

# SOMMAIRE

Remerciement.....	02
Introduction.....	03
I- <u>Présentation du projet</u> .....	04
II- <u>Acquisition des mesures de capteurs</u> .....	06
III- <u>Régulation du système</u> .....	08
1) <u>Correcteur PID</u> .....	08
2) <u>Régulation en cascade</u> .....	10
3) <u>Commande Floue</u> .....	11
Conclusion.....	12
Annexe : .....	13
- Câblage du système	
- Acquisition de l'étalonnage	
- Réalisation de la commande Floue	

# **REMERCIEMENT**

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

Nous voudrions tout d'abord exprimer notre profonde reconnaissance à Monsieur Belkacem OULD BOUAMAMA, nous encadrant et nous ayant suivi avec beaucoup d'attention, qui a dirigé notre travail de façon efficace tout le long de projet.

Nous tenons aussi à remercier toute l'équipe de l'école polytechnique de Lille 1, pour la confiance qu'ils nous ont accordé, leurs soutiens constant, leurs précieux conseils ainsi que leur disponibilité tout au long du projet.

Nos remerciements s'adressent aux membres du Jury qui nous font l'honneur de participer à la soutenance.

# **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 d'un projet d'étude en quatrième année d'ingénieur avec la spécialisation IMA (Informatique, microélectronique et automatique), nous devons réaliser une plate-forme utilisable pour de futurs travaux pratiques au sein de polytech-lille, leur apprenant ainsi à utiliser un système de régulation.

Au départ ce projet devait reprendre le travail réalisé par deux groupes d'IMA de 4ème et 5ème année, or des modifications ont été apportées entre temps, nous relançant à la lecture des capteurs et leurs étalonnages au lieu d'améliorer de façon visuel et pratique le projet.

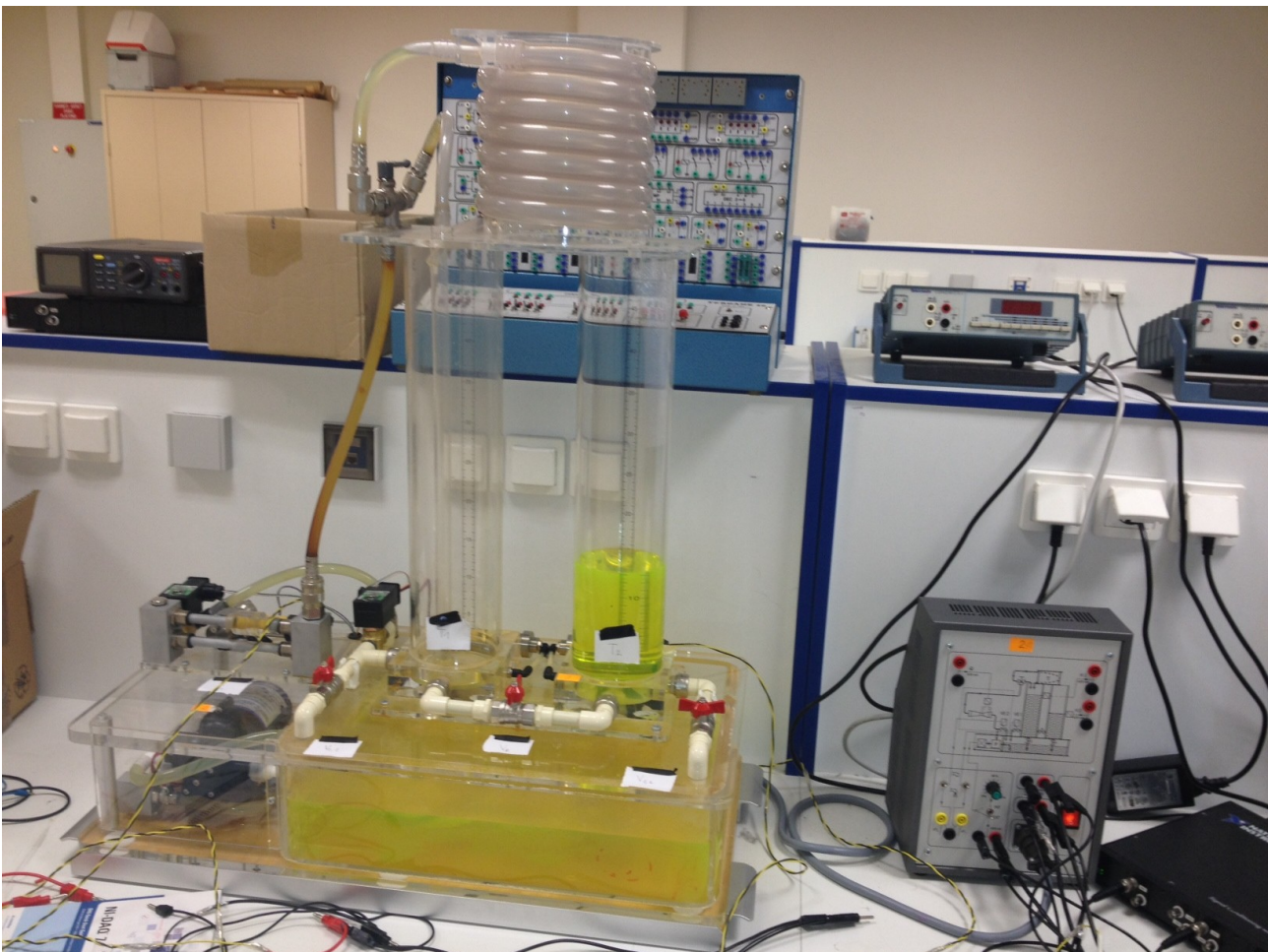
Notre objectif est donc de réussir à récupérer les capteurs et de concevoir un nouveau système de contrôle des deux réservoirs couplés, pour qu'il puisse fonctionner avec n'importe quelle contrainte (une ou plusieurs vannes ouverte, ou des fuites), ainsi qu'un interface assez agréable pour l'œil et qui soit compréhensible par le plus grand nombre de personnes possibles (même des personnes qui n'ont fait de la régulation que très rarement).

## I-Présentation du projet

Le but de notre projet est de réaliser un système de deux réservoirs à l'aide d'une pompe et d'un fluide, un tel système peut aussi être appelé une installation hydraulique, appelé ainsi à cause de sa pompe.

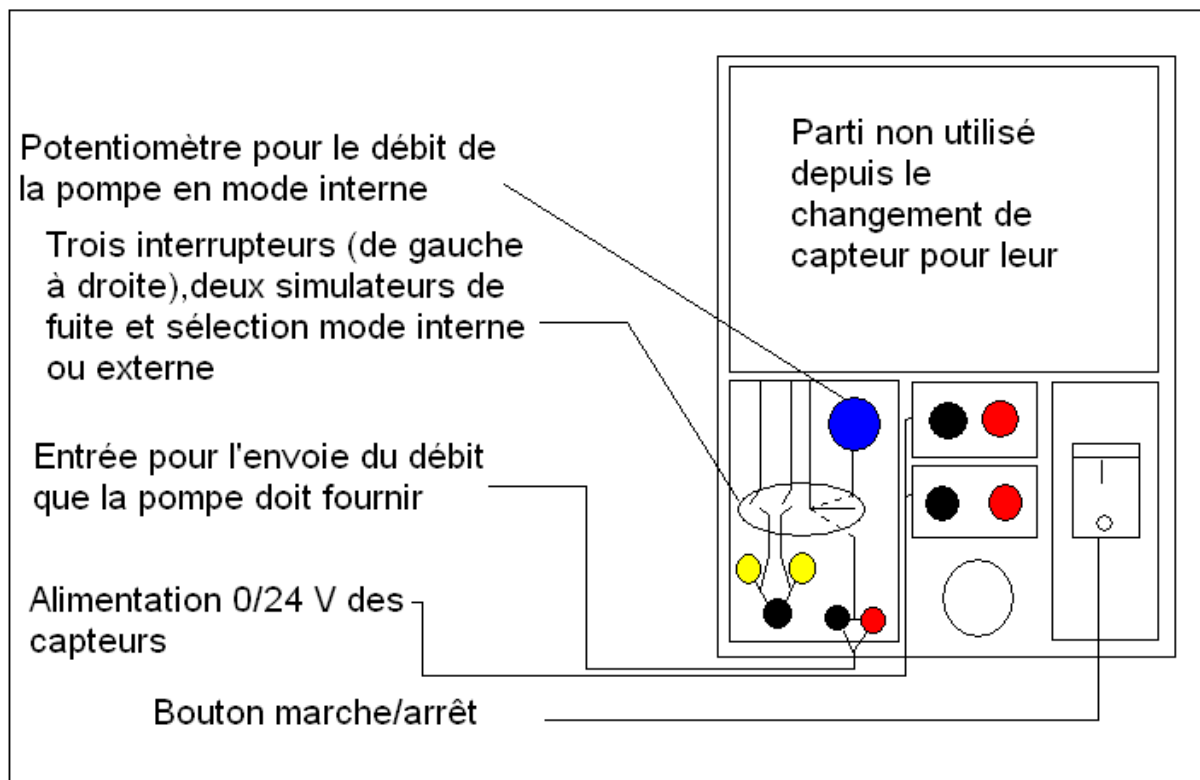
Nous avons à notre disposition ce système hydraulique qui peut être commandé de manière manuelle ou de manière automatique à l'aide d'un logiciel pouvant créer une interface ou encore un schéma bloc servant à asservir le système.

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



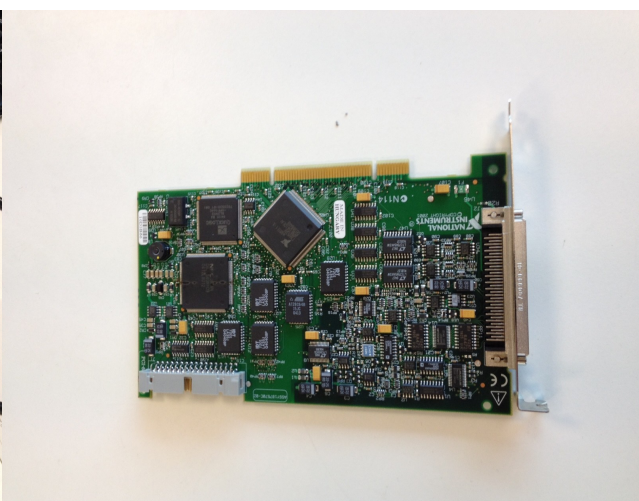
Le boîtier de commande a été modifié l'année dernière et certaines fonctions ne sont donc plus utilisées.

Voici les commandes servant toujours à la manipulation:



La lecture des capteurs se faisait directement sur la parti qui n'est plus utilisé à cause de bruit trop important lors des acquisitions, maintenant les capteurs sont directement lus sur les fils d'alimentation de ceci.

Pour l'acquisition de ces capteurs, un boîtier de connexion SC-2345 TEXAS INSTRUMENT et une carte d'acquisition PCI 6024-E sont déjà implantés.



Le câblage entre le boîtier de commande, le boîtier SC-2345 et la carte d'acquisition PCI 6024-E est expliqué dans l'annexe.

## II-Acquisition des mesures de capteurs

La première étape afin de pouvoir commencer la régulation du système est l'étalonnage des capteurs pour cela, il faut tout d'abord avoir (mais ceci n'est pas une obligation) le logiciel Measurement & Automation qui peut se trouver, gratuitement, sur le site internet de Texas Instrument.

Ce logiciel nous servira à avoir les données transmises par les capteurs sous forme d'une tension, il ne restera plus qu'à convertir cette tension en hauteur souhaitées au niveau des réservoirs (unité de mesure: centimètre).

