto-entretenues (phénomène de pompage). On note le gain  $K_0$  qui a amené le système en limite de stabilité et la période  $T_0$  des oscillations obtenues. (On trouve  $K_0=1,046$  et  $T_0=28,8s$ ).

Les paramètres du système pour que la réponse du système bouclé soit satisfaisante sont donnés par le tableau suivant :

	Kp	Ti	Td
Correcteur P	0.5*Ko	-	-
Correcteur PI	0.45*Ko	0.83*To	-
Correcteur PID	0.6*Ko	0.5*To	0.125*To

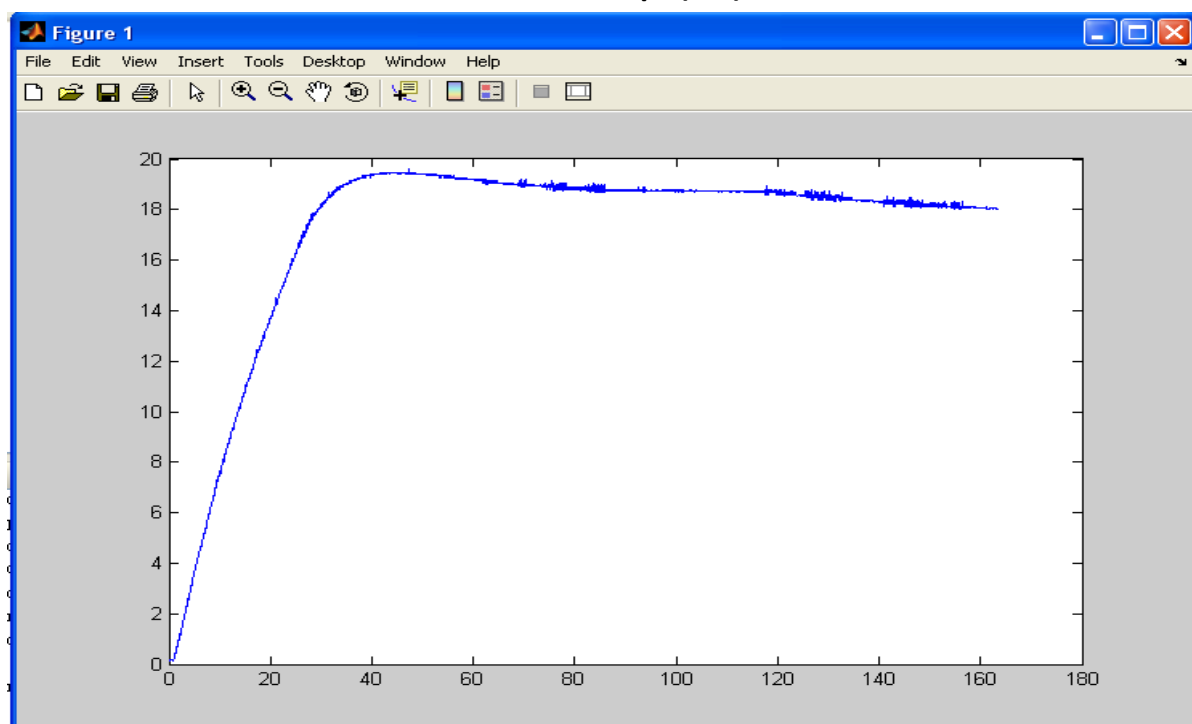
Avec  $C(p) = K_p(1 + 1/(T_i * p) + T_d * p)$

Dans notre système, nous avons utilisé un correcteur proportionnel intégral (PI).

Donc nous avons trouvé  $K_p = 0.86$  ;  $K_i = 0.036$

Pour contrôler le niveau d'eau, nous avons testé pour une consigne  $h_1 = 18\text{cm}$  et nous observons la réponse suivante :

**Evolution de la hauteur du niveau d'eau du réservoir 1 (en cm)  
en fonction du temps (sec)**



On remarque qu'il y a un petit dépassement, mais ce dépassement est acceptable. Le correcteur PI est donc satisfaisant, et donc le signal de sortie aussi (erreur à peu près nulle, temps de réponse rapide, système stable).





