

IDENTIFICATION DE SYSTÈMES ET CONCEPTION DE SYSTÈMES DE CONTRÔLE AVEC SIGNALEXPRESS

Amèle SALAH, Philippe BAUCOUR

National Instruments France
Centre d'Affaires Paris Nord – BP 217
93153-Le Blanc-Mesnil
amele.salah@ni.com, philippe.baucour@ni.com

Résumé : En moins de trente ans, l'Instrumentation Virtuelle a révolutionné la façon dont les produits manufacturés sont testés. Depuis quelques années, on observe une évolution "naturelle" de cette technologie dans les applications de contrôle temps réel. Dorénavant, le troisième axe de développement consiste à étendre l'usage de l'Instrumentation Virtuelle tout au long du cycle de développement des produits : modélisation, conception, simulation, prototypage, HIL... Nous présentons ici de manière pratique les produits NI susceptibles d'être mis en œuvre dans les phases d'identification de systèmes et de conception de systèmes de contrôle.

Mots-clés : Instrumentation Virtuelle, SignalExpress, conception, systèmes, contrôle, simulation, validation.

1 INTRODUCTION

En moins de trente ans, l'Instrumentation Virtuelle a révolutionné la façon dont les produits manufacturés sont testés. Du point de vue des industriels, la preuve est faite : "l'Instrumentation Virtuelle qui associe autour du PC des moyens d'entrée/sortie et du logiciel afin de concevoir des systèmes souples d'emploi et définis par l'utilisateur, ça marche et c'est rentable."

Ceci dit, tester les produits sur les chaînes de production est une chose mais optimiser le processus de création dans son ensemble en est une autre.

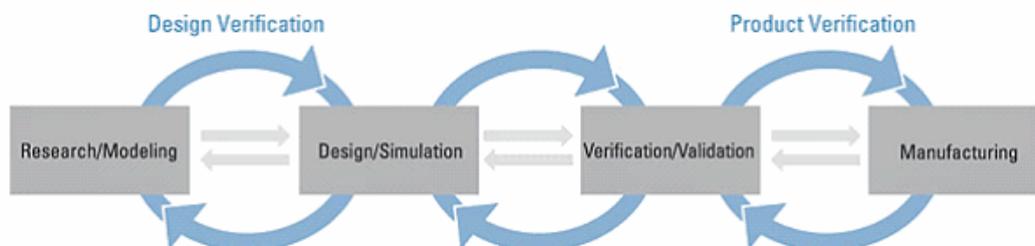


Figure n°1: les quatre phases par lesquelles passent les produits.

Un des moyens pour y parvenir consiste à faire en sorte que les différents types d'ingénieurs qui interviennent (R&D, conception, validation, production...) tout au long du cycle de développement-production s'appuient sur un socle technologique commun. Ce faisant, le

dialogue pourra alors s'établir de manière efficace et par exemple, l'ingénieur de conception pourra comprendre les requêtes de ses collègues de la production, partager du code et des moyens de test, capitaliser le savoir-faire, etc.

Nous sommes persuadés que les outils d'Instrumentation Virtuelle peuvent constituer une plate-forme sur laquelle les industriels peuvent capitaliser afin de mettre sur le marché plus efficacement des produits toujours mieux conçus.

La figure ci-dessous résume les axes de développement des outils d'Instrumentation Virtuelle.

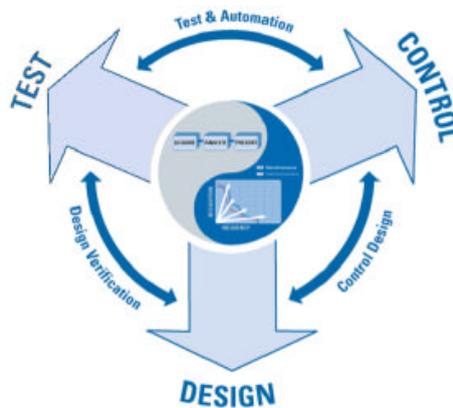


Figure n°2 : les 3 axes de développement de l'Instrumentation Virtuelle

Au centre, on retrouve le "Ying-Yang", c'est-à-dire l'association matériels-logiciels qui représente l'Instrumentation Virtuelle. Pour le reste, les trois axes de développement : test de production, contrôle/commande (application temps réel) et conception.

Nous allons dans la suite de cet article focaliser notre attention sur la partie conception. Plus précisément, nous allons montrer, étape par étape, quels sont les outils d'Instrumentation Virtuelle susceptibles d'être utilisés afin de concevoir le système d'asservissement d'un moteur à courant continu.

Bien sûr, il s'agit d'un prétexte mais nous aurons ainsi l'occasion de réaliser qu'aujourd'hui, il est possible de mettre dans les mains des non-programmeurs des outils d'Instrumentation Virtuelle et qu'il existe de réelles opportunités pour concevoir et manufacturer les produits plus efficacement.

2 CONFIGURATION MATÉRIELLE ET LOGICIELLE

Du point de vue matériel, la configuration est on ne peut plus classique : un PC de bureau et une carte d'acquisition. Le moteur à courant continu que nous allons asservir est intégré sur une platine de la société Quanser (quanser.com). Sur cette dernière, on retrouve une entrée analogique (qui contrôle la vitesse de rotation du moteur) ainsi qu'une sortie tachymétrique qui permet de mesurer la vitesse effective du moteur.



Figure n°3: la configuration matérielle utilisée

Concernant l'aspect logiciel, nous utiliserons l'application SignalExpress (ni.com/france/signalexpress) ainsi que deux bibliothèques complémentaires : le LabVIEW System Identification Toolkit et le LabVIEW Control Design Toolkit. À la fin, lorsqu'il s'agira de faire la simulation complète de l'ensemble régulateur-système, nous serons amenés à utiliser l'environnement de développement graphique LabVIEW et le LabVIEW Simulation Module.

Il est intéressant de noter que les deux premières bibliothèques que nous avons mentionnées (System ID et Control Design) fonctionnent non seulement avec LabVIEW mais s'intègrent directement à l'application SignalExpress si cette dernière est détectée lors de leur installation. Notons pour finir qu'il existe, comme pour LabVIEW, une version de démonstration gratuite de l'application SignalExpress.

