

Projet Interface de supervision avec RTW et LABVIEW



Encadré par M.Belkaem Ould-Bouamama

Réalisé par Ilyas Mabrouk
Shitao XING

Polytech Lille Département Informatique Microélectronique Automatique

Le 7 Mai 2013

Remerciement

Nous tenons d'abord à exprimer notre reconnaissance et nos profonds remerciements à toute personne nous ayant aidé de près ou de loin à la réalisation de ce projet à ses différents étapes.

Nous tenons à remercier plus particulièrement M Belkacem OULD BOUAMAMA pour la confiance qui nous a accordée, ainsi que sa disponibilité et le temps qui nous a consacré tout au long du projet.

Sommaire

I. Présentation du projet

- 1. Système étudié**
- 2. Objectifs du projet**

II. Interface Labview et boîte à outil RTW

- 1. Solutions possibles**
- 2. Le SIT**

III. Commandes du système

- 1. Limites du PID utilisé**
- 2. Prédicateur de Smith**
- 3. Commande floue**

IV. Conclusion

V. Annexes

I. Présentation du projet

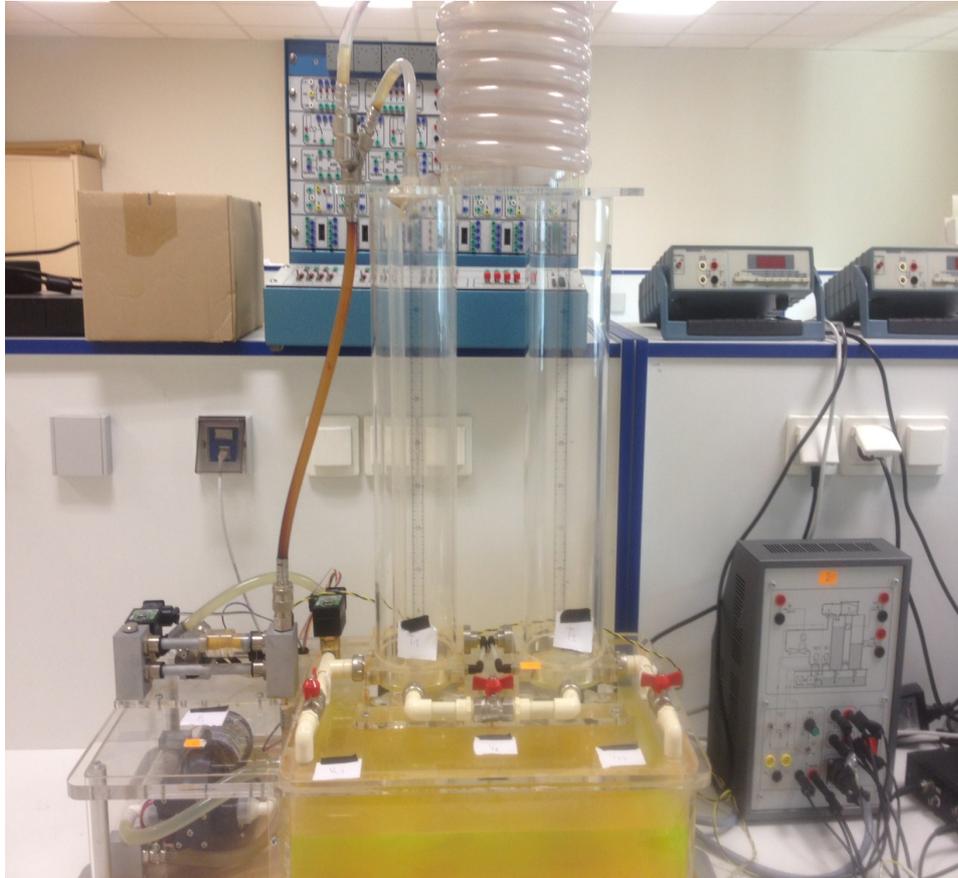


Figure 1

1. Système étudié

Il s'agit d'un système hydraulique constitué de deux réservoirs (T1 et T2) alimentés par une pompe et qui pouvant communiqués à l'aide d'une vanne commandable. Ce système est la représentation d'une station hydraulique réelle qui doit être régulée en niveau à l'aide d'une interface de supervision distante. Il contient deux capteurs de niveaux pour chacune des deux cuves et un capteur de débit à la sortie de la pompe. Toutes les entrées/ sorties sont connectées à une carte d'acquisition National Instrument PCI-2345 qui joue le rôle d'un convertisseur analogique numérique et fournit toutes les données utiles à la boîte à outils RTW de Matlab/Simulink. Les différentes perturbations sont modélisées par des

vannes (robinets) situées à différentes positions pour permettre de voir en temps réel l'action des effets parasites et la capacité du système, plus particulièrement le correcteur utilisé, à les compenser et atteindre les consignes désirées. Un serpentin est mis en place pour simuler un éventuel retard. (Annexe 1).

2. Objectifs du projet

Notre projet est constitué principalement de deux parties: La première consiste à améliorer l'aspect visuel du modèle proposé en utilisant l'interface de Labview le tout en temps réel. La deuxième consiste à corriger quelques limites du PID mis en place par de nouvelles commandes.

II. Interface Labview et boîte à outils RTW

LabVIEW et Matlab/Simulink développés respectivement par les entreprises américaines National Instruments et The MathWorks sont à la fois langages de programmation et logiciels. Ils accompagnent les ingénieurs et scientifiques du monde entier dans les différentes étapes de conception, modélisation, simulations, prototypage, tests, et déploiement d'un produit technologique.

Labview a la caractéristique d'être l'un des logiciels les plus simples à utiliser pour la supervision et de fournir des résultats très satisfaisants visuellement. En partant du modèle Simulink fourni, il est demandé de garder les solveurs de Matlab et d'exploiter les résultats (les états des capteurs) à l'aide de Labview. Il est donc nécessaire de trouver une ou des méthodes pour permettre une liaison ou un partage des données entre ces deux logiciels, de développeurs différents, et ceci en temps réel.

1. Solutions possibles

La solution intuitive, puisqu'il s'agit d'avoir des résultats plus exploitables visuellement, est de récupérer le PID utilisé dans Matlab/Simulink et de refaire tout le modèle sur Labview.

Cette solution n'a pas été choisie pour deux raisons: D'une part parce que la configuration du PID du modèle Simulink a été faite avec des tables d'étalonnages propres à ce modèle, et donc il est nécessaire, soit de reconfigurer le PID, soit de refaire les tables d'étalonnage pour Labview.

D'autre part, en choisissant cette solution on sort du cadre du cahier des charges, car il a été bien précisé qu'il faut garder les solveurs de Matlab/Simulink.

La deuxième alternative est la création d'un tunnel ActiveX entre l'interface réalisée sur Labview et un tableur (Excel par exemple) puis récupérer les données des entrées /sorties et l'action du PID sous formes de tables numériques sur ce dernier.

Cette solution se réalise facilement, mais les données récupérées du modèle Simulink ne sont prises que dans un intervalle de temps bien limité. Le tunnel ActiveX fonctionne avec un tableur statique, c'est -à- dire qu'il ne change pas de valeurs en temps réel. Cette solution est rejetée.

La dernière solution trouvée est un module créé par National Instrument, qui permet d'acquérir les données d'un modèle Simulink à partir d'un fichier C créé à partir de ce modèle. C'est le Simulink Interface Toolkit.

2. Le SIT

Principe:

En installant le module SIT, une nouvelle librairie s'ajoute dans la configuration de Simulation au niveau de Simulink. Lorsque le modèle en question est fini et compilable on construit un fichier DLL à partir de ce dernier qui n'est autre qu'un code en C traduisant tous les composants et les comportements du modèle. Après quelques configurations au niveau de Labview on importe le fichier DLL et on fait un mapping (des liaisons) entre les entrées sorties de l'interface de Labview avec celles du modèle Simulink.

Exemple:

On a réalisé un exemple simple pour voir les possibilités de cet outil, au moins en mode simulation. On a créé une sinusoïde dans un modèle Simulink et on a réalisé une interface dans Labview permettant le contrôle de l'amplitude et de la fréquence de cette sinusoïde. Vous trouverez dans l'annexe les différentes étapes pour configurer les environnements Labview et Simulink pour ce genre de simulation. Ci-dessous l'illustration de l'exemple.