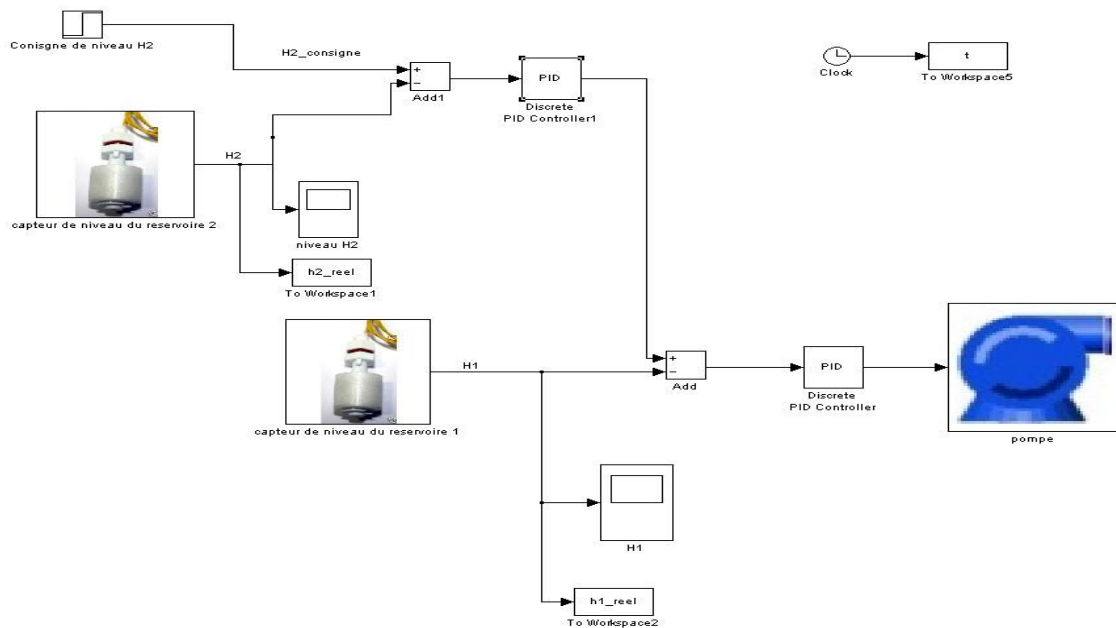
On utilise donc deux correcteurs pour un tel système, un correcteur maître et un correcteur esclave. Nous avons choisit le correcteur esclave de type proportionnel, car il agit faiblement sur la régulation du réservoir 2, et le correcteur maître de type proportionnel intégral, car c'est lui qui fait la plus grosse partie de la régulation.

Pour trouver la valeur des correcteurs PID, on se base à peu près sur la valeur du correcteur PI trouvé précédemment puis on fait varier la valeur du correcteur proportionnel.

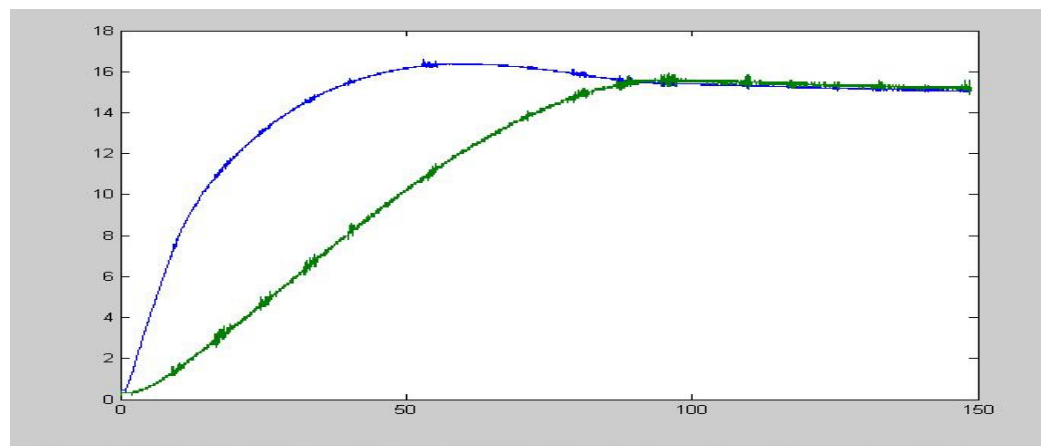
Correcteur maître :  $K_p=0.87$ ,  $K_i=0.036$