3 IDENTIFICATION DU SYSTÈME

L'objectif ici est de déterminer la fonction de transfert du système que l'on souhaite asservir (le moteur à courant continu).

Typiquement, en dehors des méthodes de calculs théoriques, pour identifier un système en boucle ouverte, il faut commencer par l'exciter avec un signal dont les niveaux de tension et le contenu fréquentiel correspondent à ses conditions de fonctionnement réel.

Différents types de signaux peuvent être utilisés pour identifier le système, tels que des signaux sinusoïdaux wobulés, des signaux carrés avec ou sans bruit, etc.

Dans un premier temps, il faut lancer l'application SignalExpress.

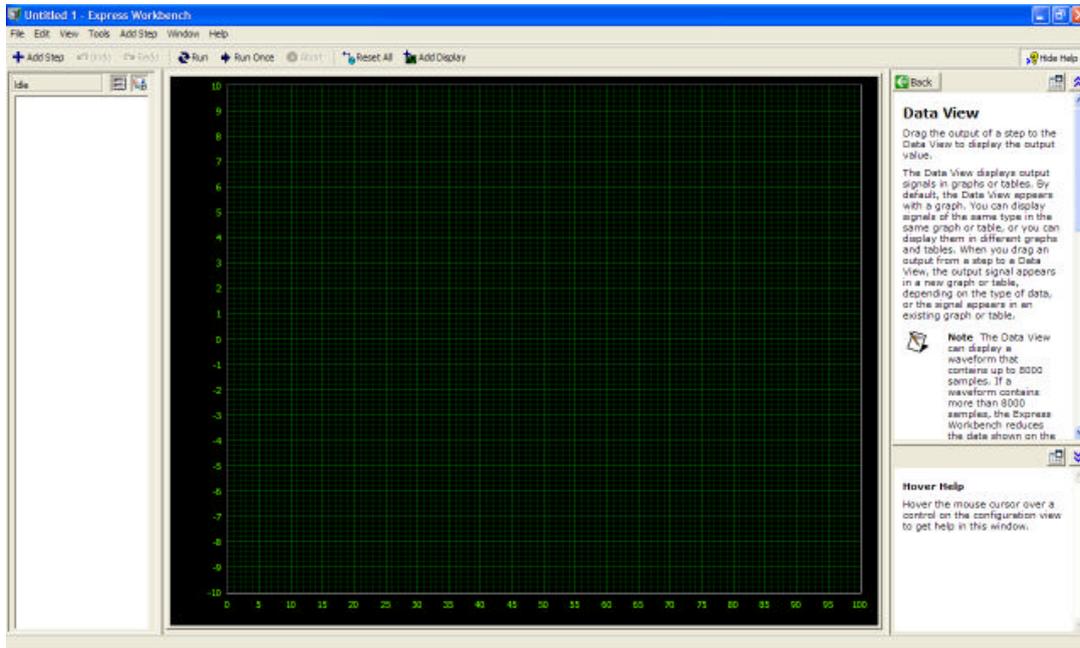


Figure n°4 : le tout premier lancement de SignalExpress

3.1 Création du stimulus

Nous avons choisi de "stimuler" le moteur avec un signal d'onde triangulaire de 3 V d'amplitude. Dans SignalExpress, ce signal est créé en cliquant successivement sur "+ Add Step", "Signal Input/Output" puis sur ?Create Signal?. La fenêtre ci-dessous apparaît alors.

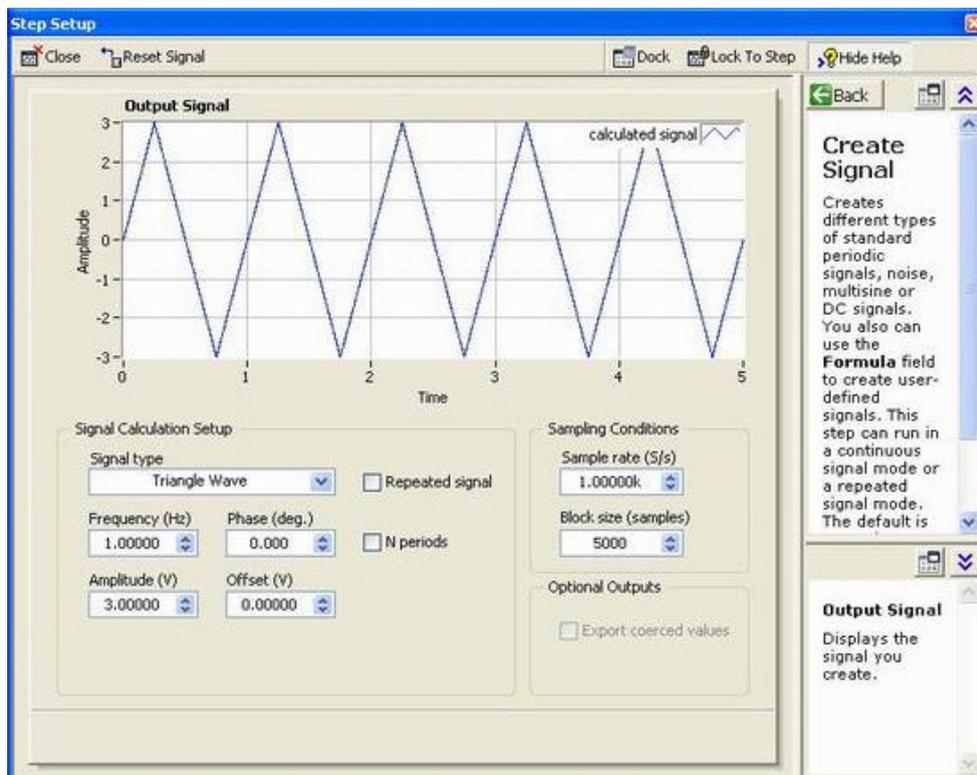


Figure n°5 : création d'une onde triangulaire de 3 V d'amplitude

Voici dans le tableau ci-dessous où trouver ce pas (Step) et les valeurs à utiliser :

Step	Valeurs/ Actions
Signal Input/Output Create Signal	Signal Type = Triangle Wave
	Frequency = 1 Hz
	Amplitude = 3 V
	Sample Rate = 1kS/s
	Block Size = 5000 samples

Une fois les paramètres saisis, il suffit de cliquer sur le bouton "Close" en haut à gauche de la boîte de dialogue.