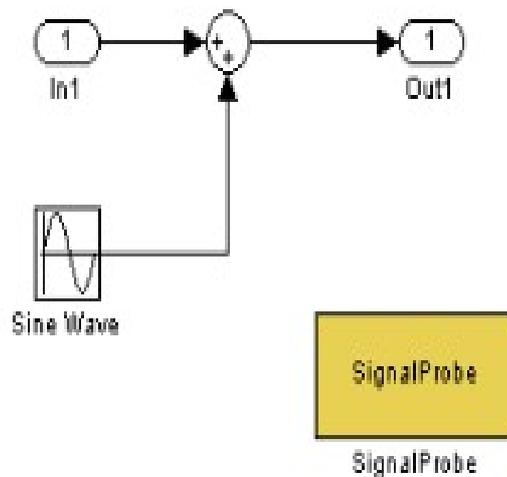


Figure 2

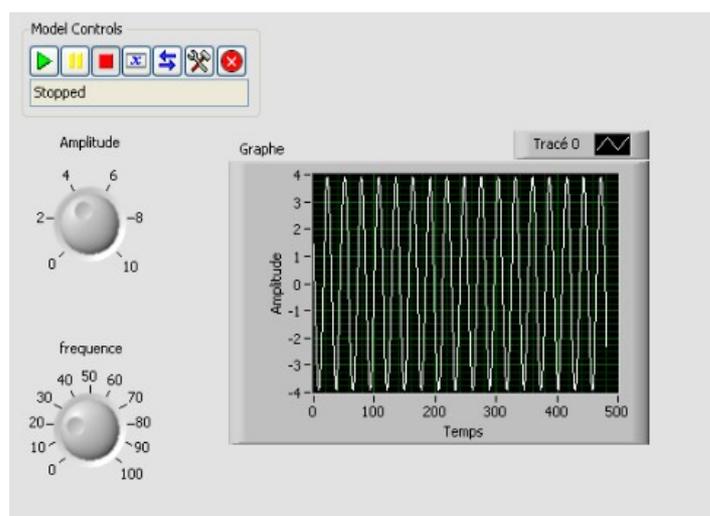


Figure 3

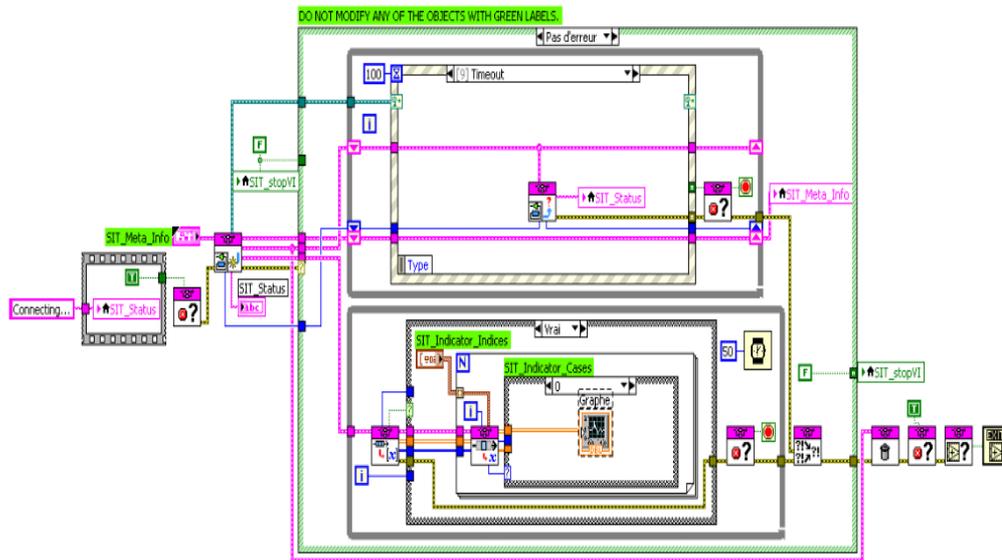


Figure 4

Conclusion:

En simulation, c'est à dire sans contact avec le système hydraulique réel en temps réel cette solution peut marcher.

Au fur et à mesure de notre avancement dans le projet et des difficultés rencontrées, nous sommes parvenus à trois solutions envisageables utilisant le SIT: La première est de garder le modèle donné tel qu'il est et de générer le fichier DLL comme dans l'exemple de la sinusoïde puis faire un mapping entre notre interface Labview et ce dernier. La deuxième est de partager le modèle Simulink initiale en deux sous-modèles; un qui s'occupe de l'acquisition temps réel et l'autre du lien avec l'interface Labview. La troisième est de récupérer les acquisitions et de générer la commande de la pompe par l'environnement Labview ensuite les envoyer au modèle Simulink qui lui va utiliser les solveurs puis récupérer cette dernière dans l'interface Labview.

Solution 1:

Comme nous l'avons précisé avant, cette solution consistait à une simple compilation du modèle Simulink en un fichier DLL. Cette méthode n'a pu aboutir car la compilation n'a pas pu être faite et un message d'erreur est affiché;

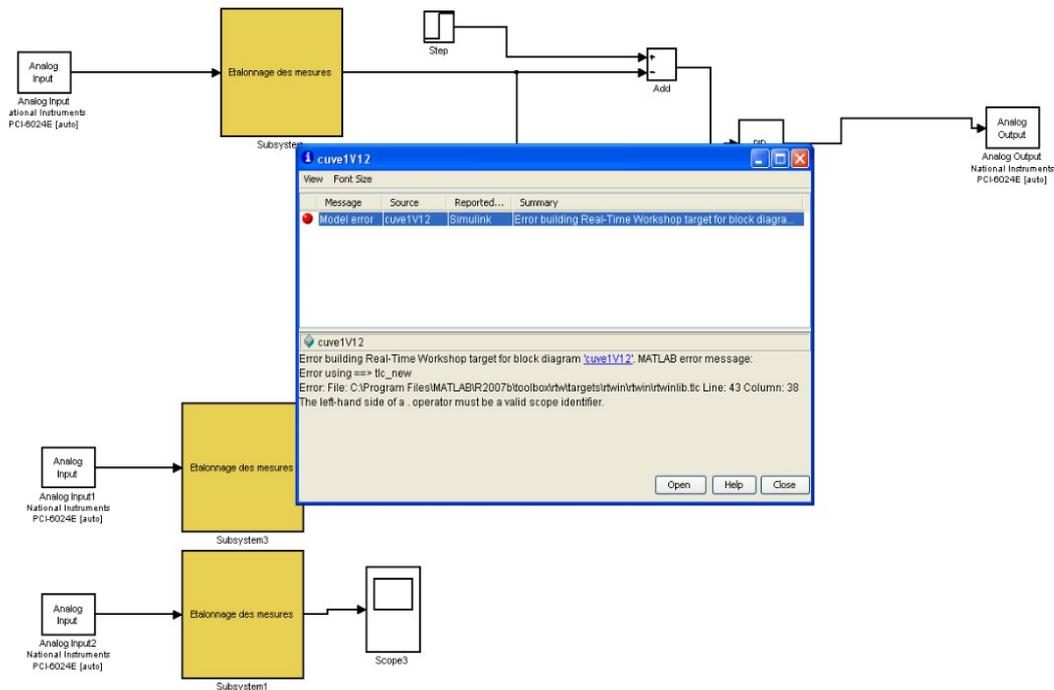


Figure 5

Le message ne précise pas la source d'erreur ce qui nous a poussé à effectuer une série de tests qui consistent à isoler chaque bloc du modèle et de le compiler à part. Nous arrivons à identifier les blocs posant le problème qui sont les entrées et sorties analogiques car en les remplaçant par des échelons le modèle est correctement compilé.

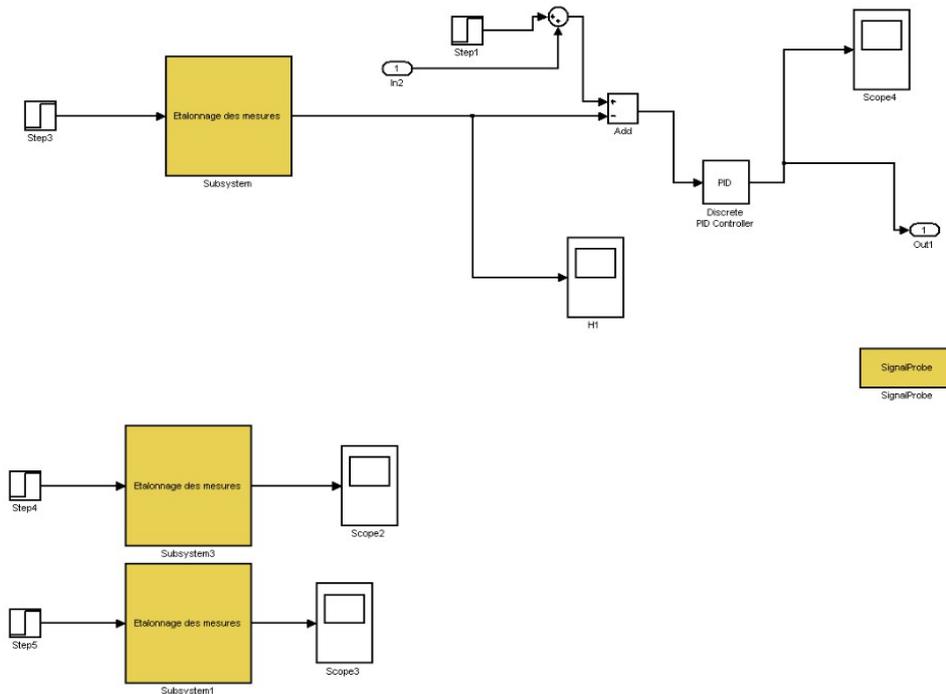


Figure 6

L'hypothèse intuitive est que les pilotes des entrées/sorties analogiques ne sont pas correctement installés ou mises à jour. Nous avons contacté le support Matlab France et National Instrument et d'après eux tout est bien installé.