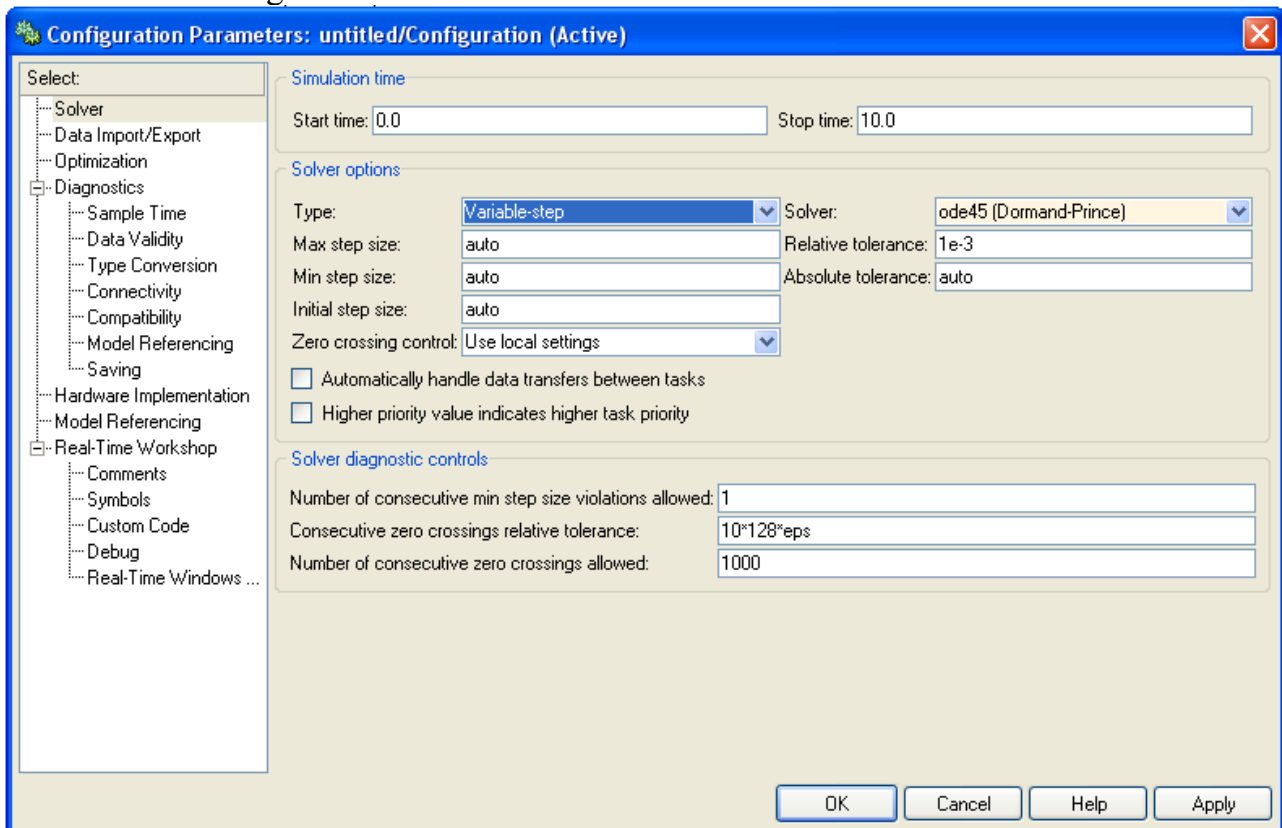
*Vous pouvez retrouver en annexe, toute la suite logique d'événement à effectuer afin d'avoir un étalonnage correcte des deux réservoirs.*

Maintenant que l'étalonnage est acquis et sauvegardé il ne reste plus qu'à l'intégrer à un modèle Matlab/Simulink afin de pouvoir le réguler.

Sous Simulink choisir dans la librairie, l'onglet Real Time Workshop, puis les deux blocs nous servant à recevoir et à envoyer les données, c'est à dire Analog Input et Output.

### Configuration de Simulink:

Avant de pouvoir utiliser c'est bloc, il nous faut configurer le Model, pour cela dans simulation > Configuration Parameters la fenêtre suivante s'ouvrira:



Dans la partie Solver, il faut changer le Type, le passant à Fixed Step, nous pouvons ainsi changer le Sample Time (en n'oubliant pas que si il est changé, il faudra le faire pour tous les blocs dont on se servira pour l'expérimentation), et dans la partie Real-Time Workshop, il faut changer le system target file en cliquant sur Browse et choisissant rtwin.tlc. Nous voilà prêt à utiliser les blocs.

Double cliquer sur un des blocs Simulink Input, choisissant alors dans data acquisition board, la carte d'acquisition PCI 6024-E, changer le Sample Time (si vous l'avait changé) et le numéro de l'entrée ou de la sortie. Il en est de même pour l'output mais il faut aussi mettre « initial value » et « final value » à zéro.

Il ne reste plus qu'à rajouter un bloc en sortie des Analog Input, pour obtenir les valeurs en centimètres, grâce à l'étalonnage que nous avons effectué préalablement et un Scope pour visualiser le niveau.

*Explication de l'implantation de l'étalonnage en annexe.*

#### Configuration du Scope:

Pour pouvoir avoir la courbe complète sur une base de temps souhaiter et sans écrasement des valeurs précédentes, il faut aller dans:

Tools > External Mode Control Panel, puis cliquer sur Signal&Trigger.

La fenêtre une fois ouverte il faut changer la valeur de la duration sachant que la duration doit être multipliée par le Sample Time pour connaître le temps total que prendra l'expérimentation sous Simulink.

Notre système est maintenant prêt à être réguler.

### III-Régulation du système

Nous avons utilisé trois façon différentes pour pouvoir réguler notre système, la première est un correcteur PID normale, la seconde grâce à une régulation en cascade et la troisième avec la logique floue.

#### **1) Correcteur PID**

Ce type de régulation a été étudié en première année d'IMA à polytech'Lille. C'est une des régulations les plus utilisées au milieu industriel, Pour déterminer les paramètres du PID, on a choisi la méthode de Ziegler et Nichols.

Le principe de Ziegler et Nichols est très simple, il consiste à amener le système à un état d'oscillation (Pompage) puis d'en déduire les valeurs des coefficients via un calcul simple.

Dans notre système il faut utiliser le serpent de retard du système pour que le système oscille. Ensuite on détermine le gain critique lorsqu'on est à la limite du pompage, et la constante de temps de la période correspondante.

Pour trouver le gain critique, nous avons effectué plusieurs manipulations avec seulement un correcteur proportionnel, on obtient le gain critique lorsque la courbe est une sinusoïde qui ne varie plus, ce qui nous permet de trouver la période.

Ces deux paramètres sont les seuls nécessaire afin de calculé la suite des paramètres pour faire fonctionner correctement le correcteur complet.

	P	PI série	PI //	PID série	PID //	PID Mixte
Xp	2 Xpc	2,2 Xpc		3,3 Xpc	1,7 Xpc	
Ti	Maxi	Tc/1,2	0,02 Tc×Xpc	Tc/4	85 Tc×Xpc	Tc/2
Td	0			Tc/8	7,5 Tc/Xpc	Tc/8

Grâce au tableau ci-dessus et aux paramètres trouvés, nous avons obtenu le graphe et nos paramètres propres qui suivent :



Le schéma bloc de la commande du niveau H1 est donné ci-dessous:

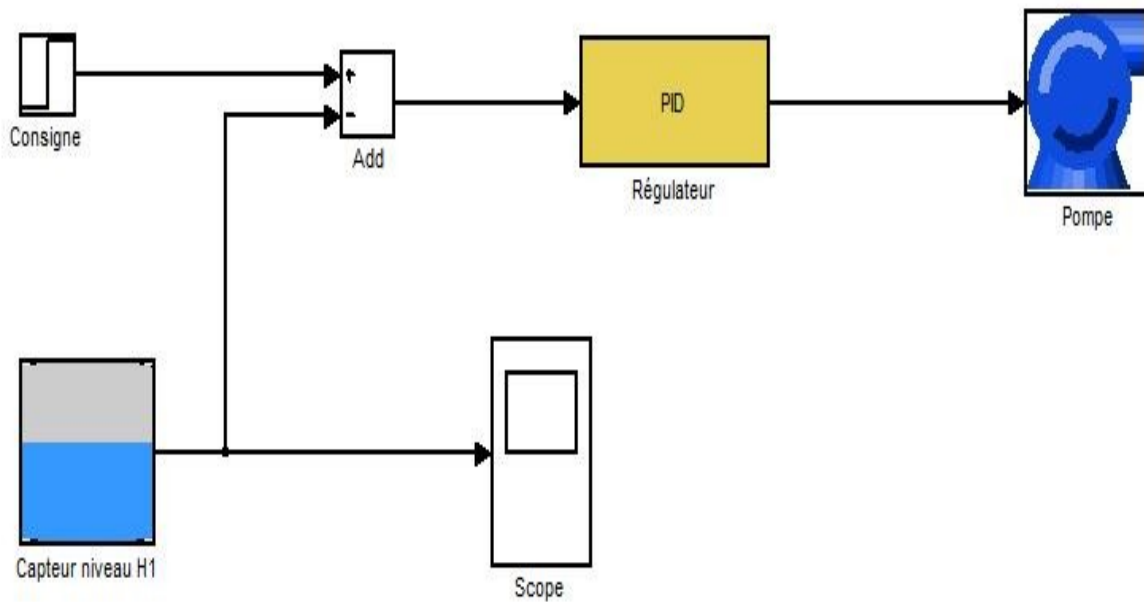


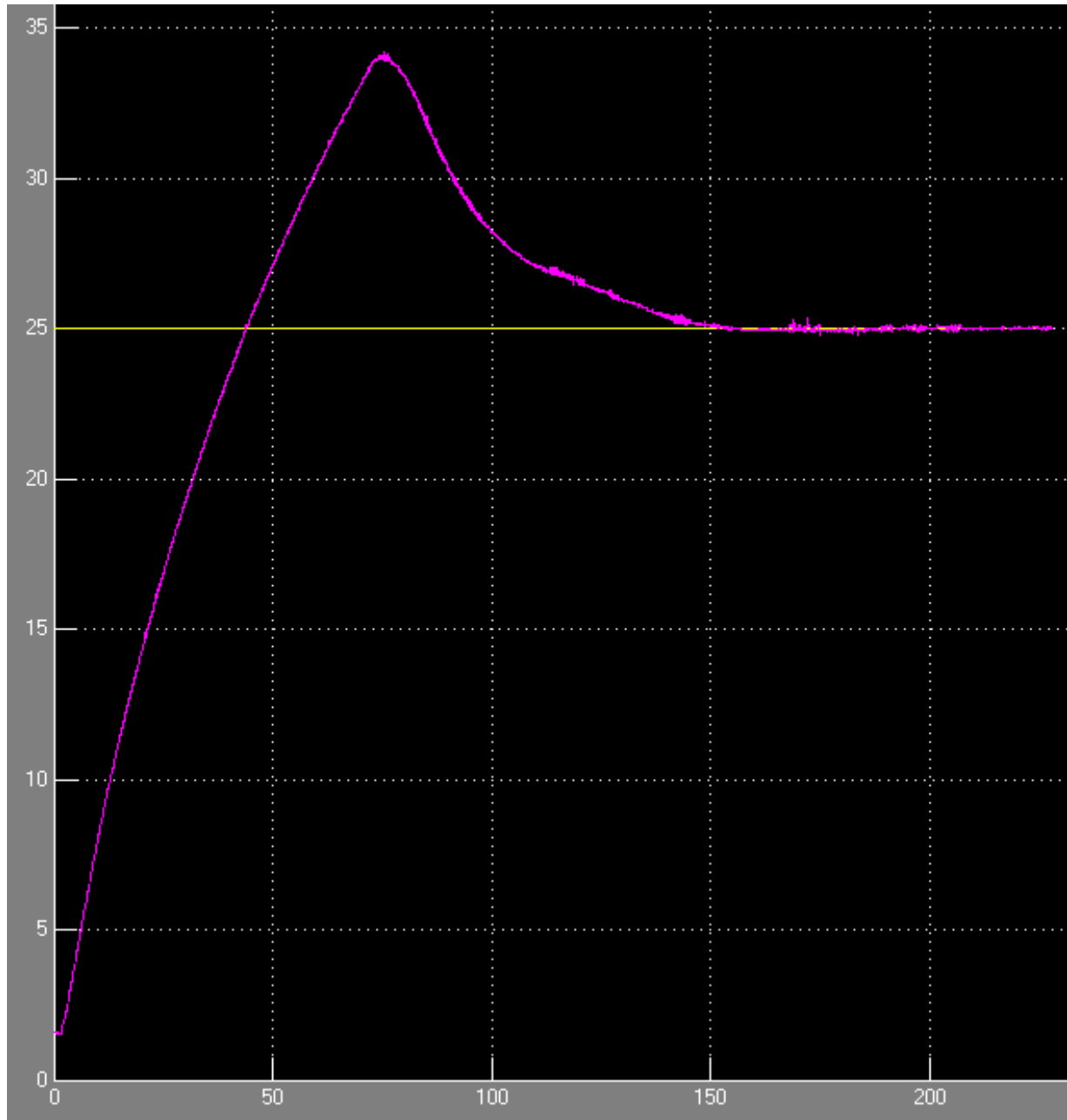
Illustration 1: Schéma block

La régulation du niveau de l'eau dans le réservoir 1 ou 2 dépend des vannes qui sont ouvertes. Et donc nous avons constaté que les paramètres du PID change en fonction des vannes ouverte.