De retour dans la fenêtre principale de SignalExpress, afin de visualiser le signal créé, faites simplement glisser le "calculated signal" de la séquence vers le graphe.

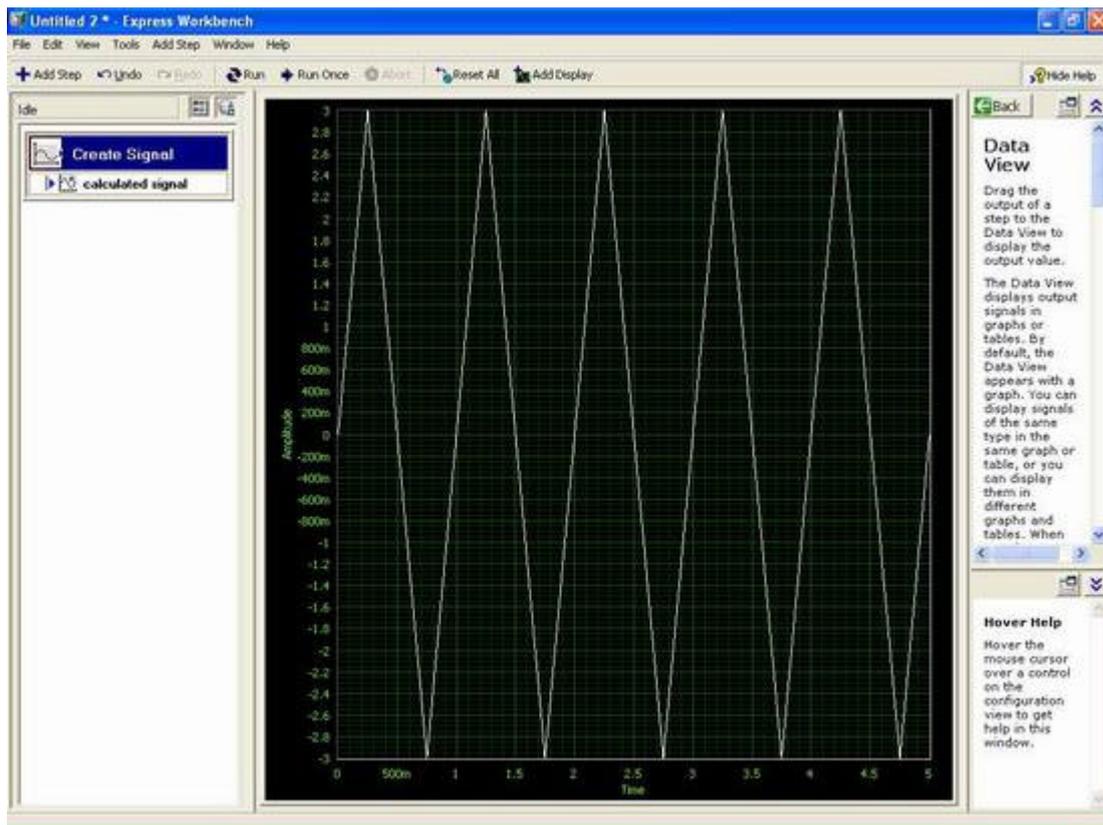


Figure n°6 : visualisation du signal créé dans la fenêtre d'affichage

À ce stade, il est utile de noter qu'avec SignalExpress vous n'êtes pas limité à quelques signaux standards. En effet, vous pourriez décrire le signal à l'aide d'une formule mathématique ou tout simplement aller le chercher dans un fichier.

3.2 Génération du stimulus sur la carte DAQ

Maintenant, il faut que le signal créé soit généré en tant que signal de sortie analogique (analog output) de la carte DAQ. Pour cela, ajoutez un Step (bouton "+ Add Step"), puis cliquez sur "Signal Input/Output", "Generate Signal" puis sur ?NI-DAQmx Generate?.

Vous retrouvez dans le tableau ci-dessous toutes les informations nécessaires pour remplir la boîte de dialogue qui apparaît :

Step	Valeurs / Actions
Signal Input/Output -> Generate Signals -> NI DAQmx Generate	Config Tab: Device: PXI1slot3(PXI-6281) Channels : ao0 Output range : ao0 max : 5V ao0 min : -5V

3.3 Acquisition de la réponse

Il est temps d'acquérir la mesure du tachymètre qui est fixé sur le moteur. Pour cela, nous allons utiliser le pas ?NI-DAQmx Acquire?.