En se référant aux principes de fonctionnement de la RTW et du Module NIDLL (SIT), nous avons trouvé qu'il est impossible d'utiliser les deux en même temps pour le même modèle. La RTW crée elle aussi un fichier C à partir du modèle Simulink qui est implémenté dans la carte d'acquisition, et lorsque le modèle est en fonctionnement temps réel les configurations de la simulation sont changées pour fonctionner avec la RTW or ces configurations ne sont pas bonnes pour fonctionner avec le SIT.

Cette solution ne peut marcher.

Solution 2

Pour résoudre les problèmes de la solution précédente, il nous a paru plus judicieux d'utiliser d'abord deux modèles Simulink en communication entre eux. Le premier ne contient que les entrées/sorties analogiques et ne

s'occupe que de l'acquisition des données temps réel puis de les envoyer au deuxième modèle qui lui est configuré pour le SIT et la communication avec Labview.

Les deux modèles sont faits et chacun peut fonctionner à part mais Matlab ne propose aucune fonction pour envoyer en temps réel les données d'un modèle vers un autre. Le système ainsi créé n'est pas en temps réel.

Solution 3

La dernière solution que nous proposons est comme suit:

Vu que la carte d'acquisition utilisée est une carte de National Instrument, le même concepteur de Labview, il existerait nécessairement des utilitaires pour pouvoir récupérer directement sur Labview les données acquises.

On récupérera donc les informations des capteurs et on actionnera la pompe directement de l'interface Labview et le modèle Simulink ne contiendrait donc que le PID.

En premier temps, nous avons fait une modélisation d'un des deux réservoirs en effectuant une identification, la méthode est plus détaillée dans la deuxième partie du rapport, puis on a essayé d'élaborer l'interface Labview commandant le système dans Simulink.

La compilation se fait correctement mais au niveau de Labview lors du mapping des entrées/sorties, le signal à la sortie du PID n'est pas considéré comme une entité complète mais comme trois signaux subissant trois corrections : le proportionnel, l'intégral et le dérivateur, et par conséquent on ne peut pas récupérer le signal corrigé complet mais trois signaux dont chacun représente l'action d'une seule partie du PID.

La solution est donc de remplacer ce correcteur par un deuxième qui fournit un signal unique. Nous avons le correcteur Floué.

L'interface est donc réalisée et on peut commander le modèle à partir de l'interface de Labview et récupérer la forme d'onde de sortie (le niveau de

T1) dans le graphe déroulant de ce dernier.

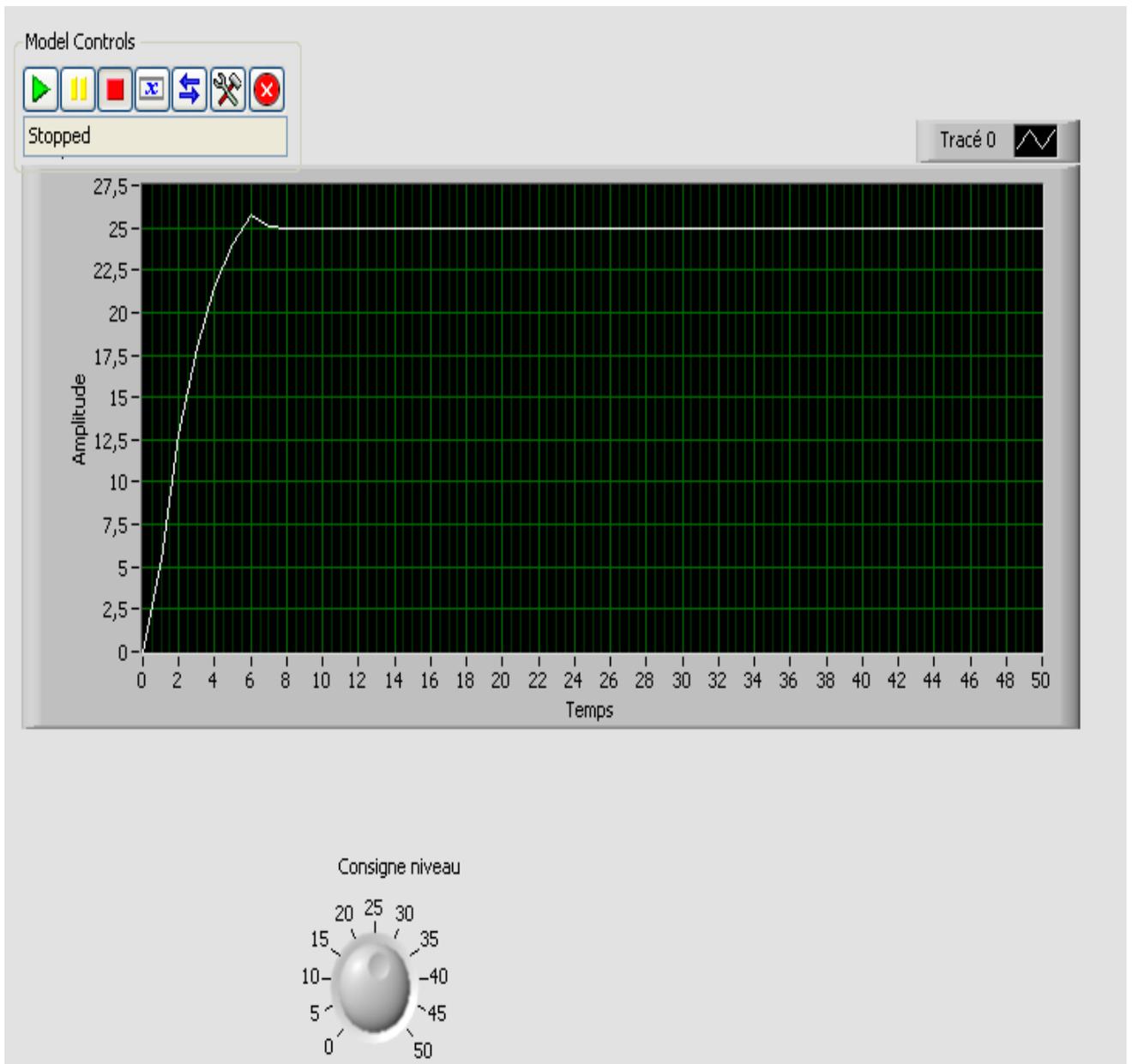


Figure 7

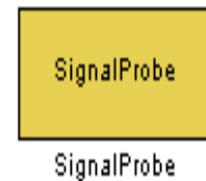
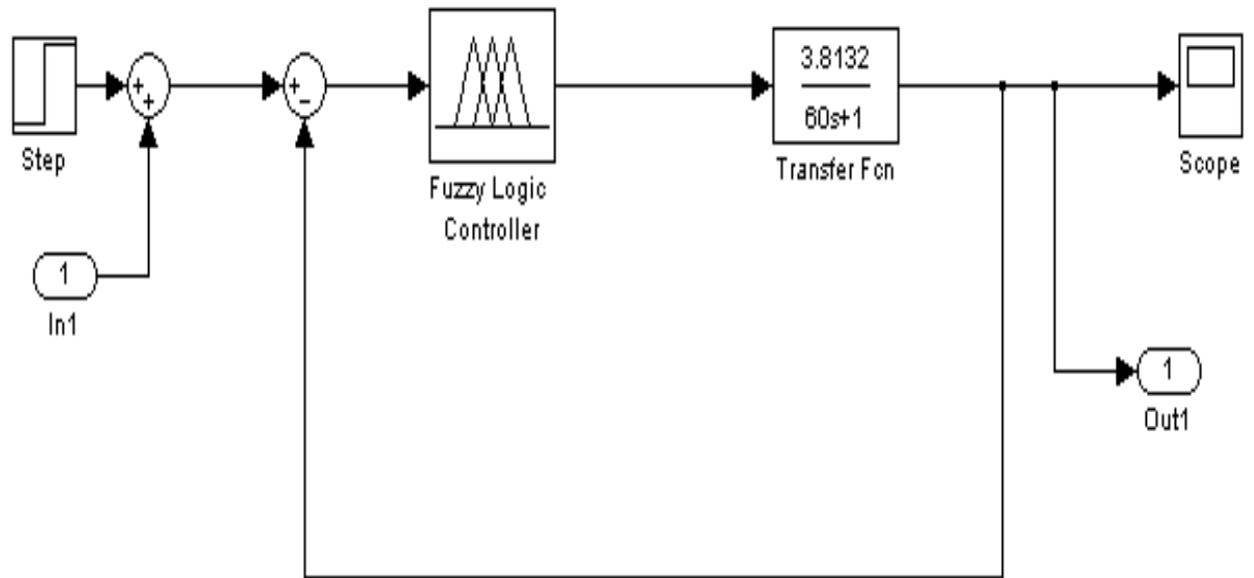


Figure 8

Récupération des données sur Labview

National Instrument met à disposition l'utilitaire assistant DAQ (Data Acquisition) pour gérer des entrées/sorties via différentes cartes d'acquisition. Dans notre cas il s'agit de la PCI-2345. Nous avons créé deux Assistants- DAQ, une pour actionner la pompe et la deuxième pour acquérir le niveau du premier réservoir en temps réel. Vous trouverez en annexe la méthode à suivre pour la configuration des DAQ.