Le tableau ci-dessous représente les paramètres du PID en fonction des vannes ouverte.

	Vanne 1 ouverte	Vanne 1,2 ouverte	Vanne 1,2,3 ouverte
$K_p$	0.86	1.1818	8.82
$K_I$	0.036	0.058	4.41
$K_D$	0	0	4.51

La figure ci-dessous présente la courbe du niveau H1 obtenue, et la consigne demandée :



On remarque que le niveau H1 demandée est atteint, il se stabilise à 150 s. avec la présence d'un dépassement, ce qui est normal dans le cas de l'utilisation d'un correcteur PID.

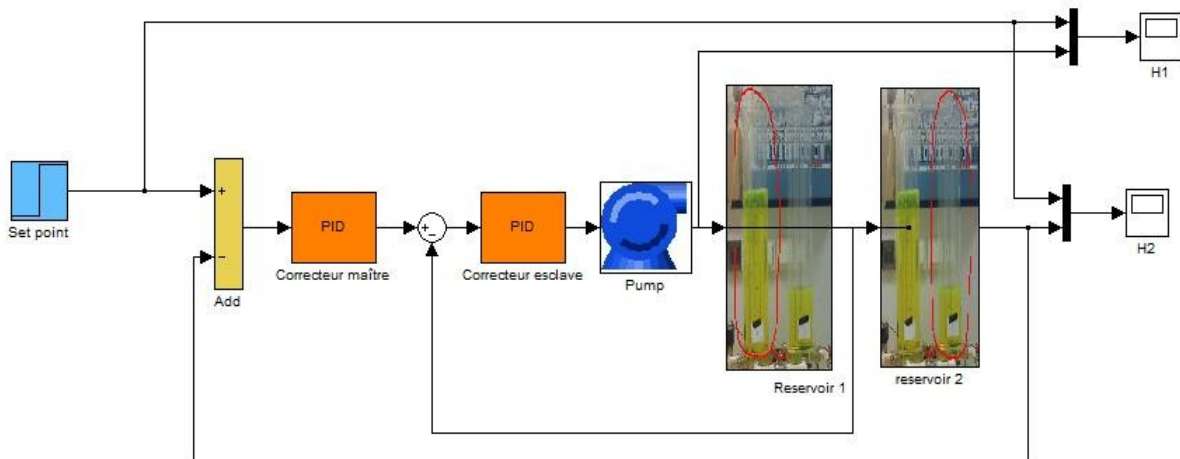
## 2) Régulation Cascade

Ce type de régulation est surtout utile avec des systèmes lents (ce qui est notre cas, comme on a pu le voir avec le correcteur PID).

La régulation en cascade permet de compenser les actions perturbatrices sur le réservoir R1 avec le correcteur esclave avant qu'elles ne se répercutent sur le réservoir R2.

Cette régulation nous sert surtout à avoir une commande sur le deuxième réservoir en fonction du premier et non l'inverse.

Schéma bloc de la régulation en cascade :

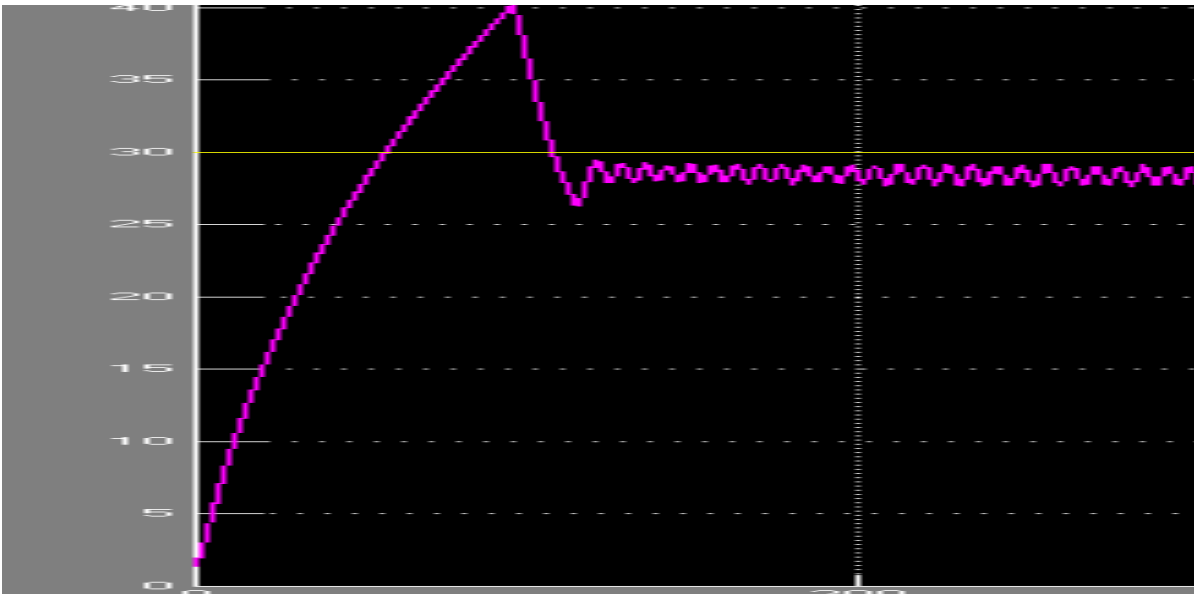


On utilise donc deux correcteur pour un tel système, un correcteur maître et un esclave. Le correcteur esclave peut être remplacé par un simple correcteur proportionnel car il n'agit que faiblement sur la régulation du réservoir deux, étant surtout fixé au réservoir un.

Le correcteur maître est par contre un correcteur PID, c'est lui qui va faire la plus grosse partie de la régulation.

Nous avons utilisé les mêmes paramètres que pour la régulation du PID et obtenu le résultat suivant.

La régulation cascade nous a permis d'avoir la courbe du niveau H1 ci-dessous :



On remarque que la régulation cascade ne résout pas le problème de dépassement ni l'erreur statique.

### 3) Commande Floue

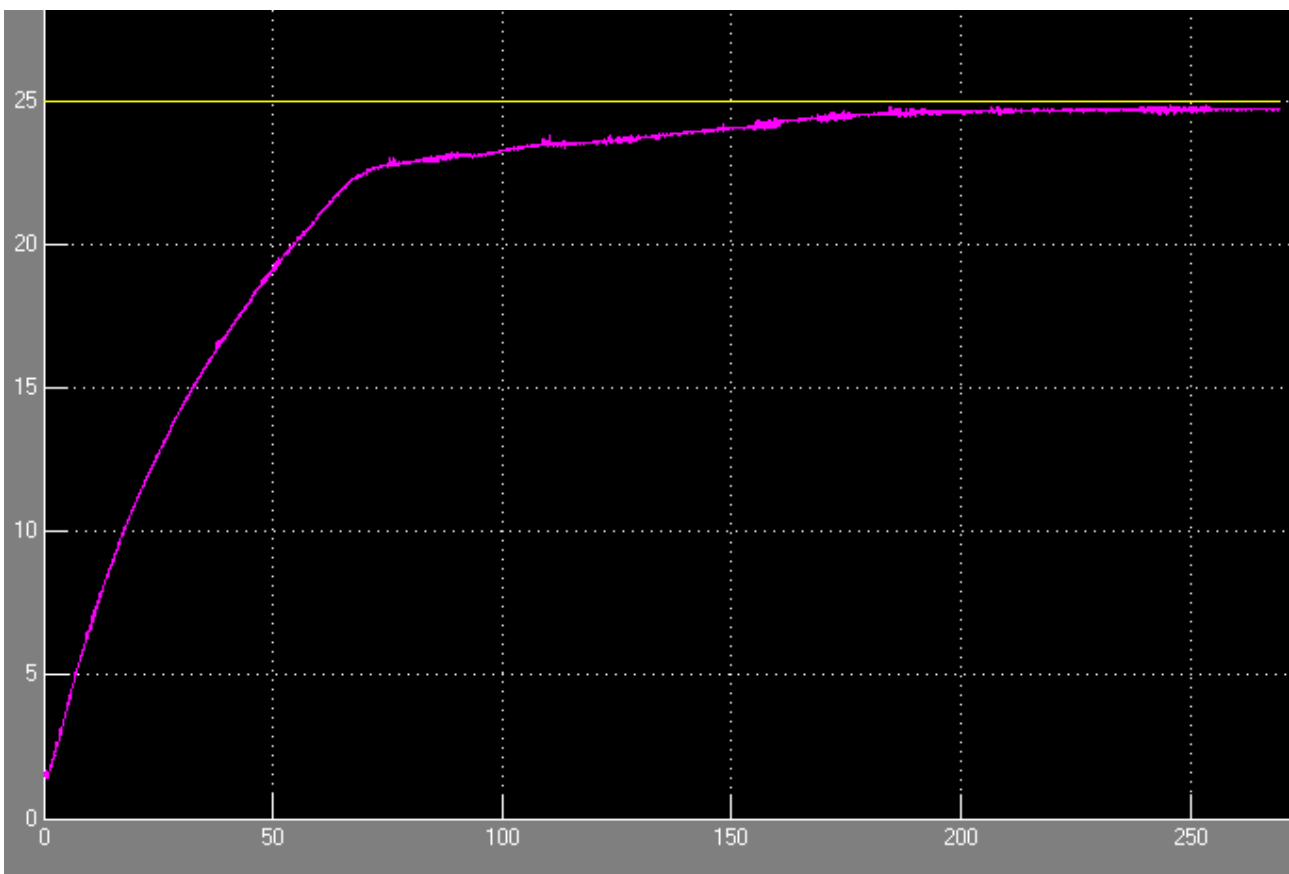
La logique floue est une extension de la logique classique qui permet la modélisation des imperfections des données et se rapproche dans une certaine mesure de la flexibilité du raisonnement humain.

La première étape du traitement d'un problème par la logique floue consiste à modéliser chacune des entrées du système par des courbes donnant des degrés d'appartenance à différents états identifiés pour ces entrées.

La logique Floue est une commande qui permet de corriger l'erreur dans un système sans avoir une connaissance du modèle mathématique du système.

*Vous retrouvez dans l'annexe la méthode à suivre pour réaliser ce type de commande.*

Après réalisation de celle-ci on obtient la courbe suivant pour une commande de niveau H1.