Correcteur esclave :  $K_p=1$

Nous avons par la suite simplifié le schéma précédent sur Simulink, pour avoir une interface plus compréhensible par tout le monde, avec des images:



Avec une consigne  $h_2=15\text{cm}$ , nous observons l'allure des courbes  $h_1$ (en bleu) et  $h_2$ (en vert) : **allure des courbes  $h_1$  et  $h_2$  (cm) en fonction du temps (sec)**



On remarque que  $h_1$  augmente plus vite que  $h_2$ , et les régulateurs PID contrôlent bien le système, le niveau d'eau atteint bien la valeur de la consigne.

→ cf annexe autre méthode pour déterminer les paramètres des PID.

## VII-Commande par retour d'état

La commande par retour d'état permet d'imposer une dynamique au système en faisant un retour d'état du système vers la commande. C'est un moyen de modifier le comportement en boucle fermée par une représentation d'état.

Le principe est de déterminer une commande telle que les pôles du système de la fonction de transfert du système bouclé soient convenablement placés dans le plan complexe et satisfassent des spécifications d'amortissement, de rapidité...

Les pôles de la fonction de transfert étant les valeurs propres de la matrice d'état, le but est de réaliser un asservissement modifiant convenablement la matrice d'état du système.

A partir des équations du modèle du système hydraulique on trouve les équations d'états. Voici les équations du modèle :

$$\begin{aligned}A1*(dh1/dt) &= Qe - (V's1 + V'b) ; & A2*(dh2/dt) &= V'b \\ Qe &= \beta1*h1 ; & V'b &= \beta3*(h1-h2) \\ \text{Avec } A1 &= 0.007625 \text{ m}^2 ; & A2 &= 0.0079 \text{ m}^2 \\ \beta1 &= \rho g / R1 = 2.25 * 10^{-4} ; & \beta2 &= \rho g / R2 = 1.6 * 10^{-4}.\end{aligned}$$

On pose  $x1=h1$ ,  $x2=h2$  où  $x1$  et  $x2$  sont les deux variables d'état

Donc on a les équations:

$$\begin{aligned}[dx/dt] &= A*[x] + B*[u(t)] \\ [y] &= C*[x] + D*[u(t)]\end{aligned}$$

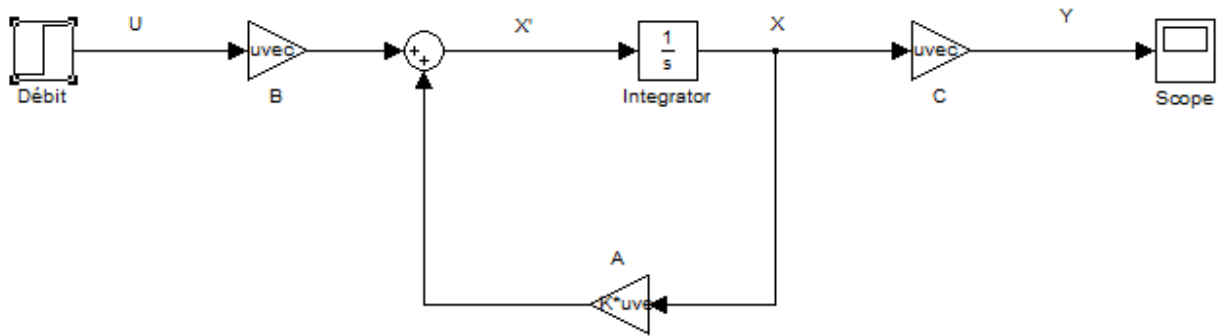
avec la matrice

$$A = \begin{pmatrix} -(\beta1 + \beta3)/A1 & \beta3/A1 \\ \beta3/A2 & -\beta3/A2 \end{pmatrix}$$

$$B = \begin{pmatrix} 1/A1 \\ 0 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad D = 0$$

On réalise le modèle en Boucle Ouverte sur Simulink:

### Schéma sur Simulink du modèle en Boucle Ouverte



Pour faire le modèle du retour d'état, il nous faut la matrice des pôles  $M$ , et la matrice du gain  $K$ .