Résultats:

On arrive parfaitement à commander la pompe en envoyant des signaux compris entre 0mA et 20mA, et à récupérer en même temps la valeur du niveau dans le premier réservoir.

D'un autre côté faisant le mapping entre cette interface et le modèle Simulink sans les entrées/sorties analogiques, la compilation se fait correctement mais la simulation s'arrête en affichant un message d'erreur: On ne peut pas réaliser ce genre de manipulations avec une version d'évaluation.

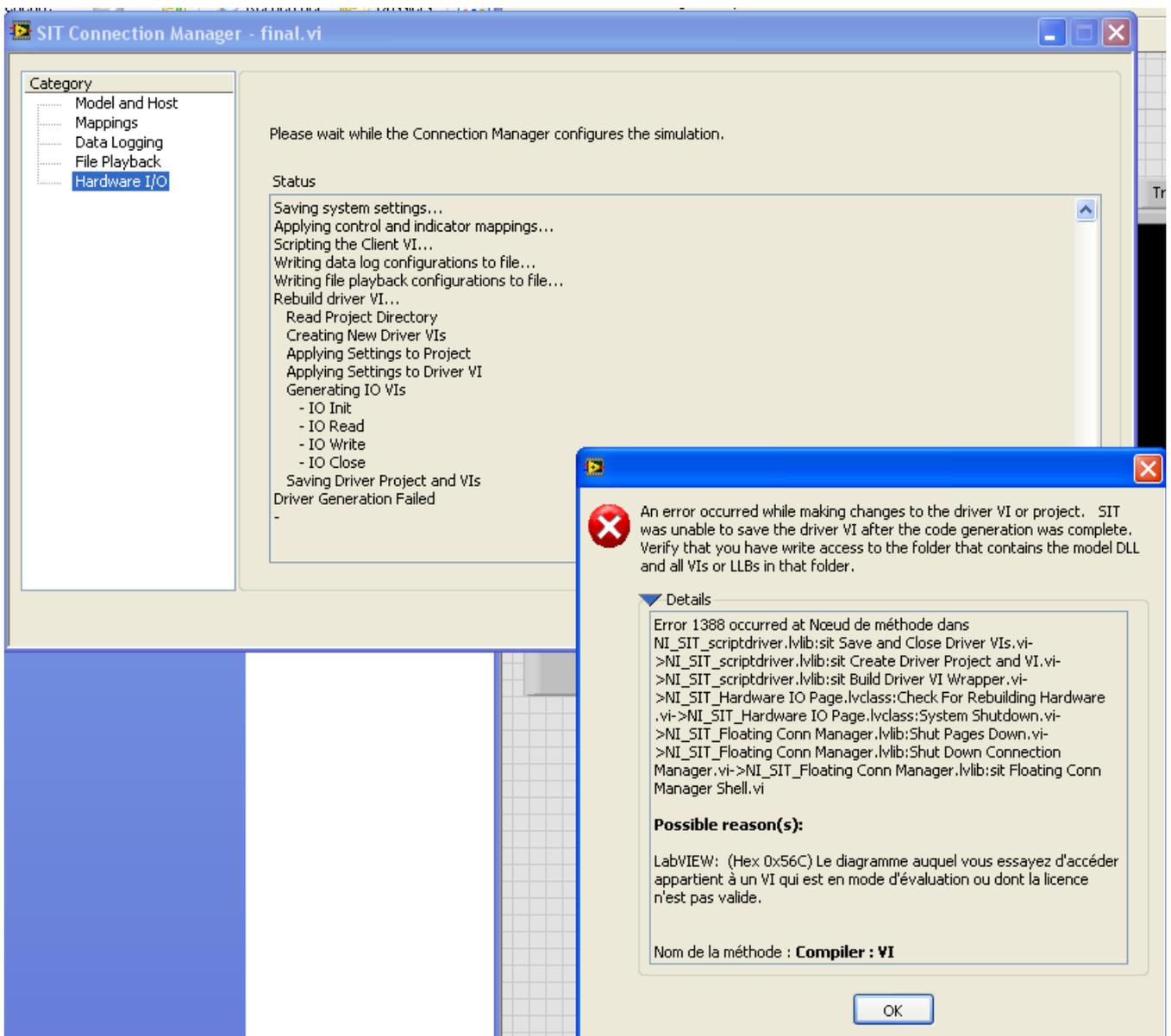


Figure 9

III. Commandes du système

1. Limites du PID utilisé

Le PID est l'un des correcteurs les plus utilisés dans le monde industriel grâce à sa simplicité et à ses résultats assez satisfaisants. Dans notre système la régulation en niveau se fait avec un PID qui présente une réponse sans écart statique avec un temps de réponse assez court et sans dépassement. Cependant, en introduisant un retard grâce au serpentin mise en place à l'entrée du réservoir T1, le PID présente un grand dépassement et le système devient instable. Le dépassement est dans la plupart des cas, notamment dans les systèmes en contact direct avec les utilisateurs, est intolérable.

Pour résoudre ce problème nous avons choisi la commande Prédicateur de Smith. D'autre part, la configuration d'un PID nécessite un calcul précis et assez compliqué, alors que dans certains, notamment celui des systèmes lents, une commande simple qui atteint les consignes est plus désirable, d'où la commande Floue.

2. Prédicateur de Smith

Identification de la fonction de transfert

Il est nécessaire de trouver un modèle mathématique pour pouvoir tester la commande en simulation avant de l'implémenter dans le système réel. Pour cela on a identifié la fonction de transfert en boucle ouverte du système suivant le modèle ci dessous:

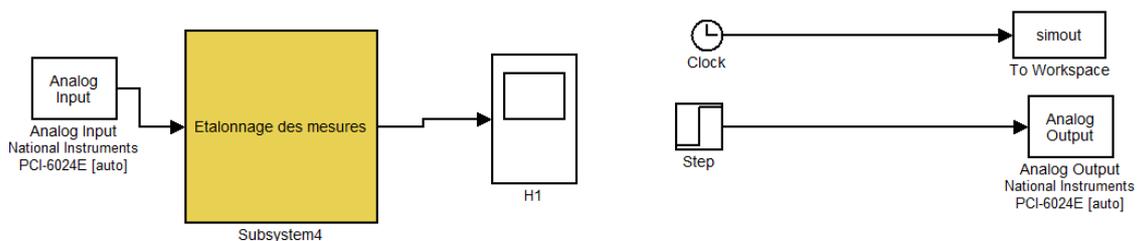


Figure 10

On retrouve la réponse temporelle suivante:

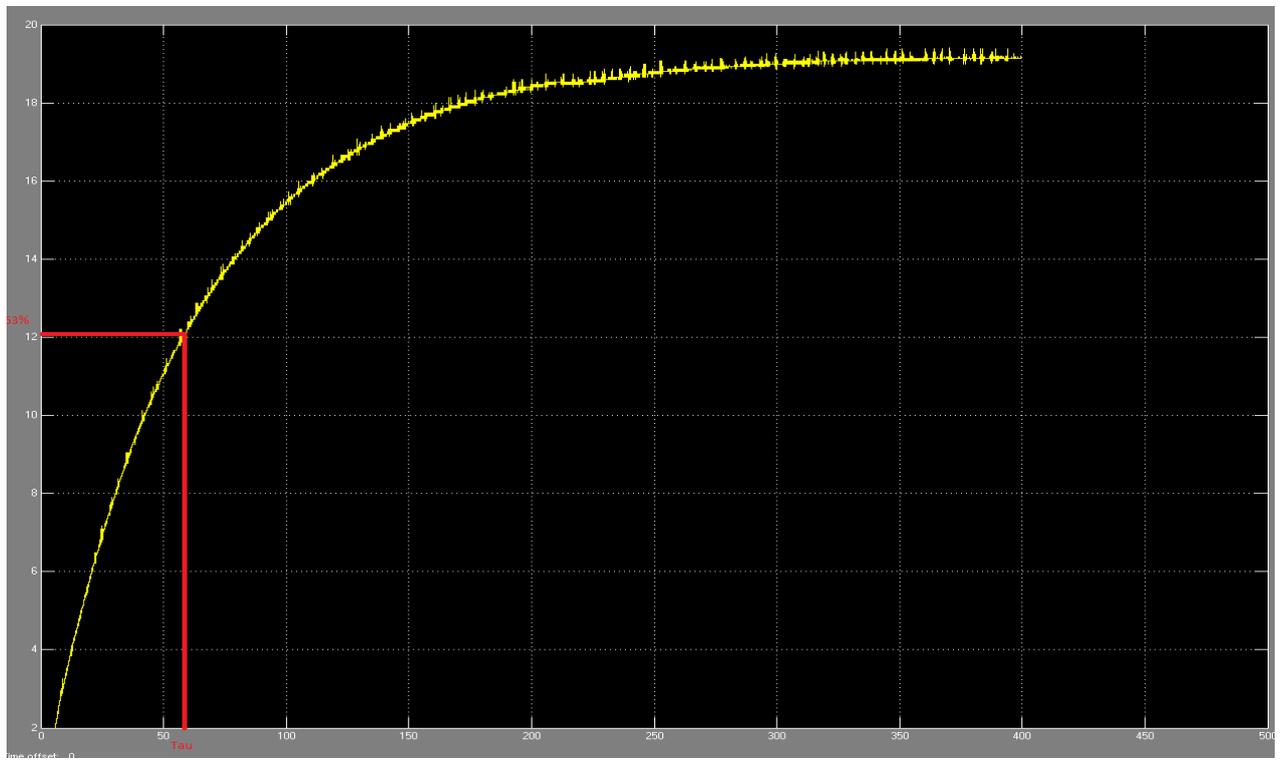


Figure 11

Il s'agit d'un système de première ordre, après l'action d'un PID on voit bien que le système est instable avec un grand dépassement si on introduit un retard. Après la commande de Smith le système devient stable avec un léger dépassement. Ci dessous les schémas blocs avec les réponses:

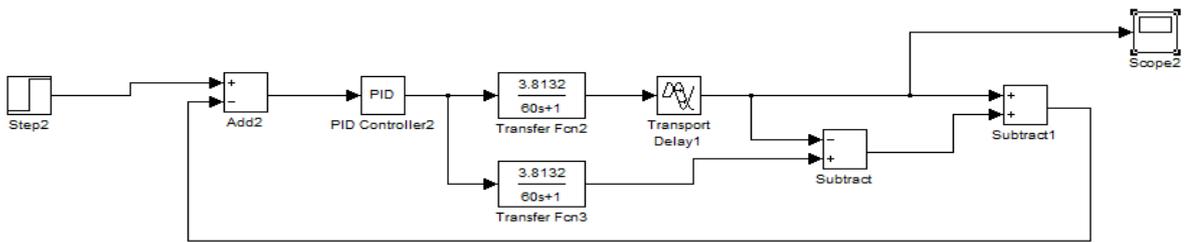
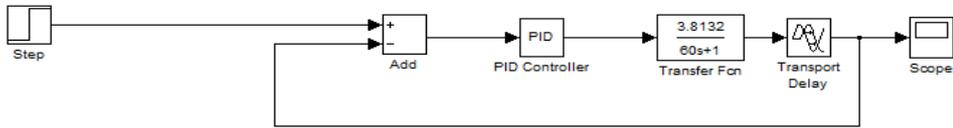
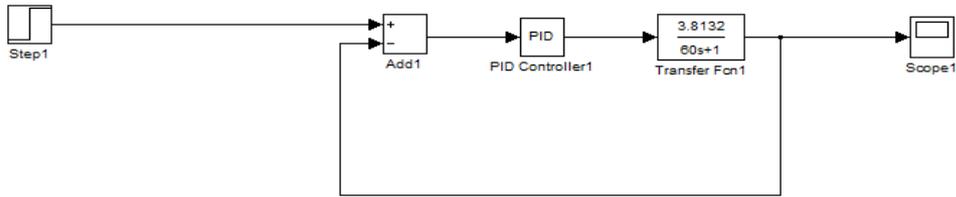


Figure 12

Résultats :