On remarque que le niveau H1 demandé est atteint, le temps de réponse est très grand, mais la réponse du système ne présente aucun dépassement. Par contre nous remarquons aussi que ce type de régulation ne résout pas toujours l'erreur statique même si le système est alors très stable.

## **Conclusion**

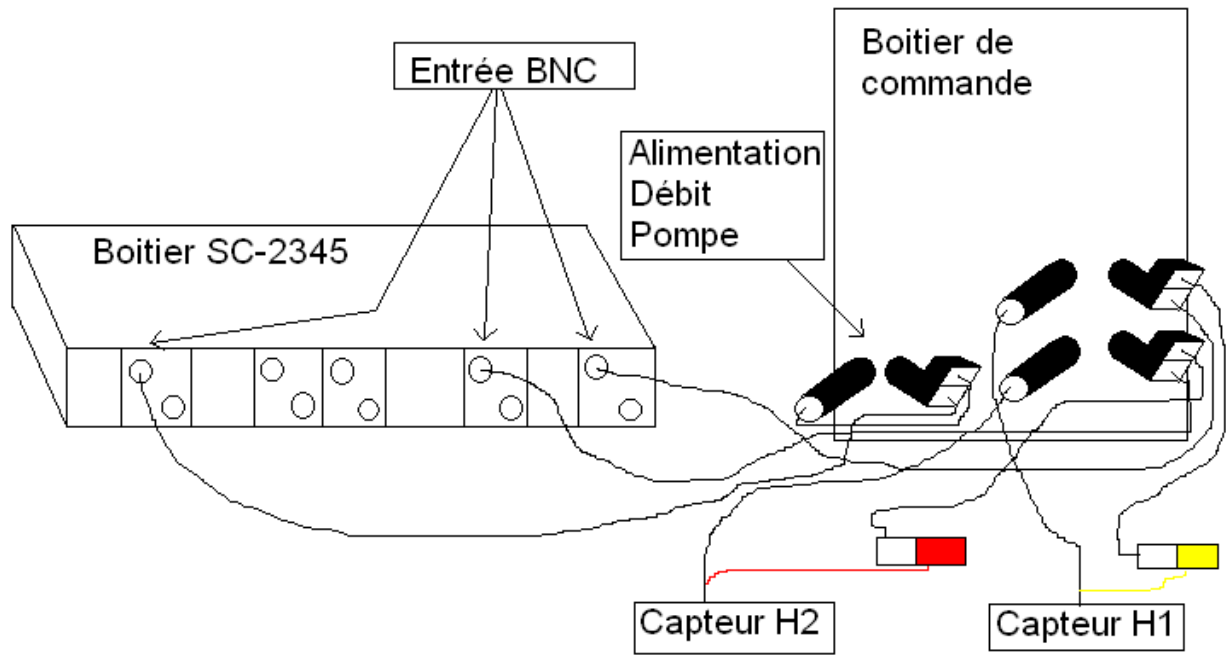
Un tel projet a été très bénéfique pour notre compréhension des outils utilisés lors des Travaux dirigés ou des Travaux pratiques, car le système est un système réel sur lequel nous pouvons avoir à faire dans notre future vie professionnelle.

Mais il nous a surtout permis de nous rendre compte qu'il y a toujours des problèmes lors d'un tel exercice, tel que la compréhension d'un tel système d'acquisition sur lequel nous avons passé presque un tiers de notre temps ou encore des paramètres à utilisés pour avoir des résultats lisible et compréhensible, à la place de pouvoir utiliser le système. Cette perte de temps nous a montré que la prise en compte d'un tel événement n'est pas négligeable dans la vie réelle.

Pour finir ce projet nous aura apporté tout ce que nous pouvions attendre d'un tel sujet, sans compter le travail en groupe pour la répartition des tâches, aussi bien pour la rédaction que pour l'expérimentation et la prise de résultat.

# ANNEXE

## Câblage du système



## Acquisition de l'étalonnage

Nous allons utiliser le logiciel Measurement & Automation afin d'obtenir un étalonnage assez précis entre la tension donnée par le boîtier et la mesure en centimètre de la cuve.

Après ouverture du logiciel:

Cliquer tout d'abord sur Périphériques et interfaces.

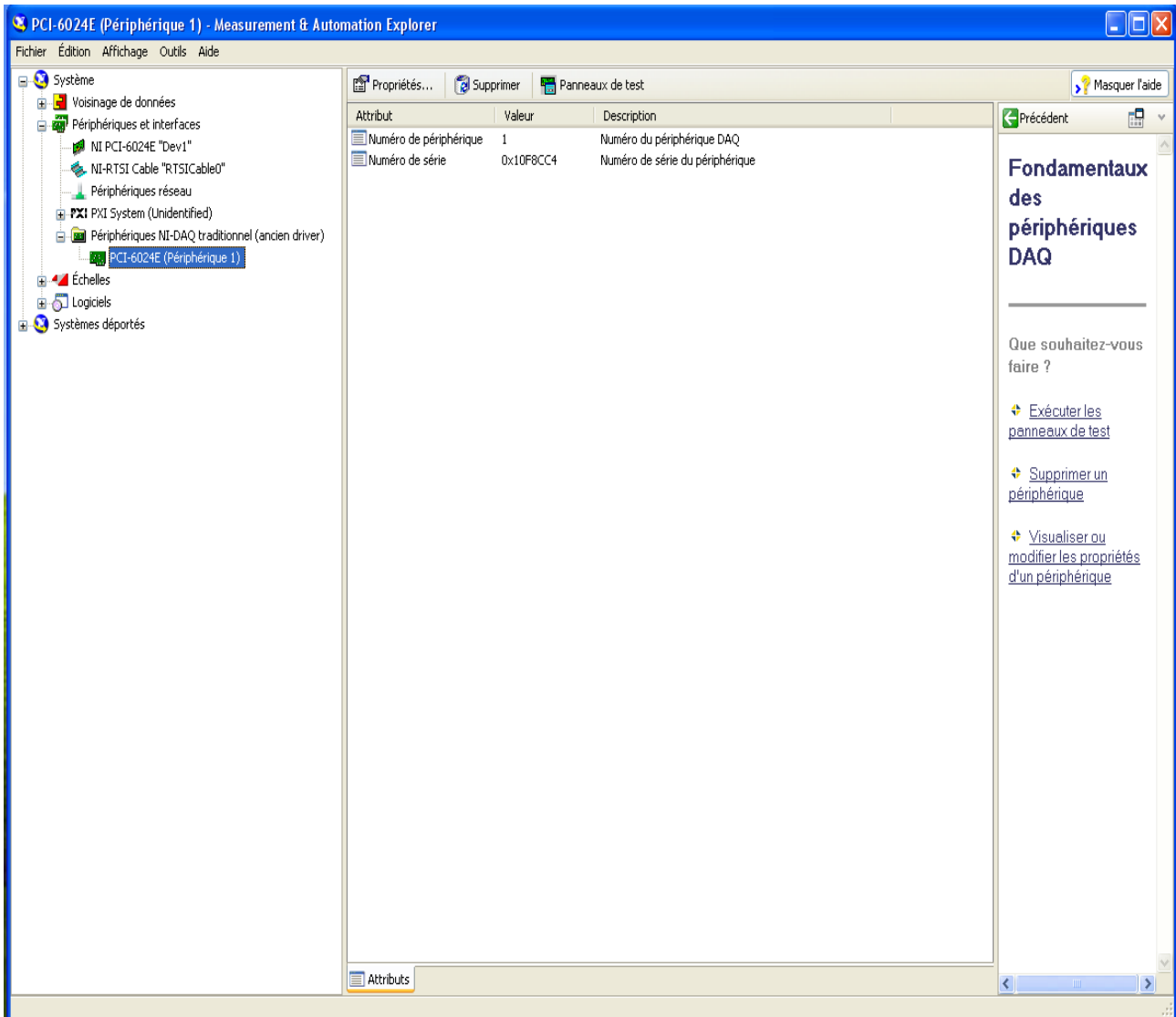
Normalement le logiciel détecte automatiquement la carte d'acquisition PCI 6024-E, si elle est installée, que nous devons configurer, si ce n'est pas le cas, installer les drivers est chose facile avec Windows qui propose de l'installer.

Nous verrons par la suite qu'il faut créer des tâches pour avoir l'étalonnage des capteurs.

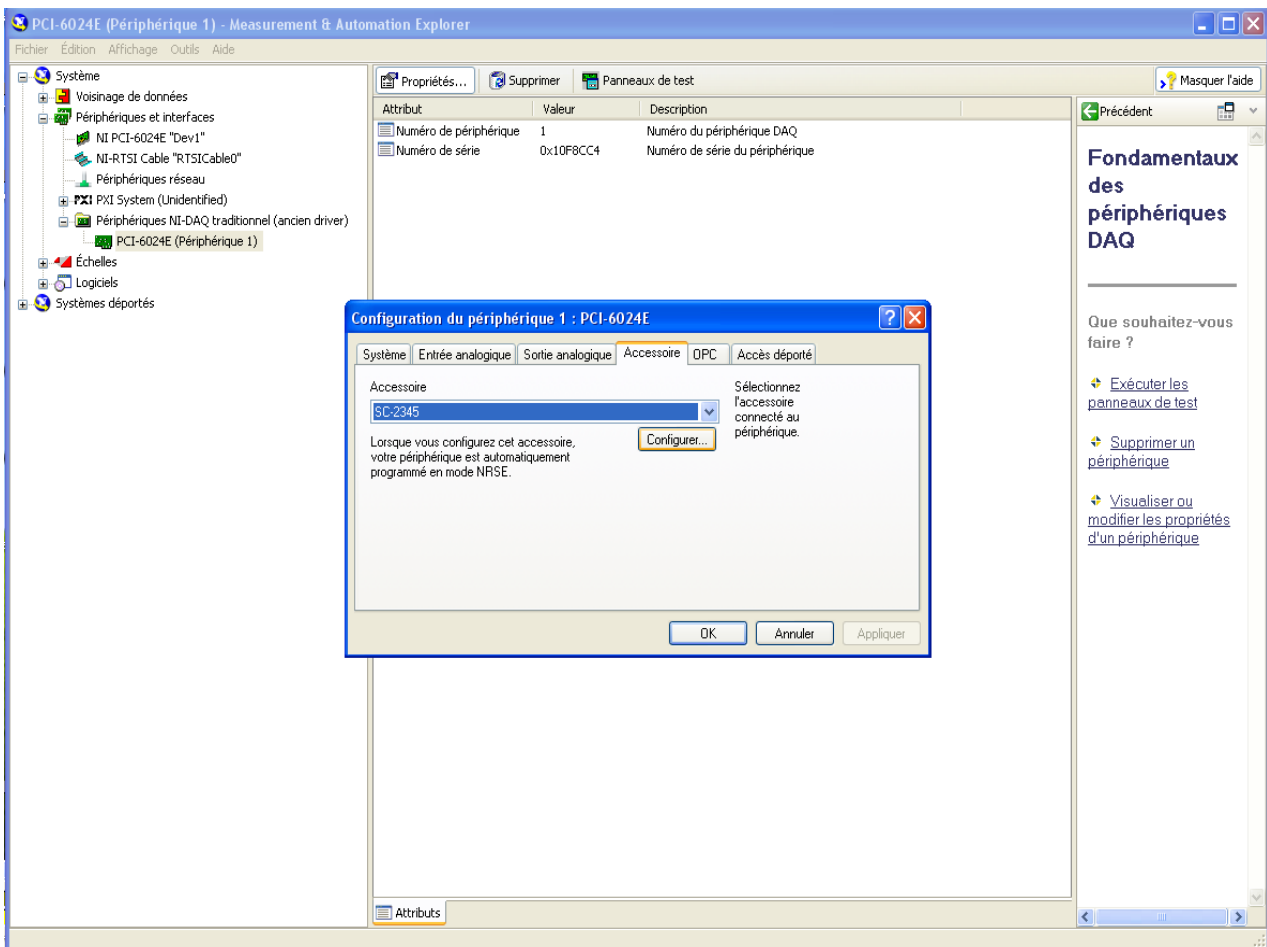
Une fois la carte d'acquisition détectée par le logiciel, vous la verrez apparaître dans l'onglet Périphériques et Interfaces sous le nom : Périphériques NI-DAQ traditionnel.