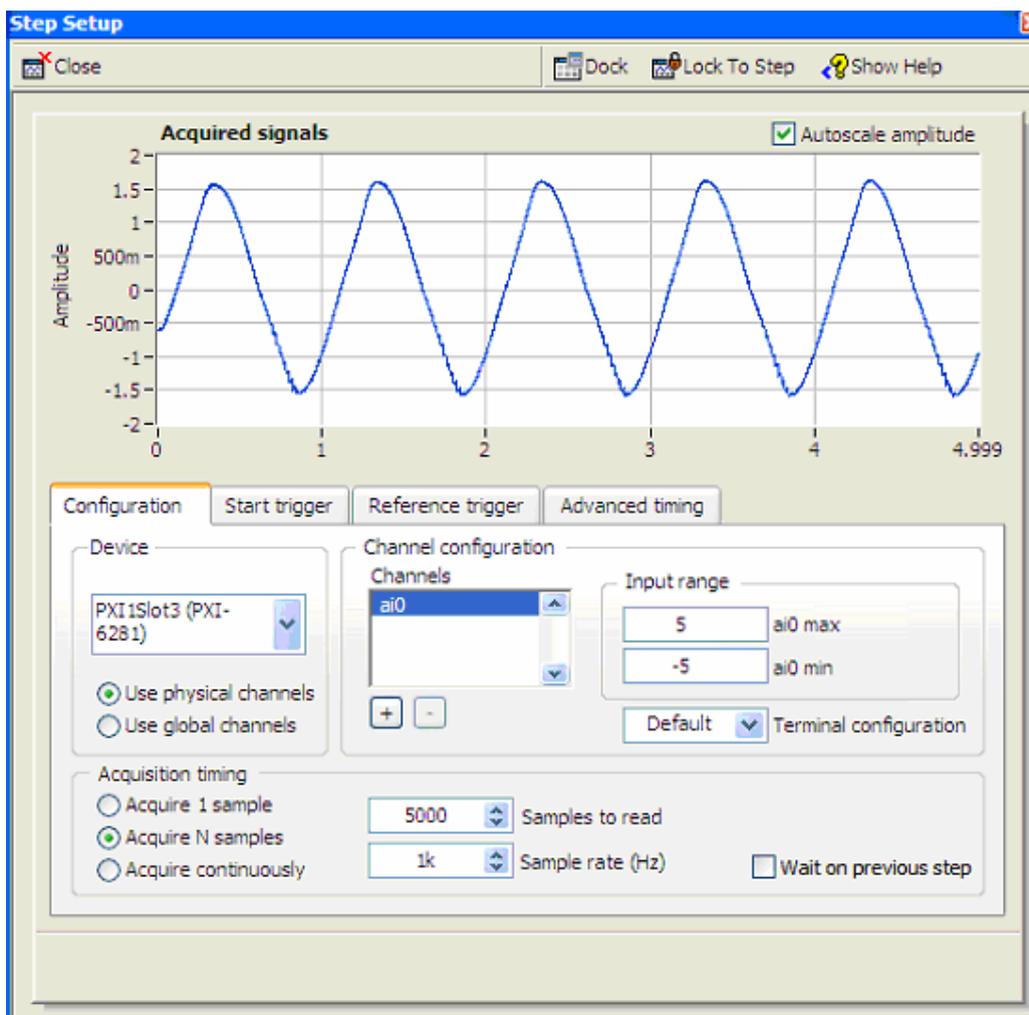


Figure n°7 : pas ?NI-DAQmx Acquire? pour acquérir la réponse du moteur en tant que signal analogique de la carte DAQ

Voici dans le tableau ci-dessous où trouver ce Step et les valeurs à utiliser :

Step	Valeurs / Actions
Signal Input/Output -> Acquire Signals -> NI DAQmx Acquire	Config Tab: Device: PXI1slot3(PXI-6281) Channels : ai0 Output range : ai0 max : 5V ai0 min : -5V
	Config Tab: Acq. Timing: 5000 samples to read
	Config Tab: Acq. Timing: 1 kHz sample Rate

De manière similaire à ce que nous avons déjà fait, afin de visualiser la réponse du moteur, faites glisser et déposez la sortie du pas "NI-DAQmx Acquire" sur la fenêtre d'affichage.

Un message vous indiquera alors que le pas actuel n'est pas relié aux données déjà à l'affichage. Un symbole de déconnexion (voir cercle rouge ci-dessous) apparaît également entre le pas ?NI-DAQmx Generate? et le pas ?NI-DAQmx Acquire?. Ce symbole disparaîtra dès que les pas de création et d'acquisition seront utilisés dans le processus d'identification du système.

Sélectionnez ?No? sur la boîte de dialogue pour créer une nouvelle fenêtre d'affichage afin de visualiser la réponse du moteur.

Cliquez sur le bouton ?Run Once? pour que l'enchaînement des pas s'exécute une fois et faire une acquisition de données pendant 5 s (acquisition de 5000 points à 1 Kéch./s).

La réponse du système à l'excitation par le signal triangulaire sera alors affichée dans la nouvelle fenêtre qui a été ajoutée.

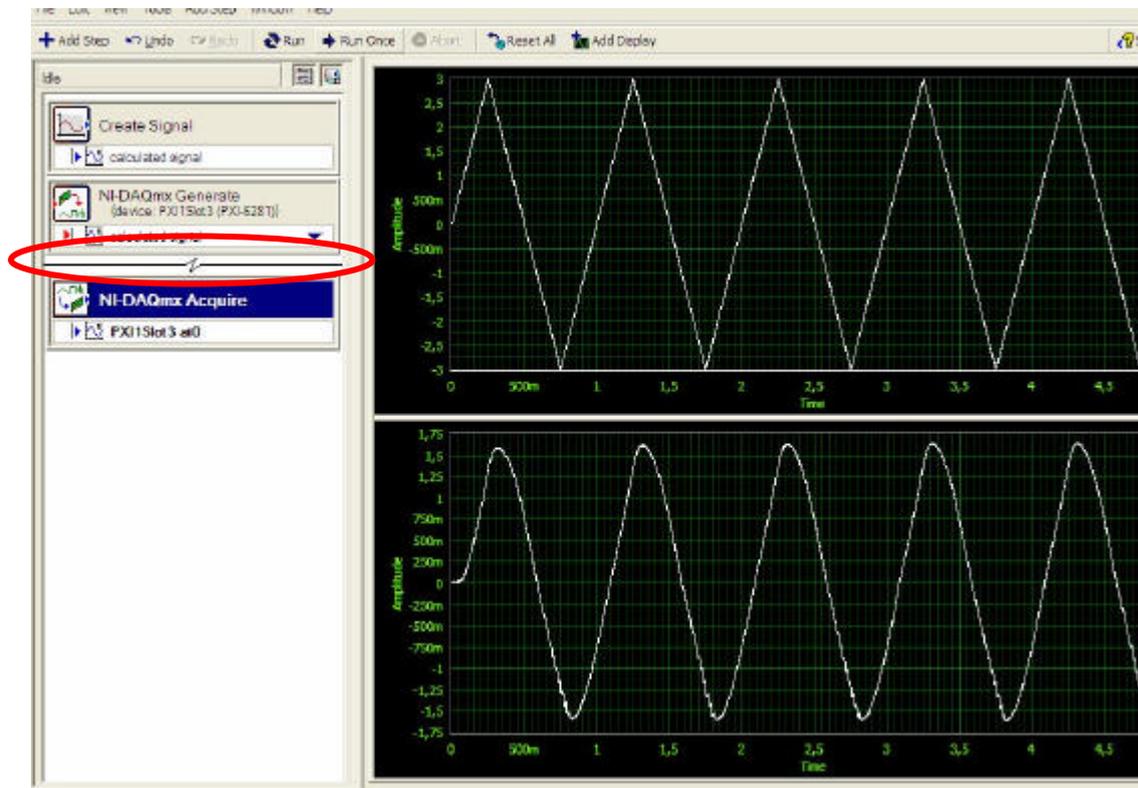


Figure n°8 : stimulus et réponse

3.4 Détermination de la fonction de transfert

Les données du stimulus et de la réponse vont maintenant être utilisées pour définir une fonction de transfert du système du moteur en boucle ouverte.