La réponse avec un PID sans retard

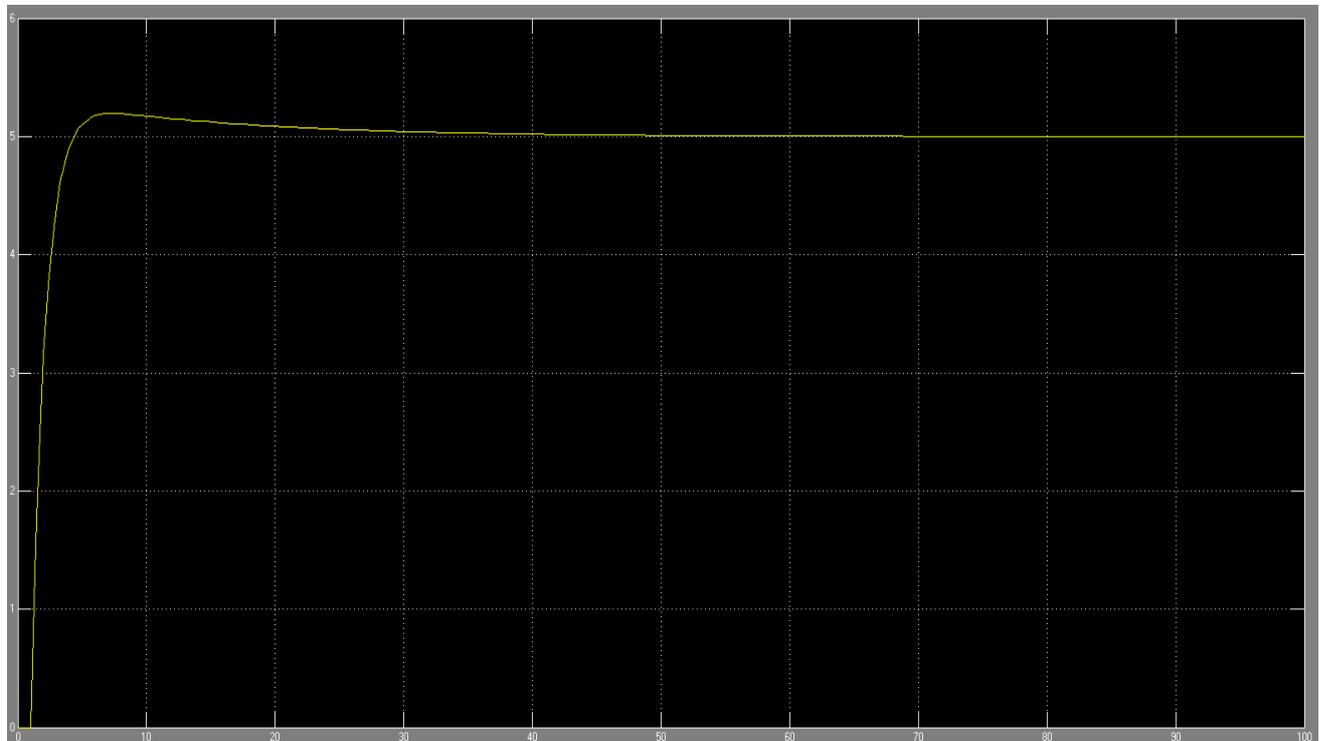


Figure 12

La réponse avec un PID avec retard

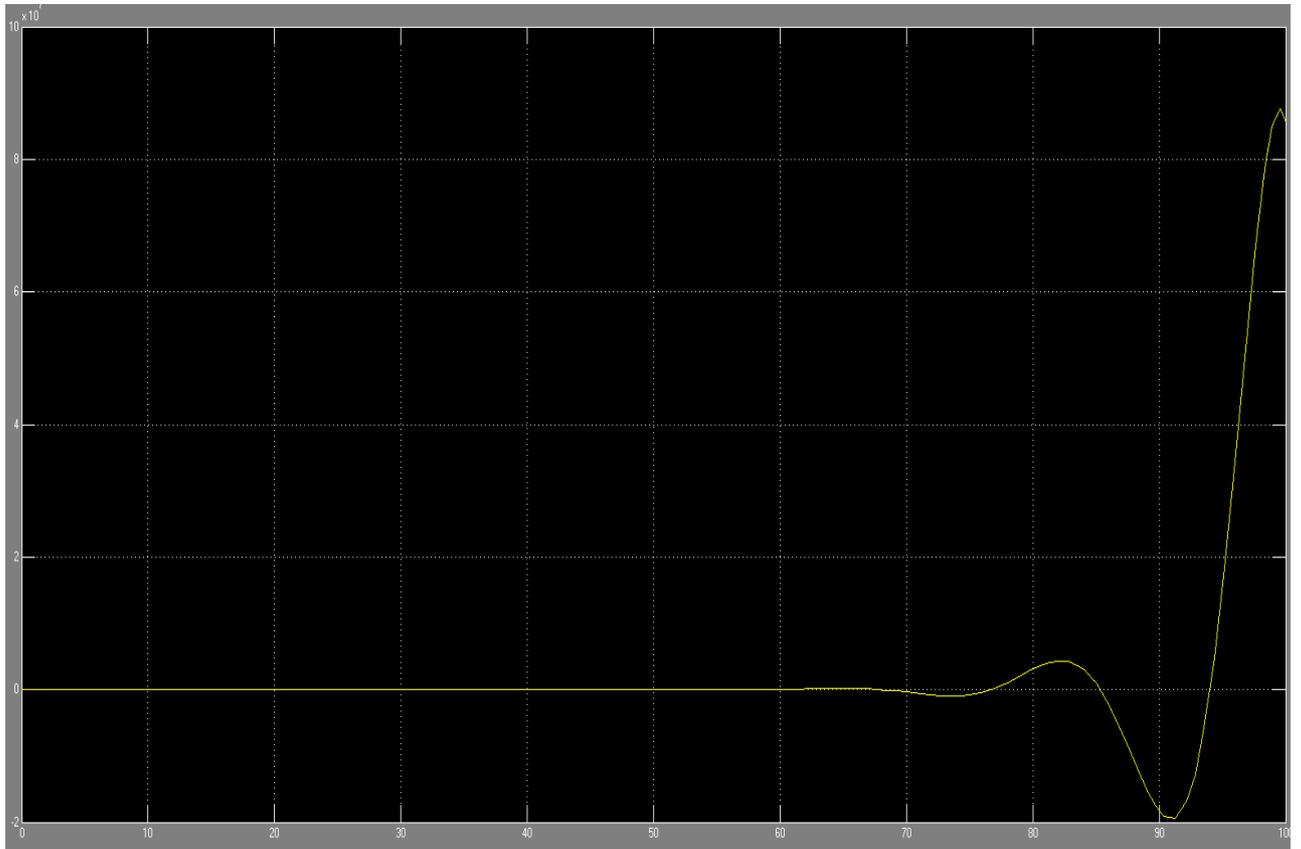


Figure 13

On voit bien que le système devient instable. Avec des dépassement trop importants.

Réponse avec un prédicteur de Smith

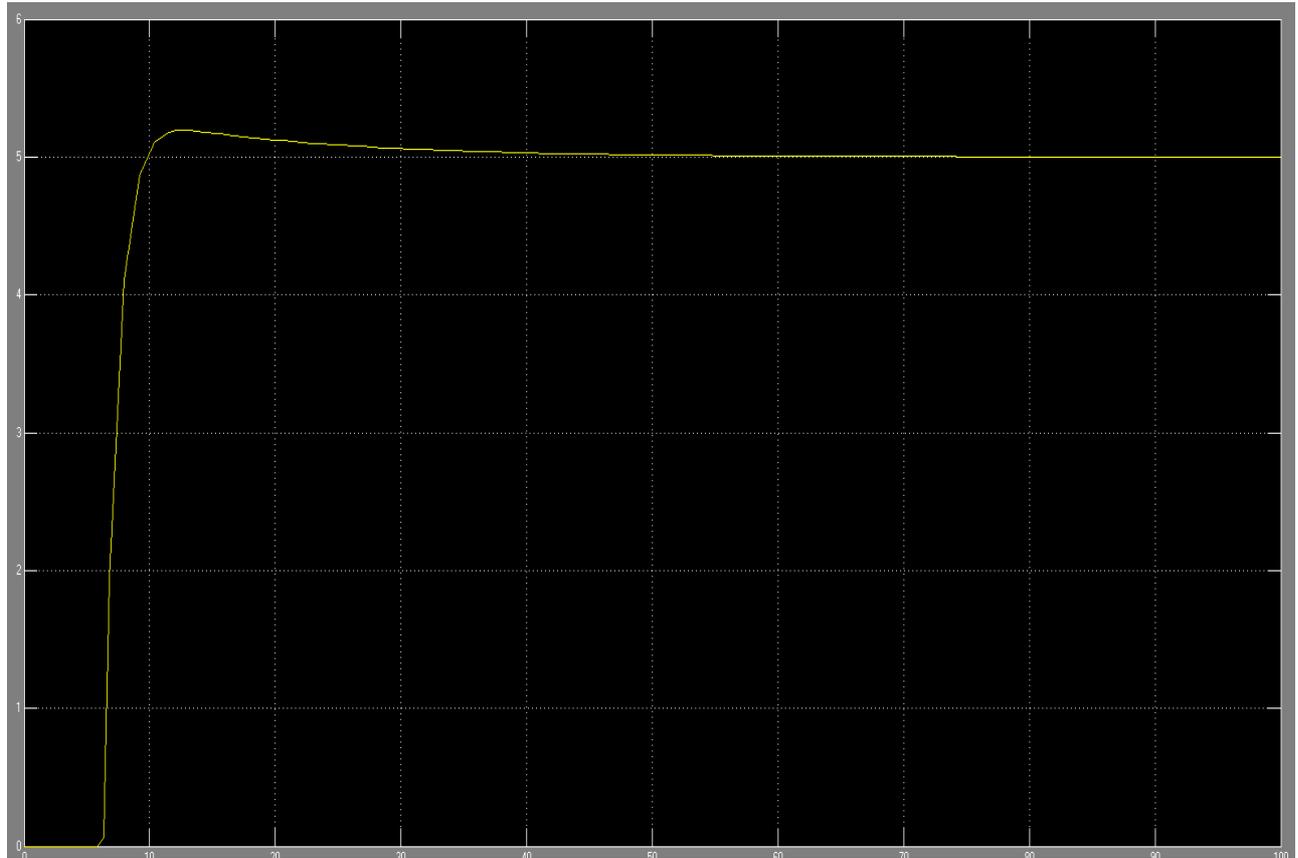


Figure 14

Le système devient stable avec un léger dépassement.

3. La commande Floue

Description de la logique Floue:

La logique Floue permet de définir une appartenance graduelle d'un élément à une classe (appartient + ou – fortement à cette classe): cette appartenance est définie par un degré d'appartenance entre 0 et 1.

Application au système hydraulique:

L'application de cette commande à notre système donne des résultats assez satisfaisant mais avec un temps de réponse assez grand.