Sélectionner alors le PCI 6024-E et cliquer sur propriétés.



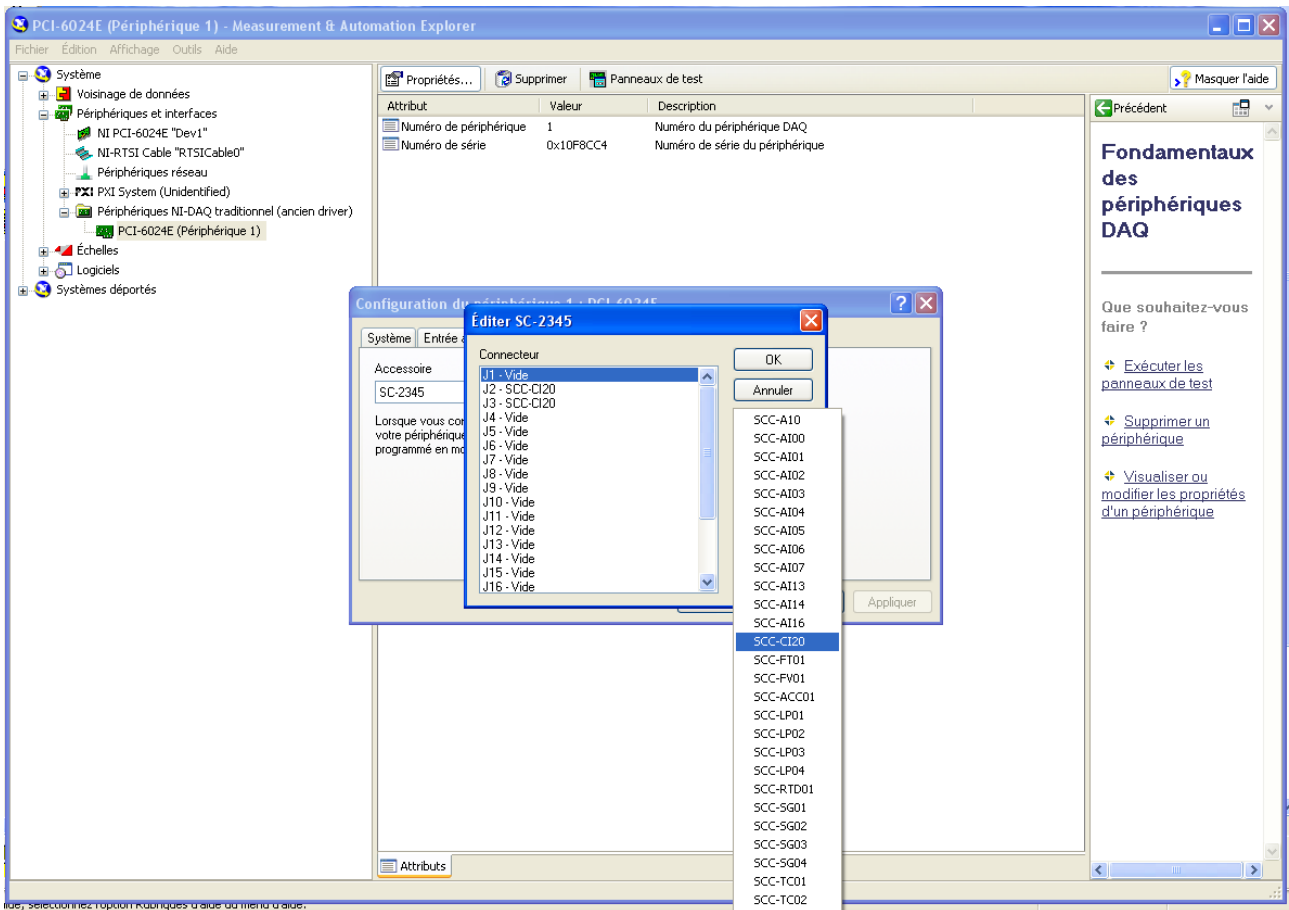


Après avoir cliqué sur propriété une fenêtre configuration du périphérique, aller dans accessoire, sélectionner l'accessoire SC-2345 puis cliquer sur configurer, comme ci-dessous.



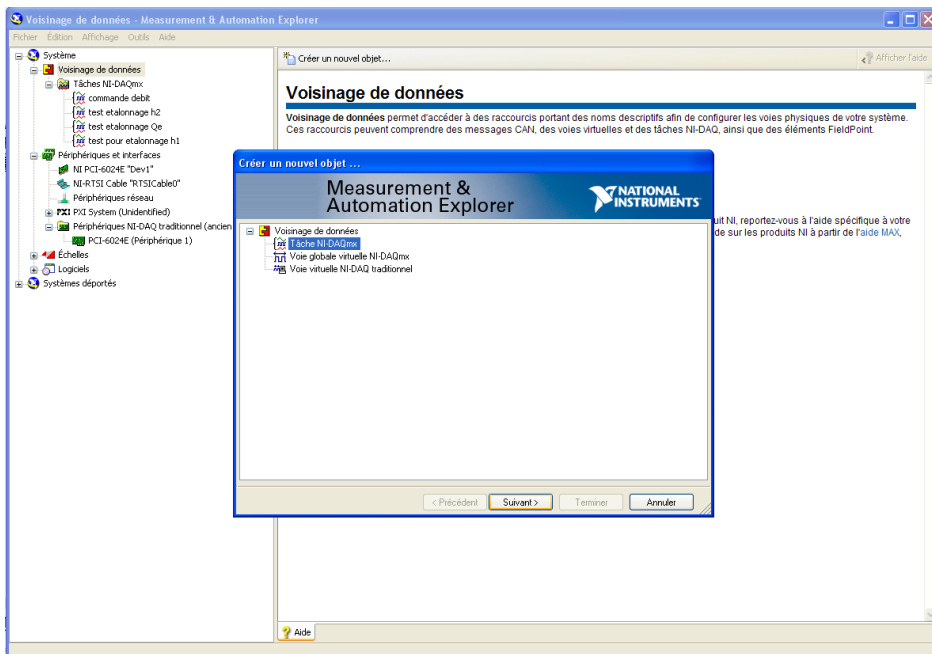
A l'intérieur du boîtier SC-2345, il y a 5 composants électriques pour l'acquisition des données ou pour activer la pompe.

Sélectionner alors pour le J1, J2 et J3 un SCC-CI20, pour le J17 un SCC-CO20 et pour le J21 un SCC-PWR02, puis cliquer sur ajouter.



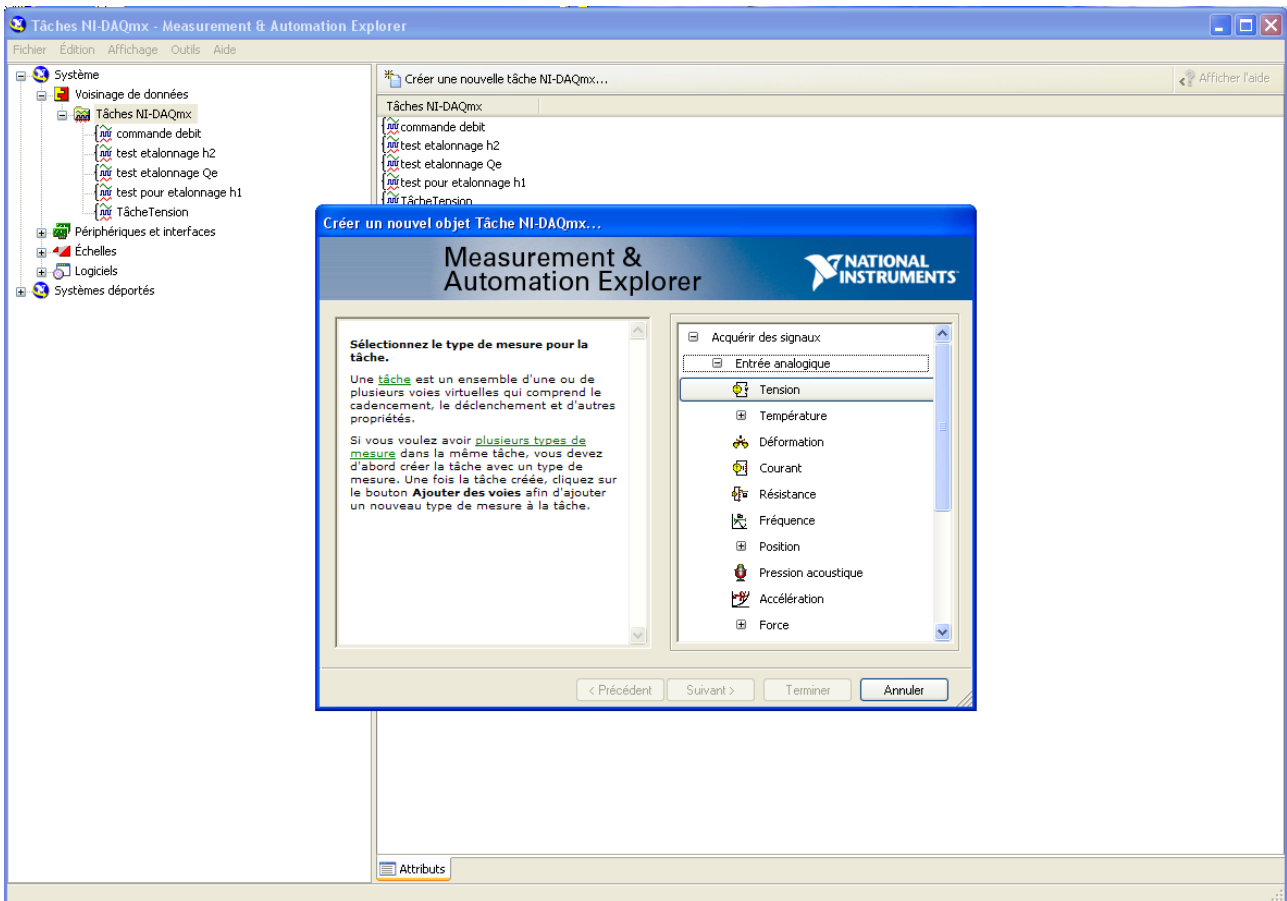
Votre périphérique est alors configuré et les mesures peuvent être obtenues.

Il ne nous reste plus qu'à effectuer l'étalonnage des capteurs à l'aide du logiciel. Pour cela aller sur voisinage de données, faire un clic droit avec la souris puis créer un nouvel objet.