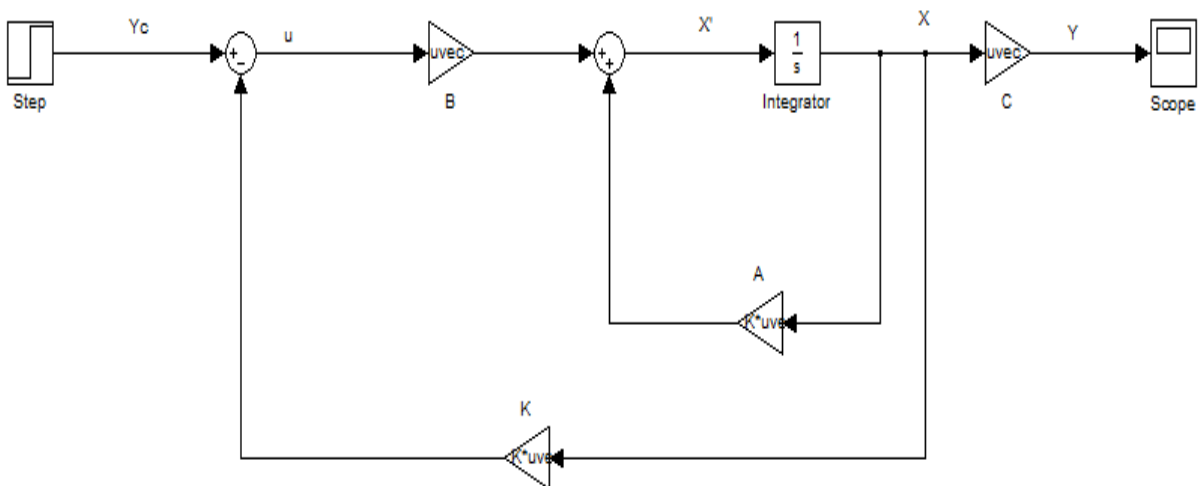
Pour cela, on écrit un programme sur Matlab( voir annexe ) permettant de calculer  $M$  et  $K$ . Avec ce programme, on remarque que notre modèle est commandable et observable.

On trouve également les valeurs la matrice  $K$  grâce à la fonction Matlab `acker( A, B, M)`.

$$K = [-0.2041 \cdot 10^{-19} \quad 0] \quad \text{pour la matrice } M = \begin{pmatrix} -0.0609 \\ -0.0098 \end{pmatrix}$$

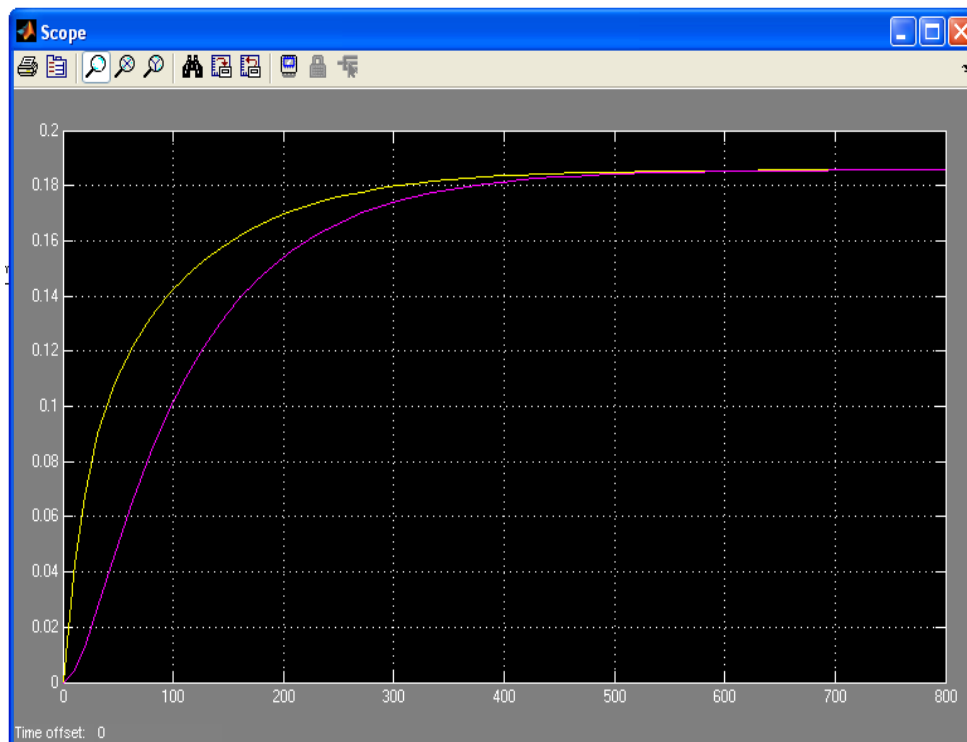
Nous pouvons alors tester la simulation sur Simulink de la commande par retour d'état en posant  $U = Y_c - K \cdot X$ :