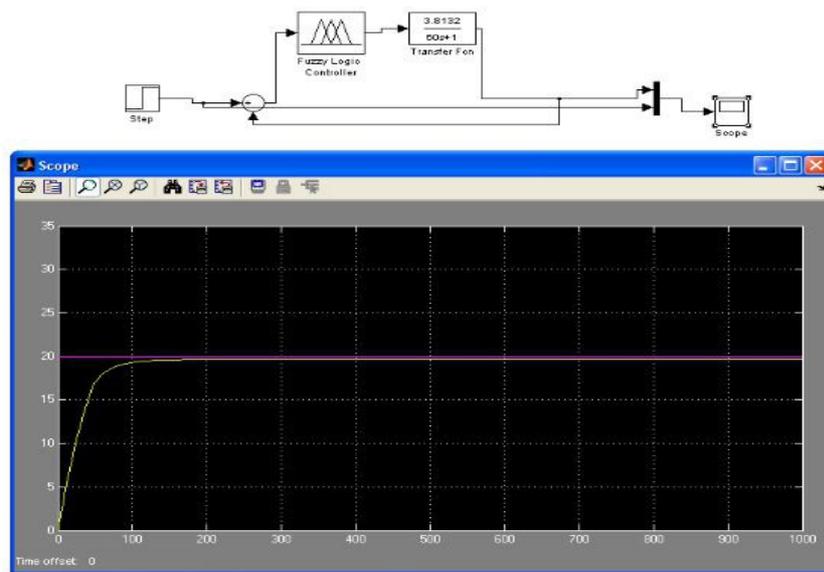


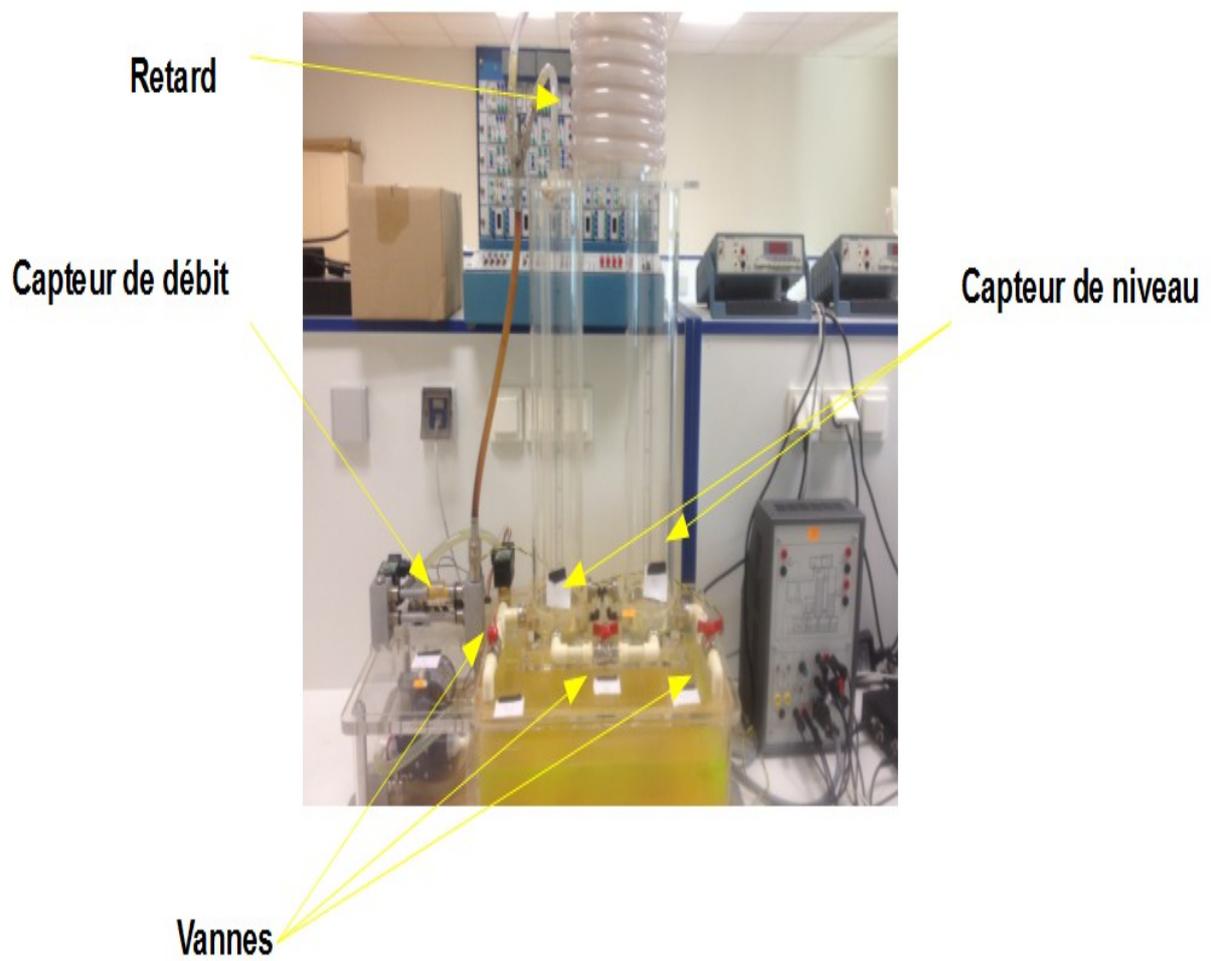
Figure 15

IV. Conclusion

Ce projet nous a permis d'appliquer et de bien assimiler les différentes commandes avancées vues en cours d'ingénierie des systèmes automatisés. D'autre part, on a acquis une bonne maîtrise du logiciel de simulation et de calcul formel Matlab/Simulink ainsi qu'une très bonne culture du logiciel Labview. Ce projet aussi nous a permis d'entrer en contact avec des ingénieurs de National Instrument avec lesquels on a eu un contact continu et on a entretenu de nombreux entretiens sur place.

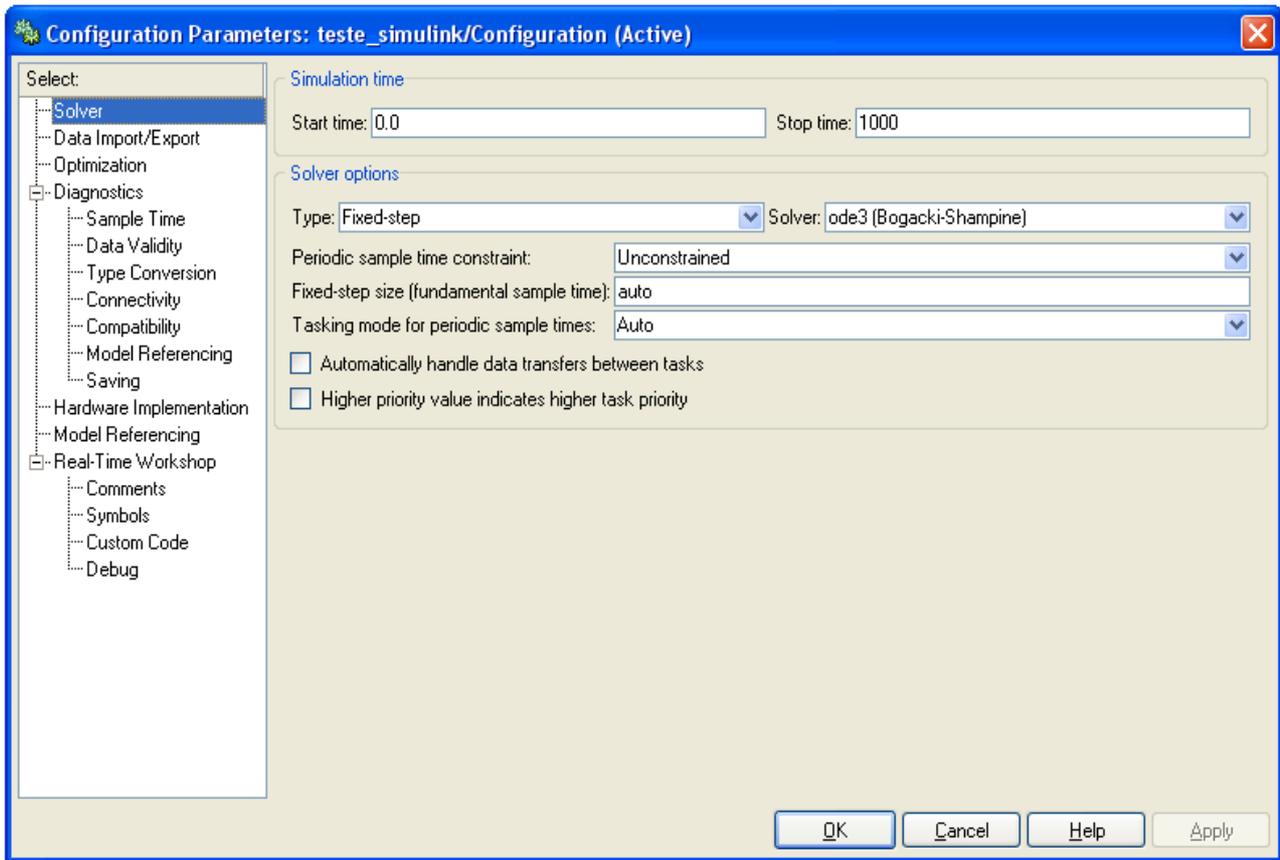
V. ANNEXES

Composants du système hydraulique :