Pour cela, nous allons utiliser une estimation paramétrique. Cliquez successivement sur "+ Add Step", "System Identification", "Model Estimation" puis "Parametric Estimation". La fenêtre suivante devrait apparaître.

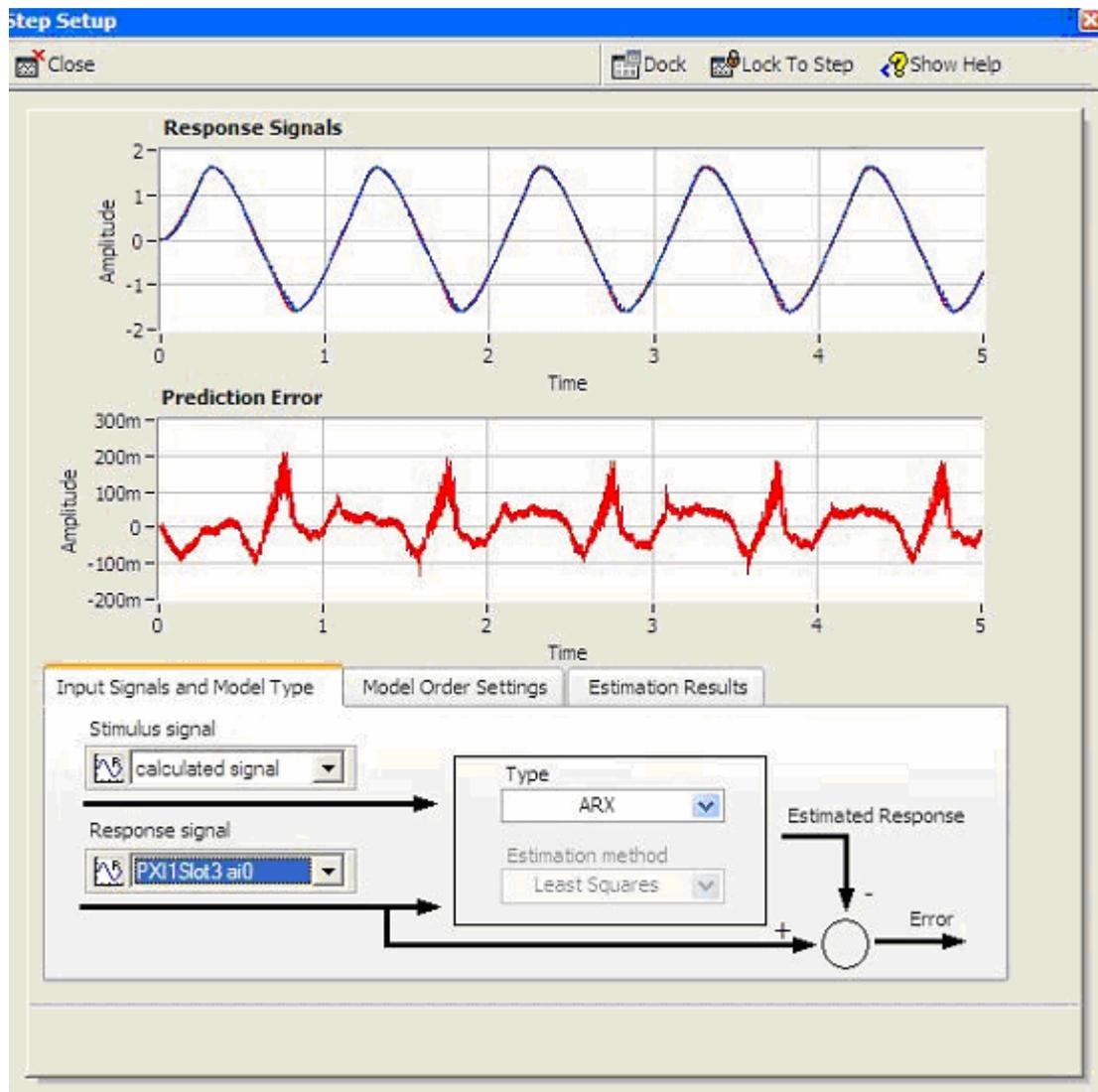


Figure n°9 : identification de l'estimation paramétrique du moteur

Cliquez sur les différents onglets afin de vous familiariser avec les différentes options. Notez que de nombreux types de modèles sont disponibles, que l'ordre et la méthode d'estimation peuvent être spécifiés, etc.

Ici, nous allons travailler avec une fonction de transfert du premier ordre ce qui vous permet de laisser les valeurs par défaut.

Voici dans le tableau ci-dessous où trouver ce pas (Step) et les valeurs à utiliser :

Step	Settings / Actions
System Identification -> Model Estimation -> Parametric Estimation	Input Signals and Model Tab: Stimulus Signal: Calculated signal Response Signal: PXI1slot3ai0
	Add Display under "DAQmx Acq" Output Display
	Drag "Estimated Response" to new Display

Remarquez que le symbole de déconnexion décrit plus haut a disparu aussitôt que les pas de création et d'acquisition ont été utilisés.

Remarquez également que l'erreur de prédiction la plus importante a lieu au début du signal. Ceci est dû au fait que le moteur peut démarrer plus ou moins rapidement. Autrement dit, les coefficients de la fonction de transfert résultante changeront donc légèrement à chaque fois que le script sera lancé.

Une fois le système identifié, la fonction de transfert doit être sauvegardée pour une analyse ultérieure. Ceci se fait en cliquant successivement sur "+ Add Step", "System Identification", "Import Export Model" puis sur "Save System Identification Model".