### Schéma sur Simulink de la commande par retour d'état



Pour un débit d'entrée  $Q_e=4.06 \cdot 10^{-5}$ , voici l'allure des signaux de sortie ( $h_1$  et  $h_2$ ) que l'on obtient :

**Evolution du niveau d'eau des deux réservoirs (en mètres)  
en fonction du temps (secondes)**



On remarque que  $h_1$  (en jaune) et  $h_2$  (en violet) atteignent bien 18 cm rapidement, ça satisfait bien notre consigne.  
Donc la commande par retour d'état est bien efficace car on impose la dynamique que l'on veut au système.

# Conclusion

Ce projet aura été à la fois enrichissant sur le plan des méthodes de travail que sur la compréhension des outils et logiciels utilisés. Lors de l'étude de ce système hydraulique nous avons ainsi acquis de nouvelles connaissances scientifiques telles que les méthodes d'identification des paramètres d'un modèle et l'implémentation en temps réel d'un ensemble de commandes simples et avancées.

Ce projet nous a surtout permis de nous rendre compte qu'il est nécessaire de bien s'organiser et de bien se répartir les tâches. Il a été nécessaire également de savoir résoudre les problèmes auxquels on a été confronté tels que la compréhension du système d'acquisition, des paramètres à utiliser pour avoir des résultats compréhensibles, ou encore l'utilisation de l'outil 'ident'.

Enfin, grâce aux divers objectifs qui nous ont été attribués, nous avons pu mettre en pratique nos connaissances. Et dans cette phase pratique, nous avons ainsi réalisé les objectifs demandés. Ce projet nous a ainsi permis d'être confrontés à un cas réel d'étude et a donc enrichi notre parcours.

# Annexes

- Programme sur Matlab pour le retour d'état

```
%syms B1 B3 A1 A2
B1=2.25*10^-4;k=15;B3=1.6*10^-4; A1=0.007625; A2=0.0079;
A=[-(B1+B3)/A1 B3/A1;B3/A2 -B3/A2]
B=[1/A1;0]
C=[1 0;0 1]
D=0;
%syms k1 k2 k3
%K=[k1 k2 k3];

%calcul du rang de la matrice de commandabilité
disp('analyse commandabilité')
SYS=ss(A,B,C,D);
rang=rank([B A*B]);
rang

%calcul du rang de la matrice d'observabilité
disp('analyse de l'observabilité')
OBS=[C C*A];
Rang=rank(OBS);
Rang

poles_BO=eig(A);
poles_BO

M=poles_BO

%on trouve pole_BO1=-0.0609 et pole_BO2=-0.0098
%M=[-0.0609;-0.0098];
K=acker(A,B,M);
K
```