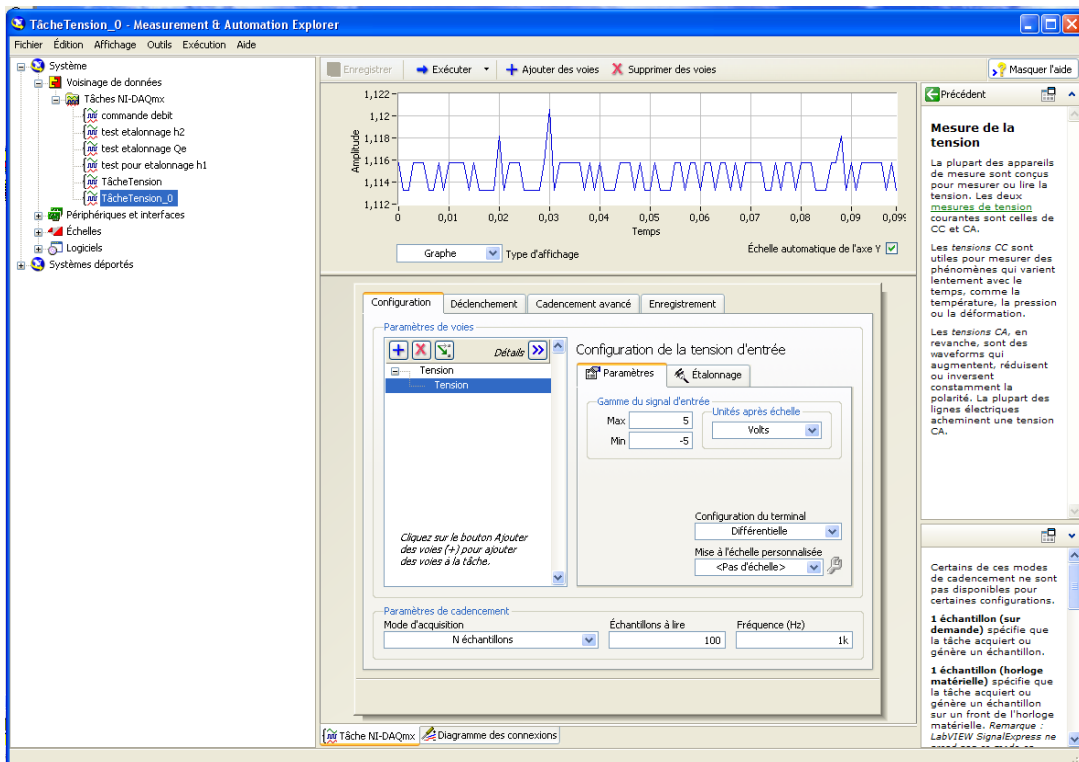
Sélectionner Tache NI-DAQmx, puis cliquer sur suivant.

Choisir dans Acquérir des signaux > Entrée analogique > Tension, puis cliquer sur suivant



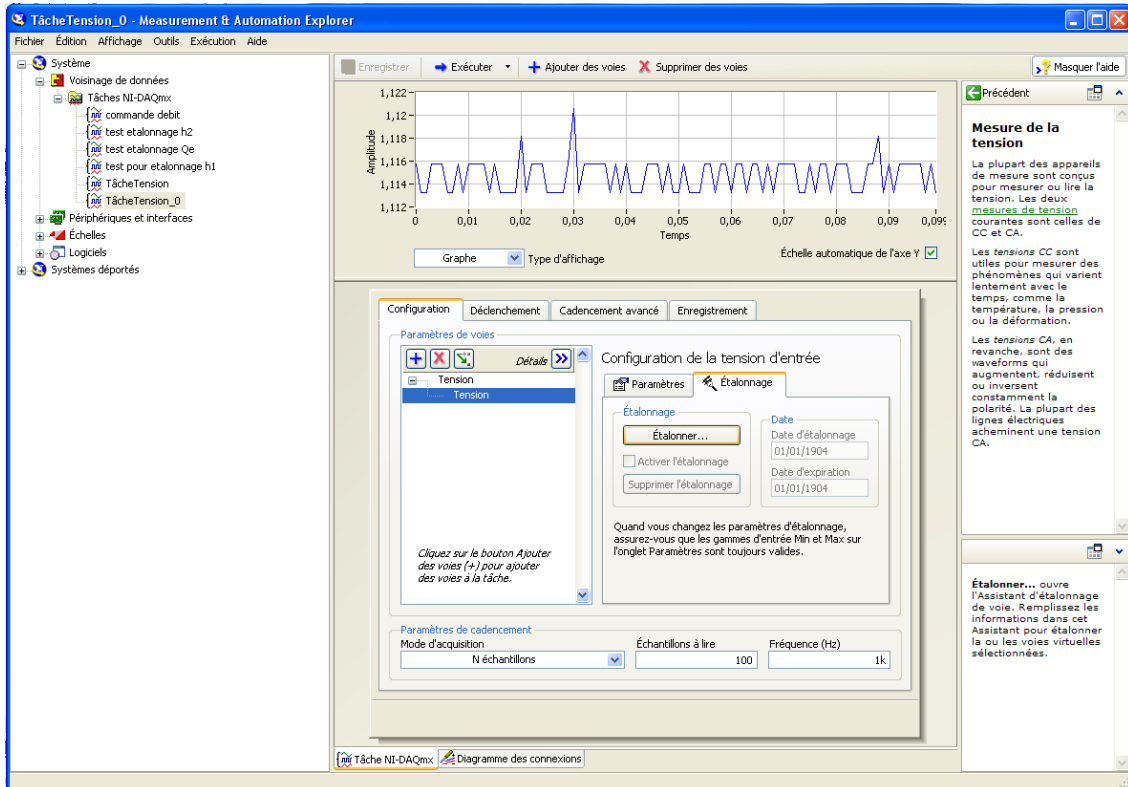
Sélectionner alors les entrées qui vous intéresse, c'est-à-dire a0 pour le capteur H1 et pour le capteur H2, puis cliquer sur terminer.

Sélectionner maintenant la tâche que vous avez créée pour la paramétrer.

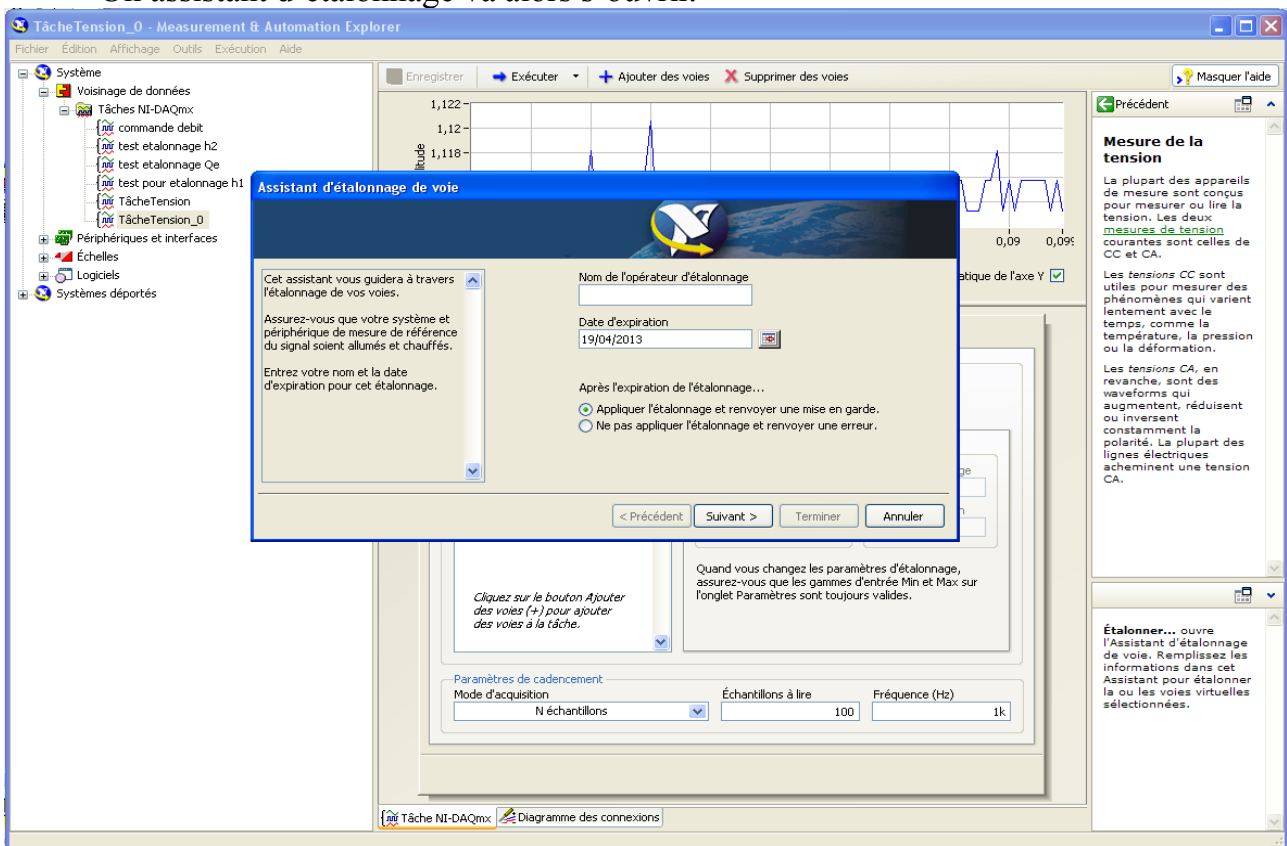


Entrer une fréquence de 20kHz, un nombre d'échantillons à lire de 50kHz et une

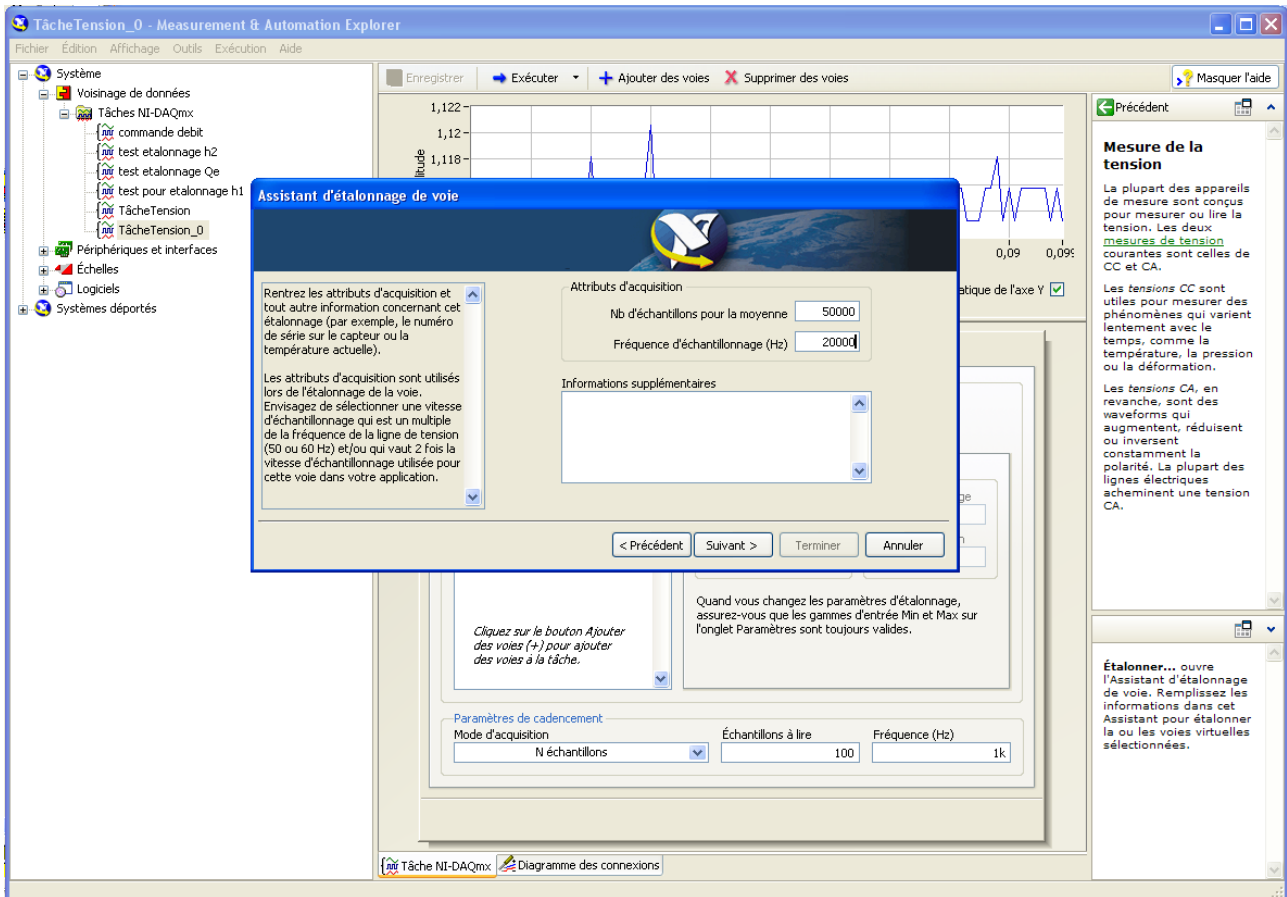
gamme de signal d'entrée de 0 à 10 Volts, puis aller dans l'onglet Etalonnage et sur Etalonner.