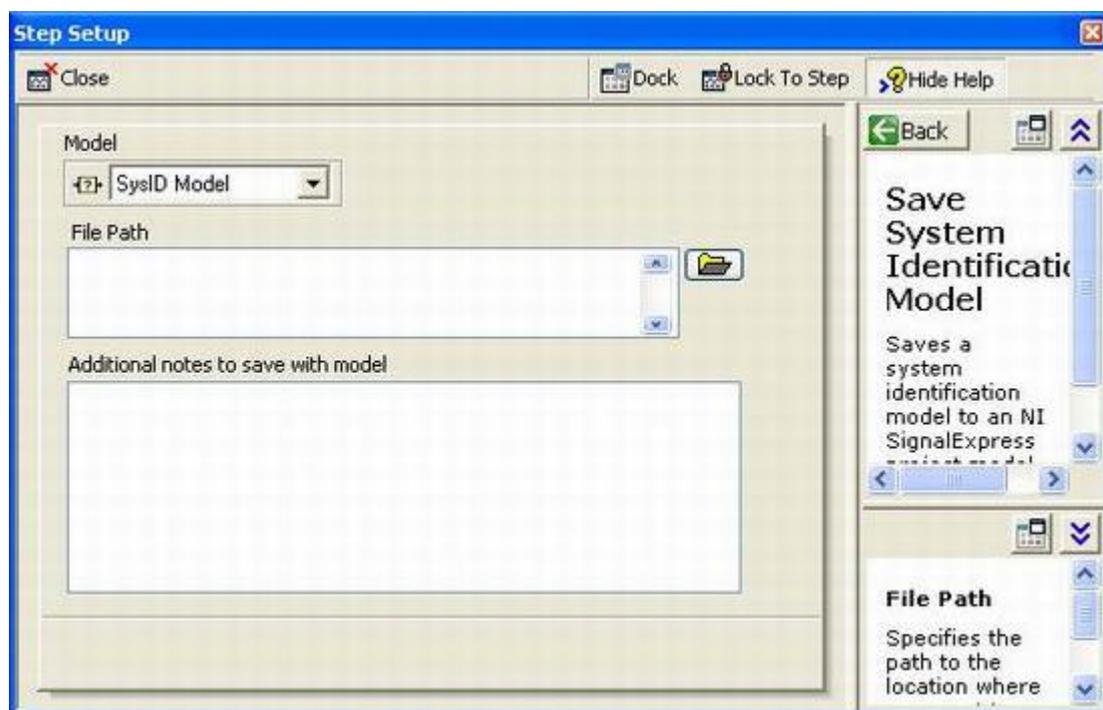


Figure n°10 : sauvegarde du modèle de l'identification du système

3.5 Transformation du modèle System ID en modèle Control Design

À ce niveau, la fonction de transfert est discrète. Bien qu'il soit possible de réaliser une conception de contrôle/commande discrète avec SignalExpress, le moteur est un système continu, nous allons donc convertir la fonction de transfert et concevoir le contrôleur en nous basant sur cette fonction de transfert continue.

Pour cela, nous allons procéder en deux étapes :

1. Conversion du modèle System ID en modèle Control Design
2. Avec les options du Control Design Toolkit, conversion du modèle discret de la fonction de transfert en un modèle continu.

L'étape qui permet de transformer du modèle System ID en un modèle Control Design est représentée sur la figure suivante. Elle se trouve dans "+ Add Step", "System Identification",

"Import-Export Model", "Convert to Control Design Model". Aucun paramétrage n'a besoin d'être fait dans cette boîte de dialogue.

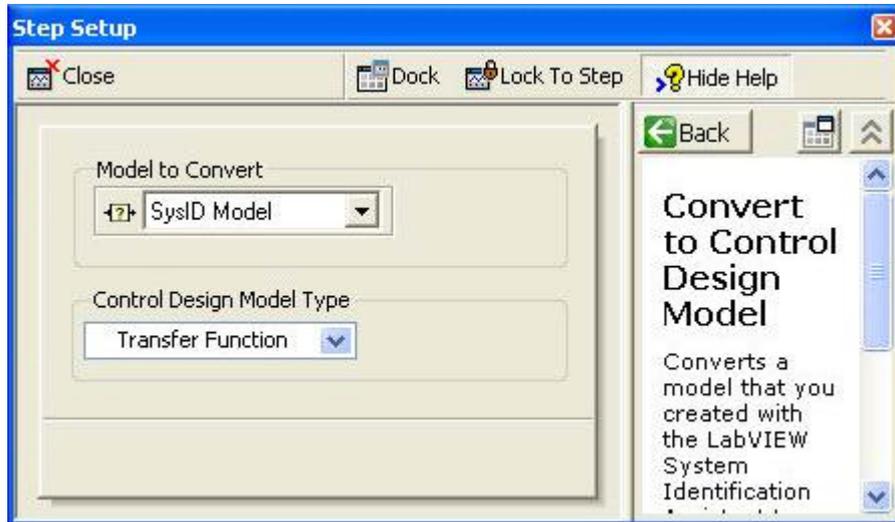


Figure n°11 : conversion du modèle System ID en un modèle Control

3.6 Transformation de la fonction de transfert discrète en une fonction de transfert continue

L'étape à utiliser se trouve dans "+ Add Step", "Control Design", "Model Transformation", "Discretize Model".