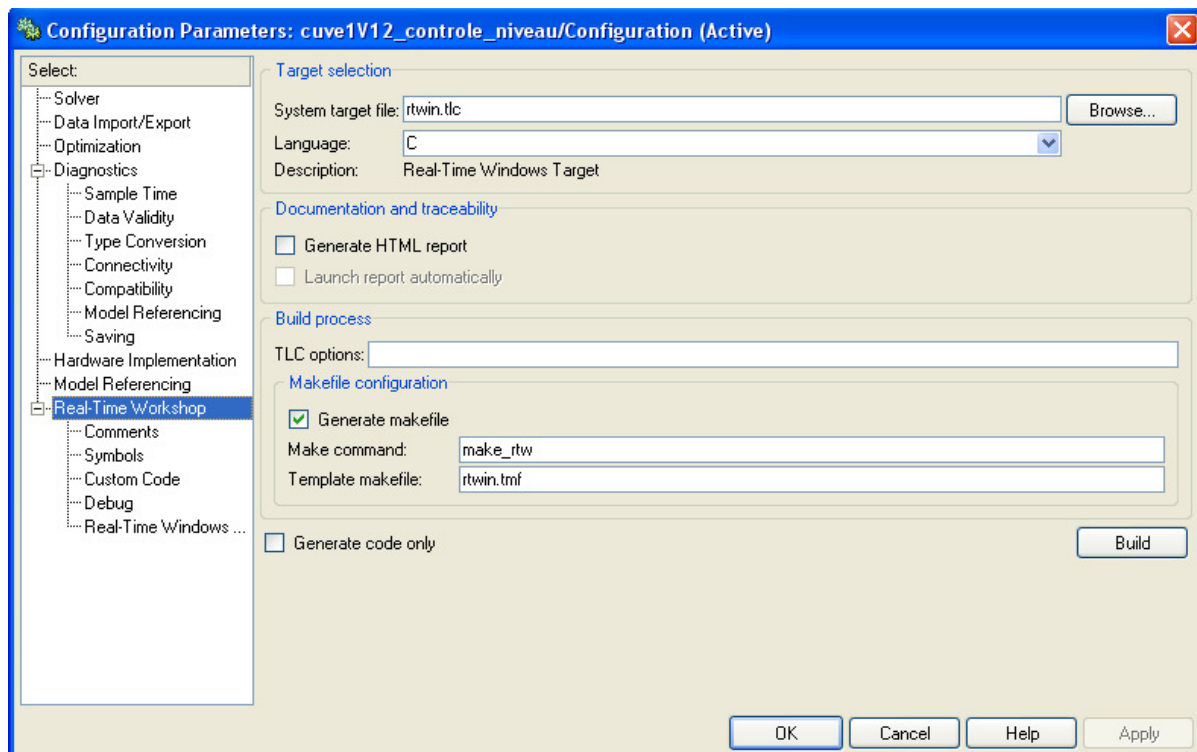
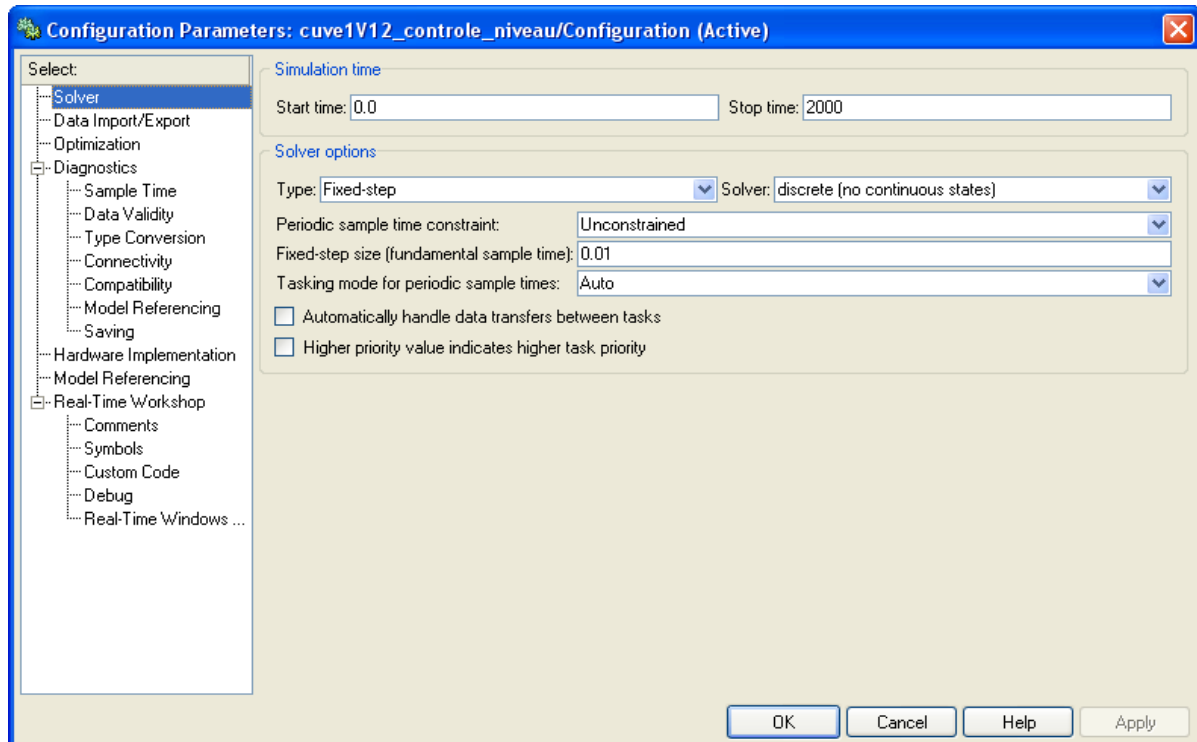
- Vidéo pour l' utilisation de l'outil 'ident' de Matlab