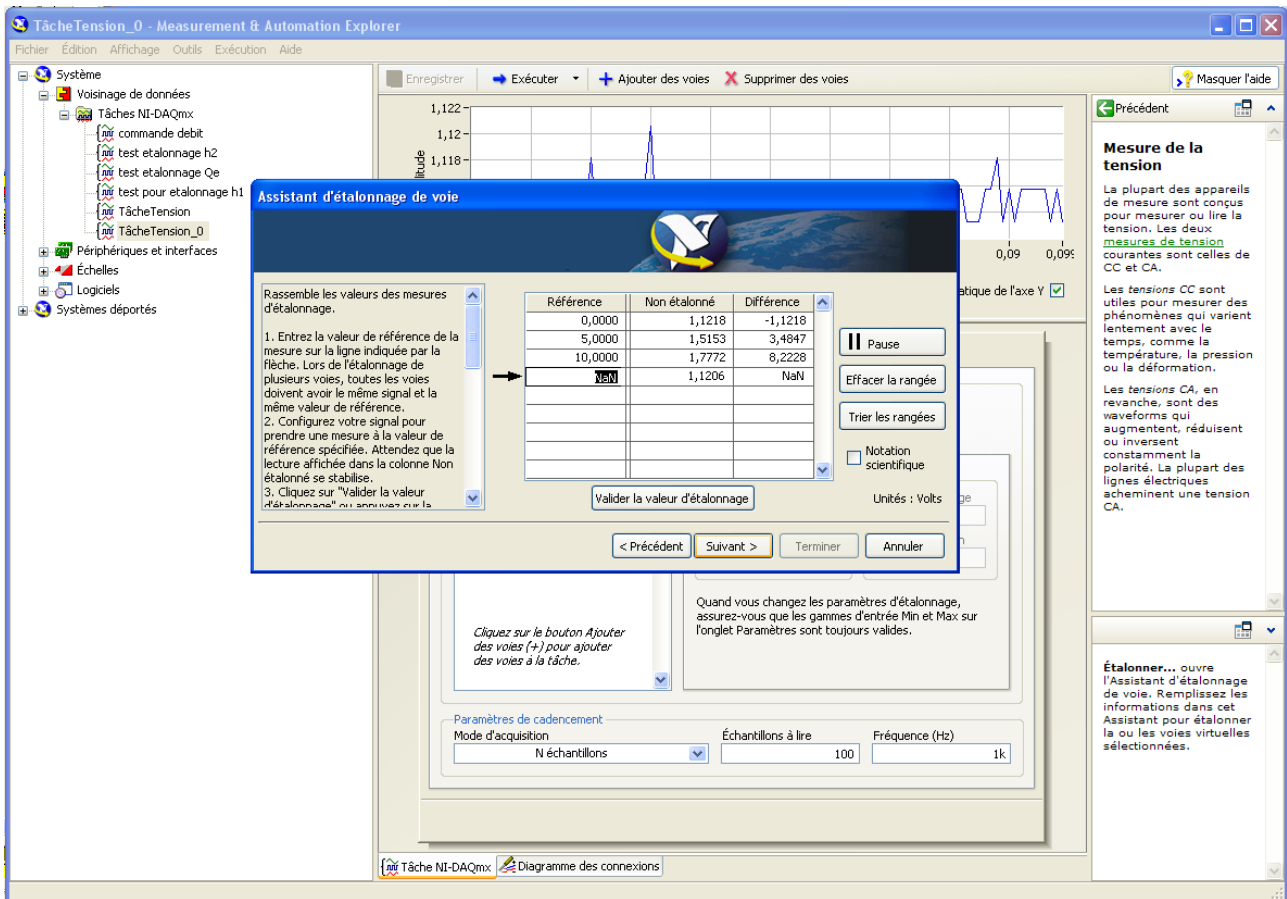
Un assistant d'étalonnage va alors s'ouvrir.



Donner lui un nom pour la suite puis cliquer sur suivant.



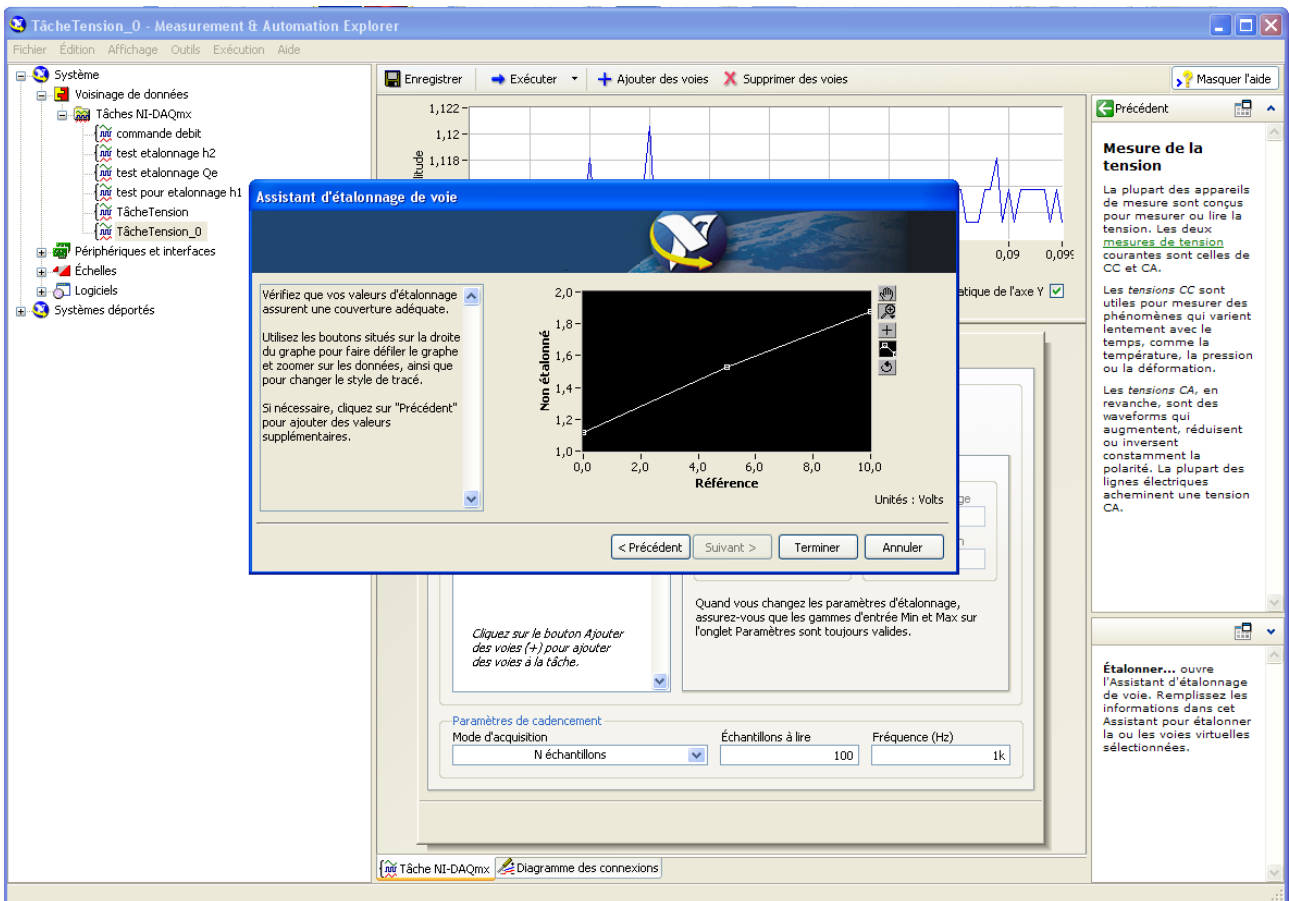
Rentrer les différents attributs d'acquisition comme ci-dessus.



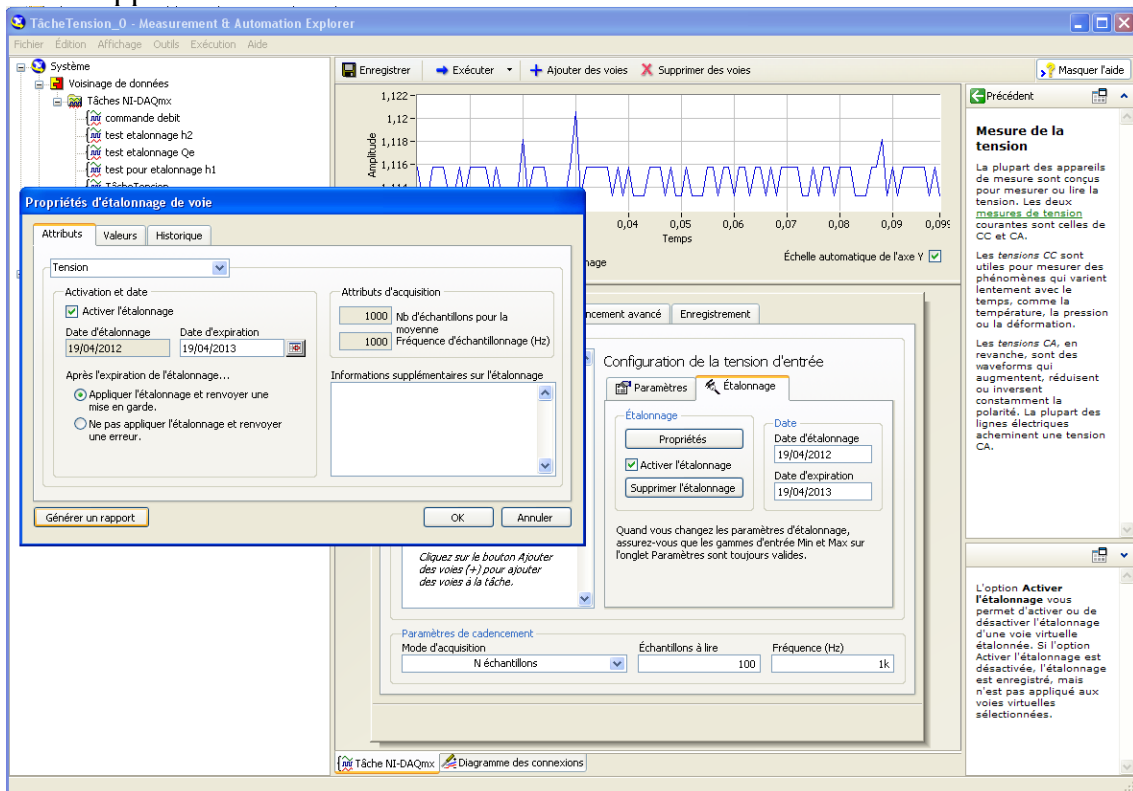
Vous pouvez commencer l'étalonnage.

Mettez (cuve à vide) la référence « 0 » puis validez la valeur d'étalonnage, remplissez la cuve de 5 en 5 et validez la valeur à chaque fois jusqu'à 50 cm (la référence est la hauteur).

Une fois l'étalonnage de la cuve 1 terminé, cliquez sur suivant pour visualiser la courbe obtenue (comme ci-dessous).