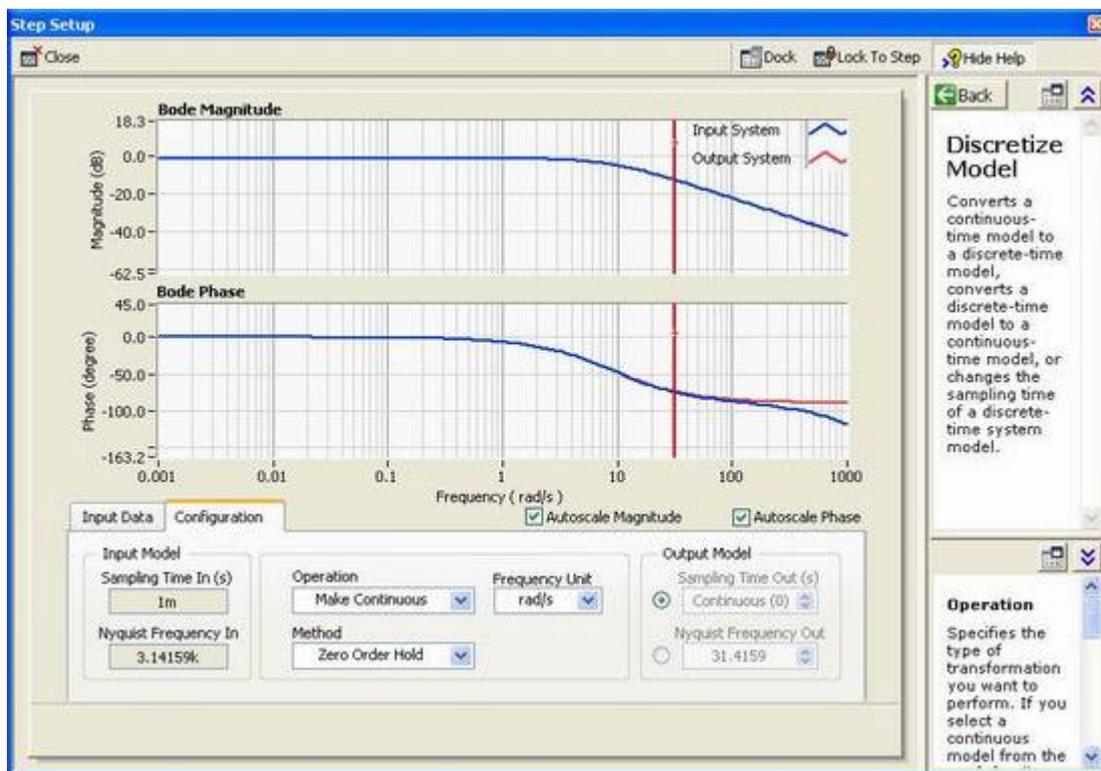


Figure n°12 : transformer le modèle discret en modèle continu

Le seul changement à faire ici consiste à sélectionner dans l'onglet ?Configuration?, l'option ?Make Continuous? dans le champ ?Operation?.

Pour afficher la fonction de transfert dans la zone d'affichage, commencez par créer une fenêtre supplémentaire. Ceci se fait en cliquant droit dans une des fenêtres et en sélectionnant les options ?Add Display" et "Below?.

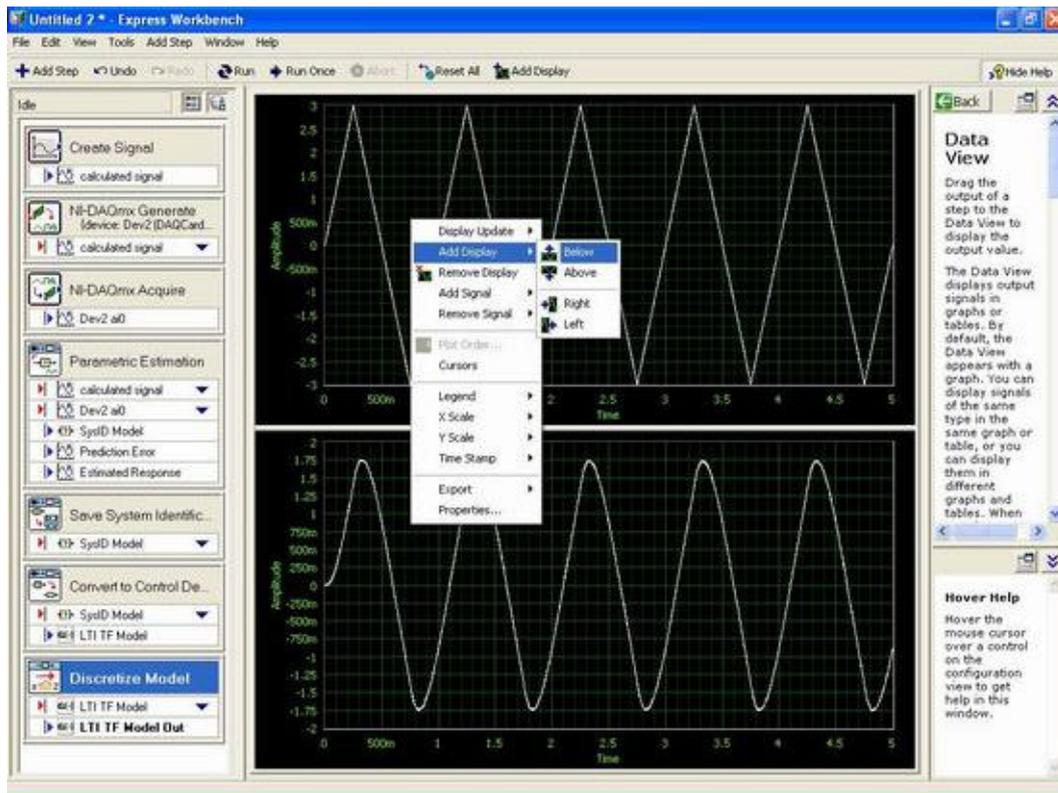


Figure n°13 : ajouter une fenêtre d'affichage pour la fonction de transfert du système en boucle ouverte.

Une fois cette nouvelle fenêtre créée, faites glisser alors le résultat de l'étape ?Discretize Model?.