[www.youtube.com/watch?v=t6SPSRacD\\_0](http://www.youtube.com/watch?v=t6SPSRacD_0)

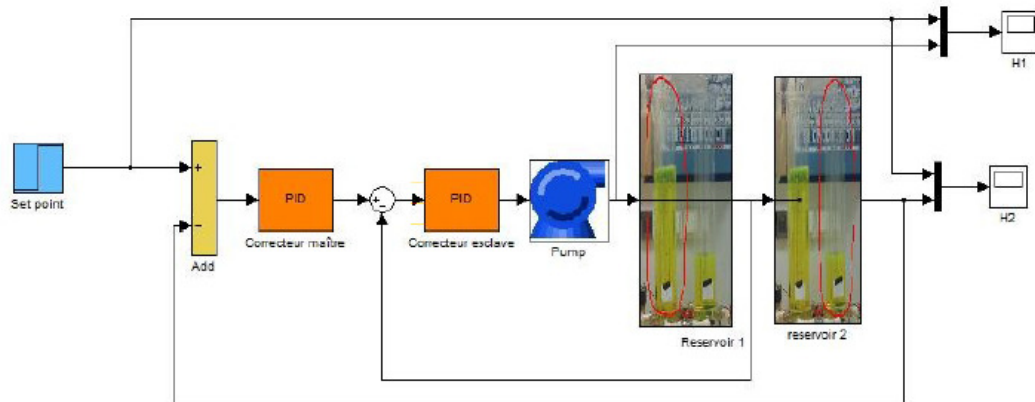
- Document de M.BOUAMAMA :

Méthode des moindres carrés : étude de la boîte à outil 'ident'

- Configuration des paramètres du panneau de commande



- Autre méthode de calcul pour déterminer les paramètres des gains du PID pour le schéma cascade :



La fonction de transfert du réservoir 1 est déterminée expérimentalement grâce à la boîte à outil 'ident':  $H1(p)=h1(p)/Qe(p)=K/(1+\tau*p)$

On peut trouver la valeur de K et  $\tau$  grâce à la boîte à outil 'ident' de Matlab

La fonction de transfert de la pompe (du moteur à courant continu) est également déterminée, avec une documentation technique :

$$G(p)=\Omega(p)/U(p)=10/(0,3*p^2+0,95*p+1)$$

On décide de travailler avec un correcteur PI pour la boucle interne, ie  $C(p)=Kp*(1+T*p)/(T*p)$  avec  $T=Kp*Ti$

$$C(p) \text{ peut également s'écrire } C(p)=Kp+(1/Ti)*p$$

En étudiant la fonction de transfert de la boucle interne en boucle fermée, et en supposant que le gain vaut 1, on pose alors  $T= \tau$  pour simplifier la fonction de transfert. Or  $\tau$  est connue et  $T=Kp*Ti$

On peut alors déterminer les paramètres  $Kp$  et  $Ti$ .

On teste ensuite les paramètres calculés du PID sur le système réel puis on vérifie que ça fonctionne bien en faisant varier le correcteur PID maître de type P ou PI.