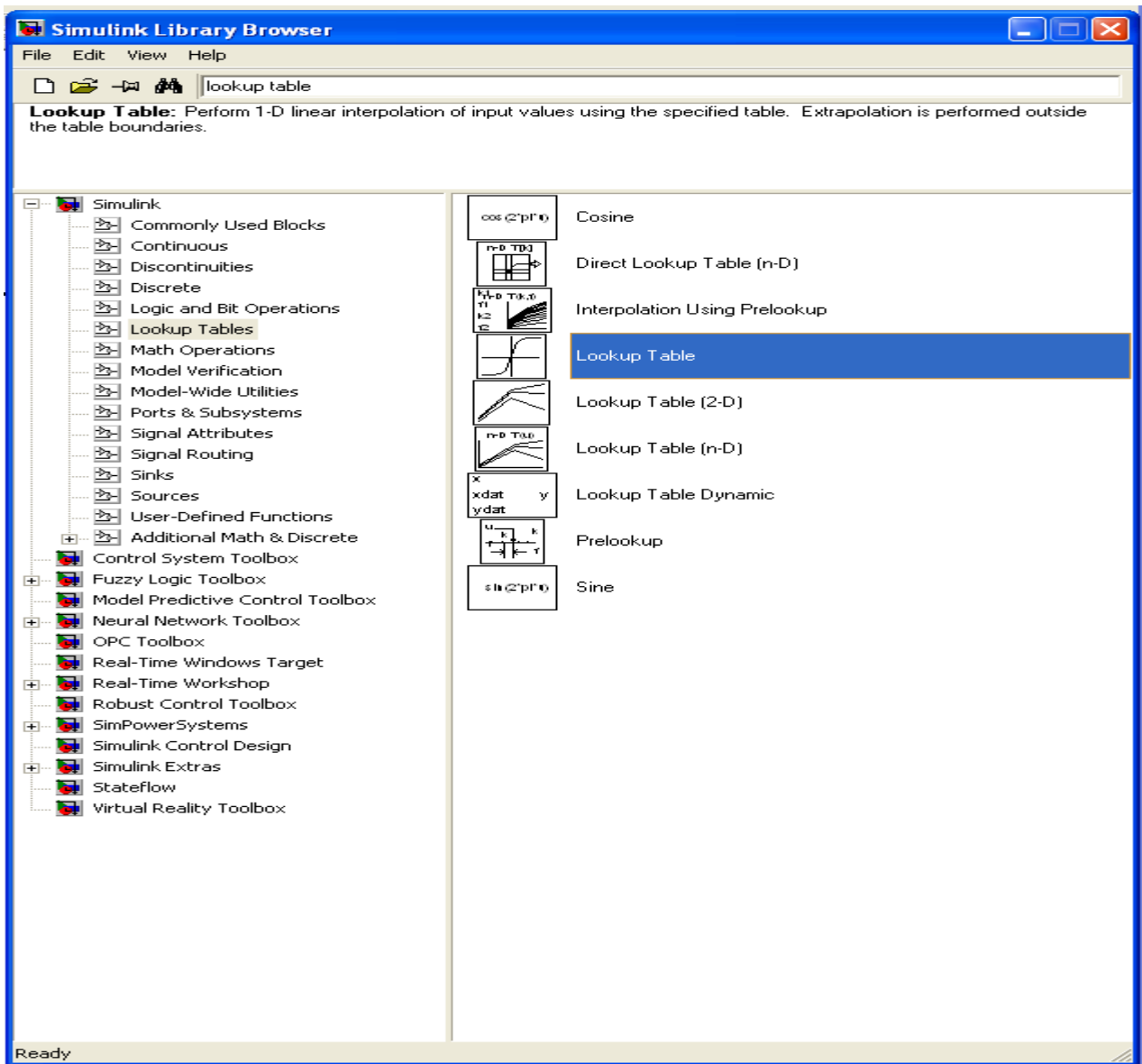


Pour utiliser ces valeurs, appuyer sur propriétés dans l'onglet Etalonnage puis générer un rapport.

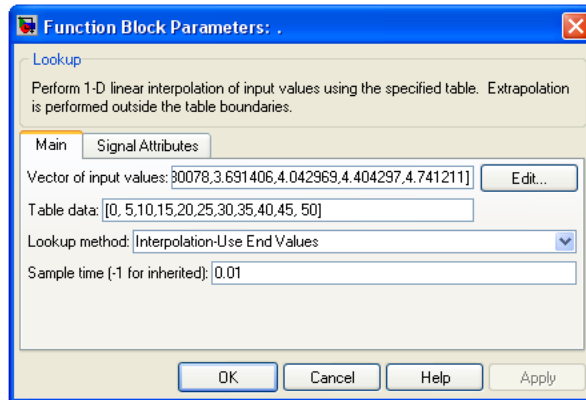
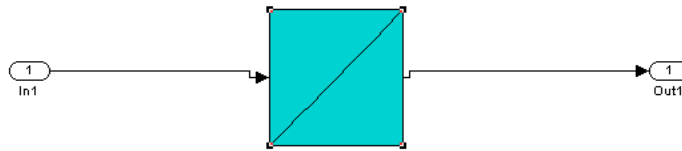


Maintenant sur votre Model Matlab Placer le bloc Look Up Table



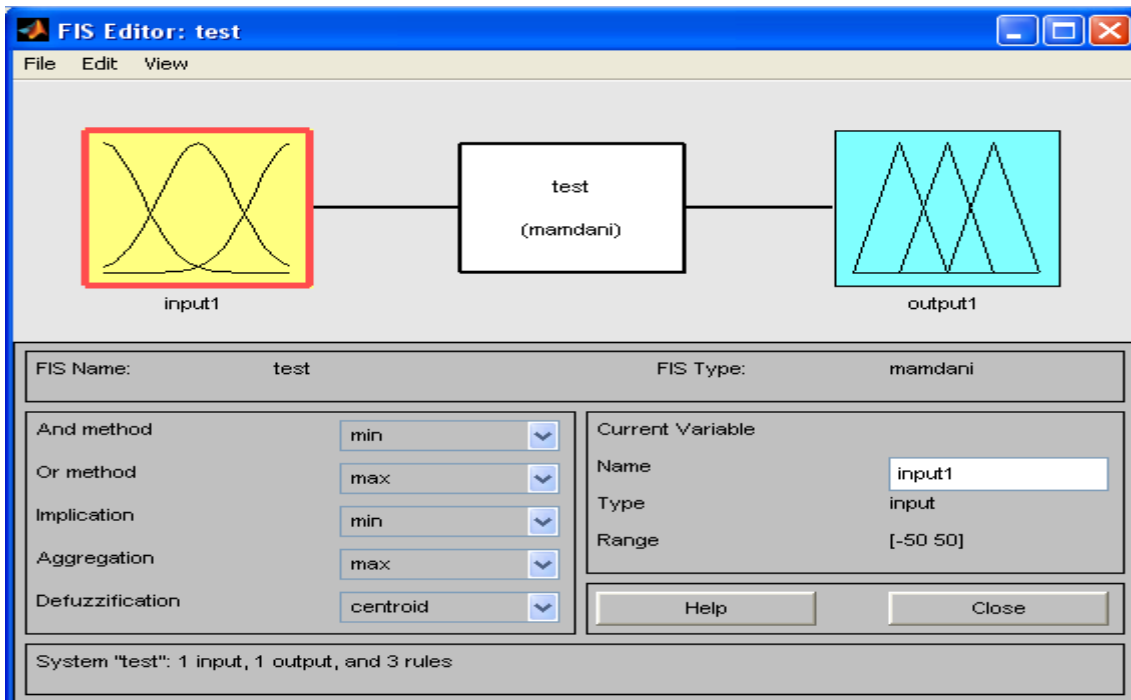


Double cliquer dessus, puis entrer les valeurs d'entrée qui sont dans le rapport générer dans la case Vector of input values et les valeurs de la sortie dans la case Table data.

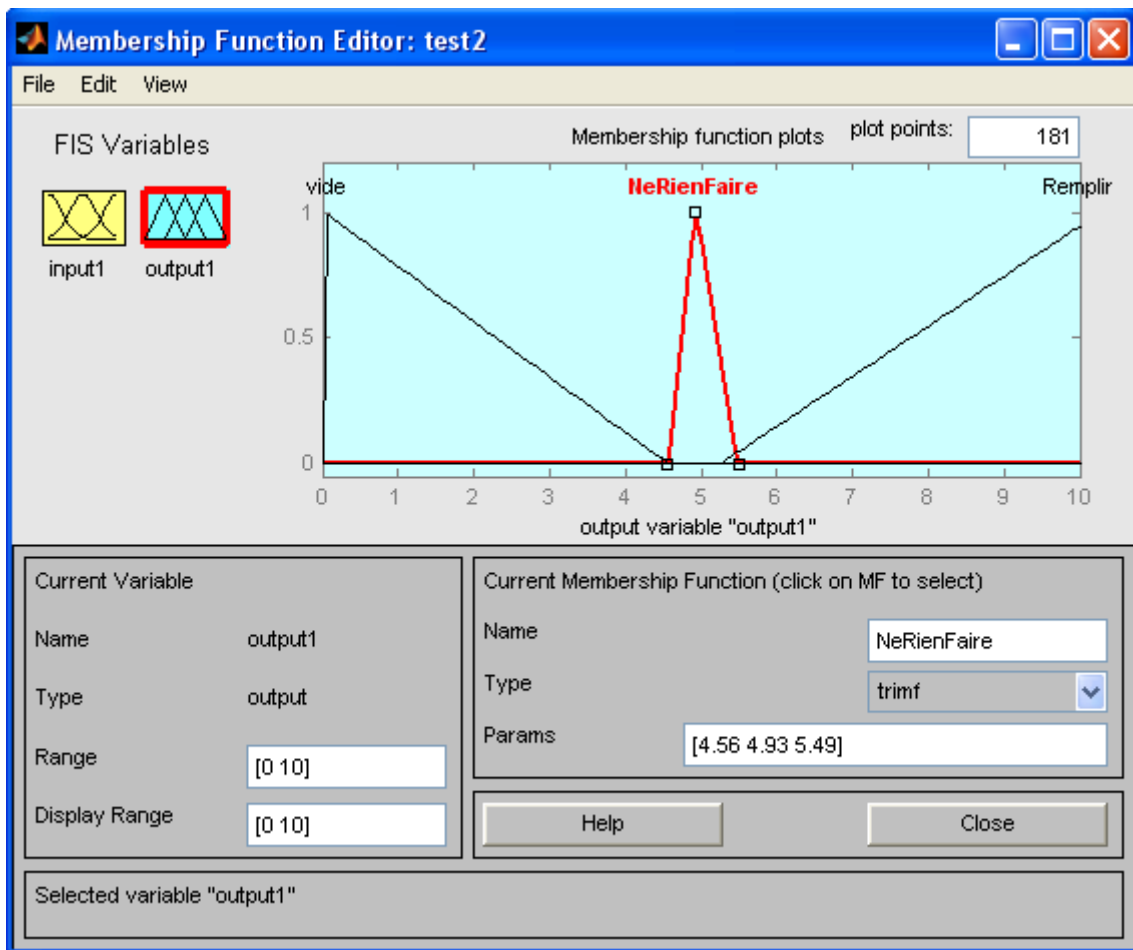


### Réalisation de la commande Floue

D'abord il faut ouvrir Matlab, ensuite cliquer sur start → ToolBoxes → Fuzzy Logic.  
La fenêtre suivante s'ouvre :



Ensuite il faut cliquer sur input.  
Cette fenêtre permet de définir les fonctions d'entrées et de sorties.



Ensuite aller dans edit → Rule.

Cette fenêtre permet de définir l'algorithme de la commande du système c'est à dire. Que doit générer le correcteur en sortie pour chaque entrée (erreur).

A la fin n'oubliez pas d'enregistrer. Ensuite cliquez sur File → Export → To Workspace.