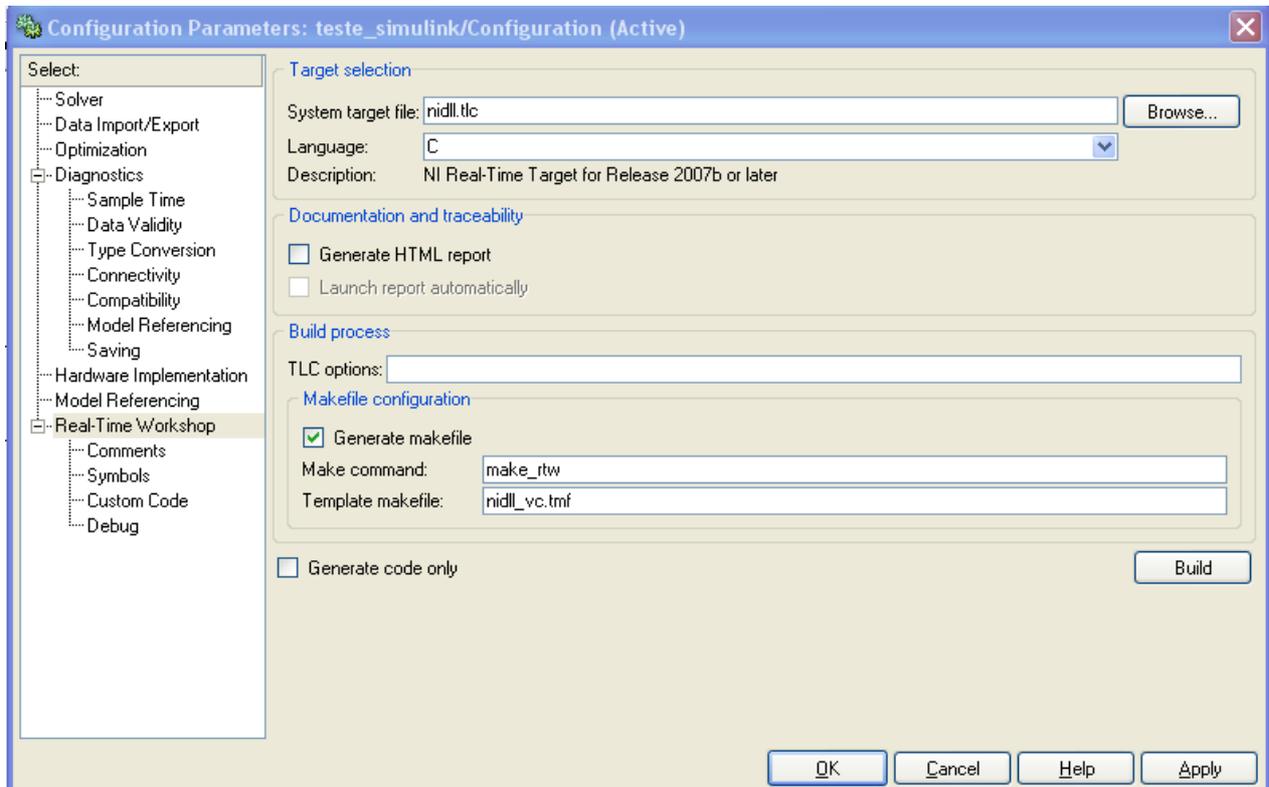
Annexe 1

Configuration de Matlab/Simulink pour le SIT



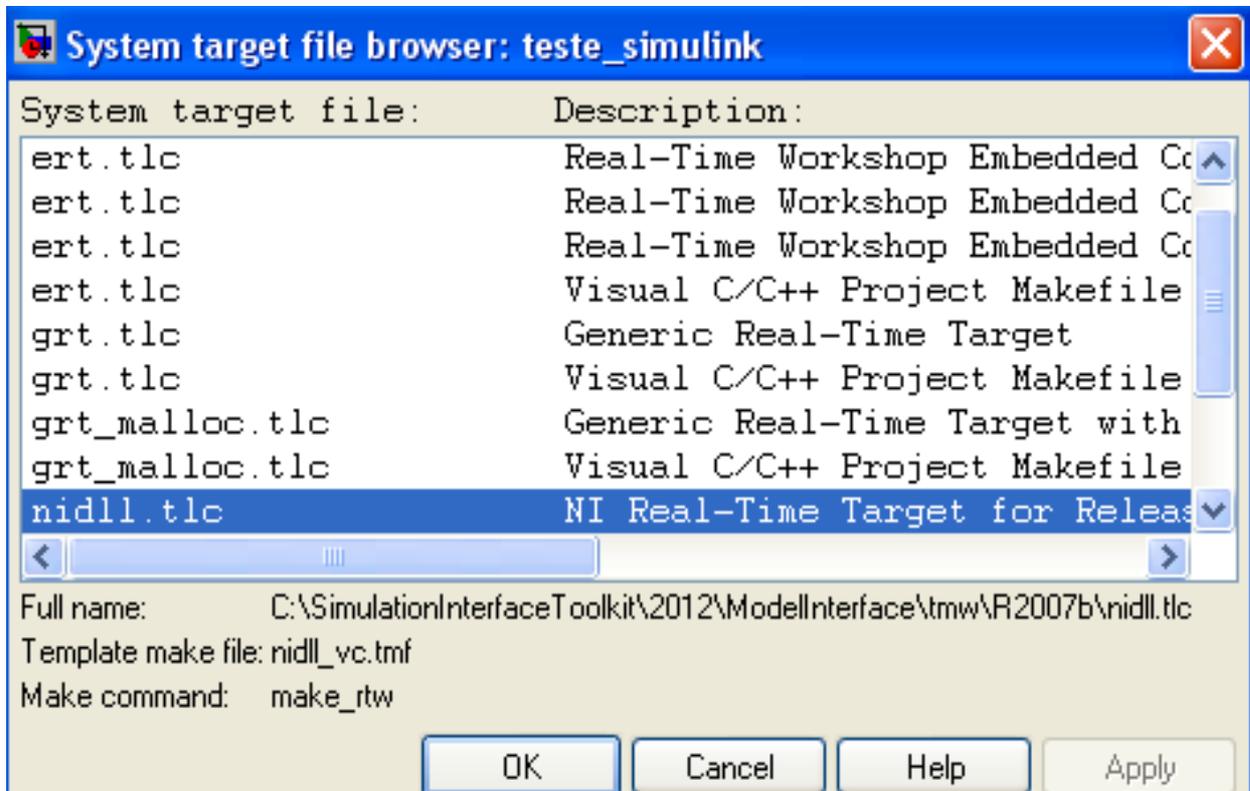
Annexe 2

Simulation --- Configuration Paramètre---- Solver----Type--Fixed Step.



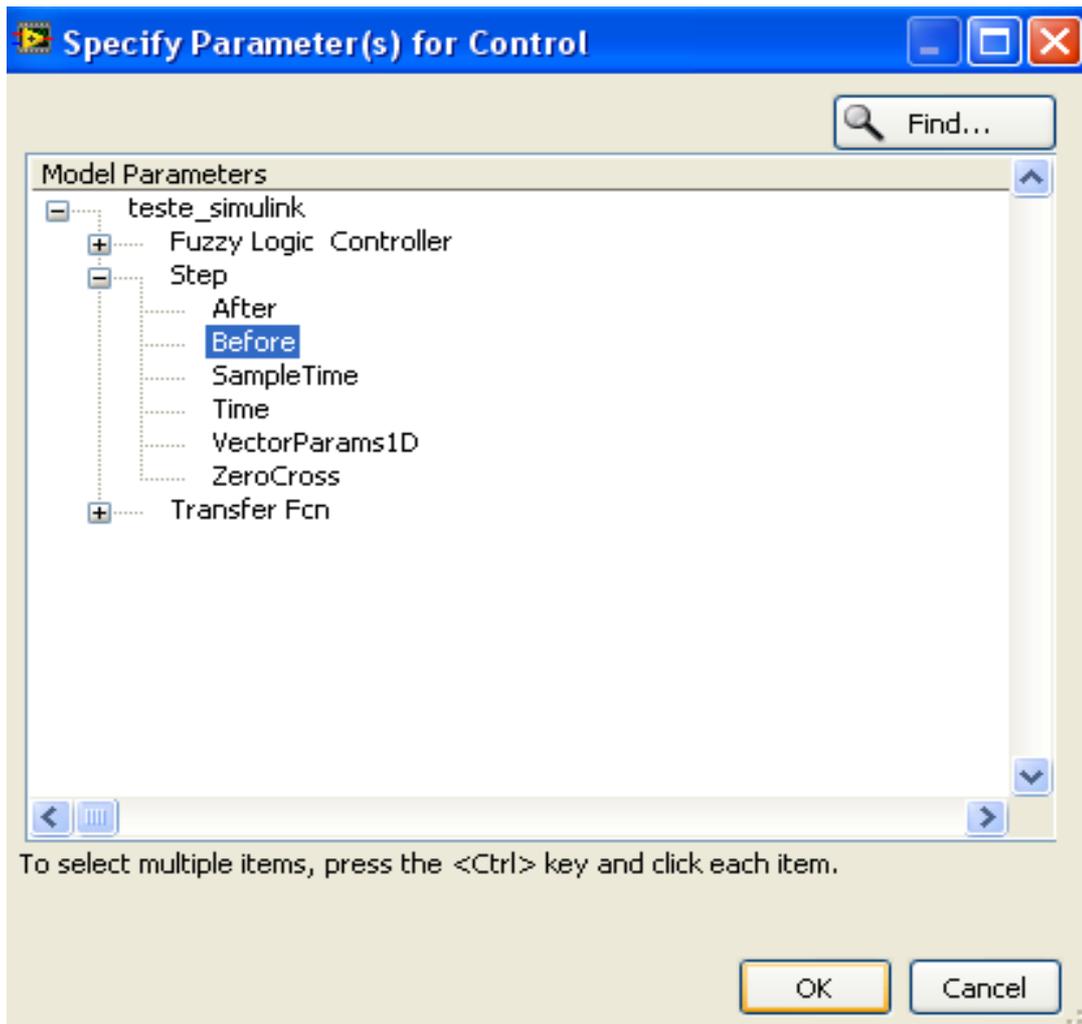
Annexe 3

Real Time Workshop -system target file browse ---- NIDLL.tlc



Annexe 4

Pour compiler, Ctrl+B



Annexe 7