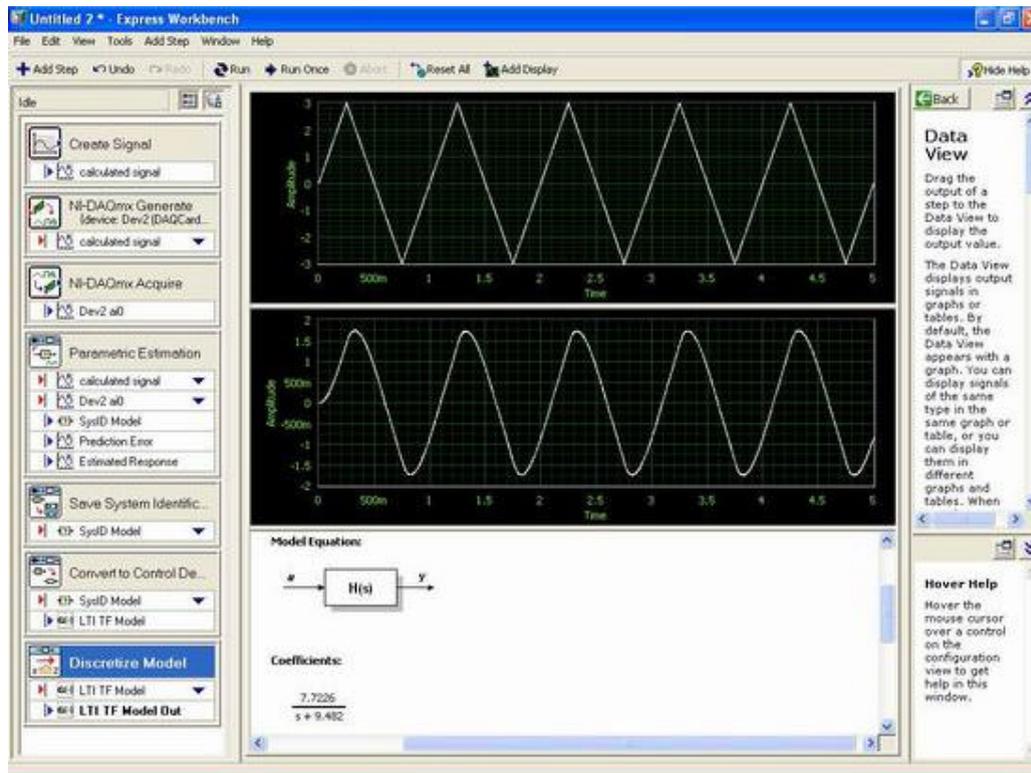


Figure n°14 : afficher la fonction de transfert continue du système.

À ce niveau, le système a été identifié comme une fonction de transfert du premier ordre et nous disposons de sa fonction de transfert en mode continu.

4 CONCEPTION DU RÉGULATEUR

Étant donné que la fonction de transfert du moteur est disponible, un contrôleur peut être conçu afin de répondre aux exigences de temps de montée, temps d'établissement et de dépassement (overshoot), etc.

Pour simplifier, nous allons concevoir un contrôleur de type PI en utilisant le Step "PID synthesis". Notez qu'il est également possible d'utiliser la méthode du "lieu des pôles" ou des diagrammes de Bode pour la conception du contrôleur.

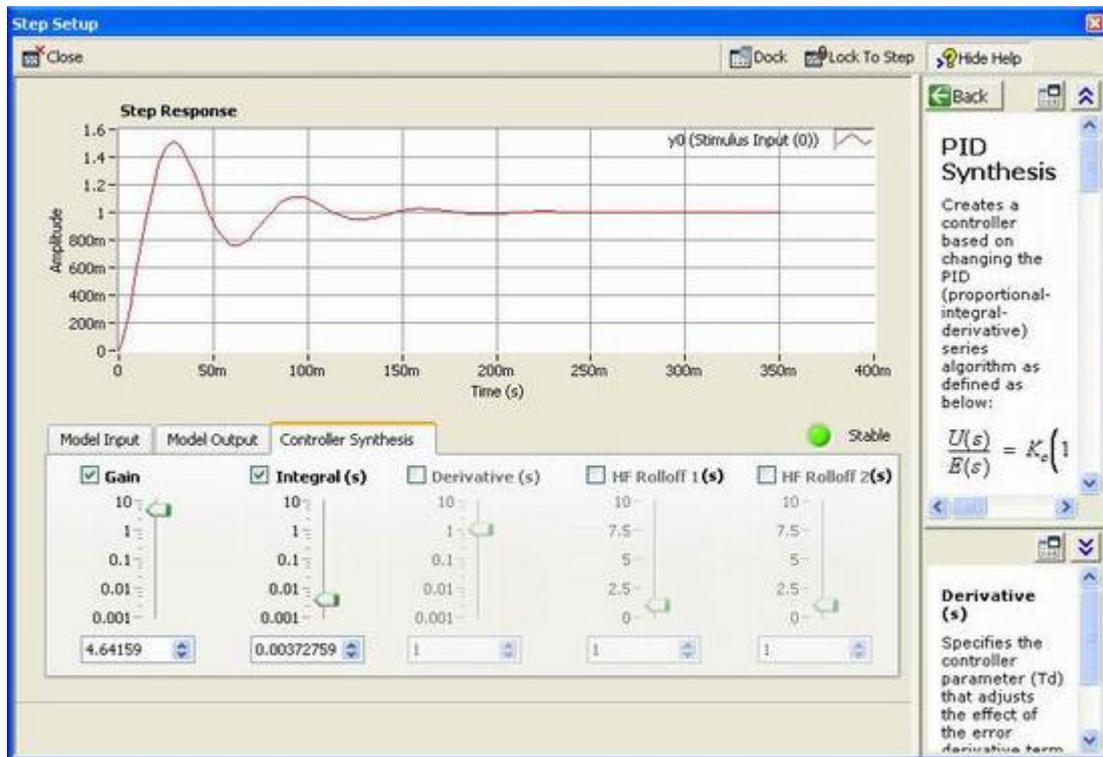


Figure n°15 : conception d'un contrôleur PI

Voici dans le tableau ci-dessous où trouver le Step et les valeurs à utiliser :

Step	Settings / Actions
Control Design -> Controller Design -> PID Synthesis	Controller Synthesis Tab: Côtcher les cases "Gain" et "Integral (s)"
	Ajuster les gains P et I pour obtenir la réponse à un échelon voulue

Lorsque le P et le I sont modifiés, la réponse indicielle varie en termes de temps de montée, de temps d'établissement ainsi que de dépassement (overshoot). Il suffit juste alors d'ajuster les gains jusqu'à ce que la réponse indicielle vous paraisse correcte. Une réponse indicielle semblable à celle représentée sur la figure précédente pourrait être acceptable avec un temps de montée d'approximativement 25 ms et un overshoot de moins de 50%. Un overshoot trop important risque de causer en sortie du contrôleur une commande de tension plus élevée que ce que peut supporter la carte DAQ, mais aussi le moteur.

Le modèle du régulateur doit être sauvegardé pour pouvoir être réutilisé dans l'étape de simulation du système en boucle fermée. Pour se faire, utilisez le Step "Save Control Design Model" ("Control Design", "Import/Export Model"). Il suffit d'indiquer un chemin et de nommer le modèle du contrôleur.

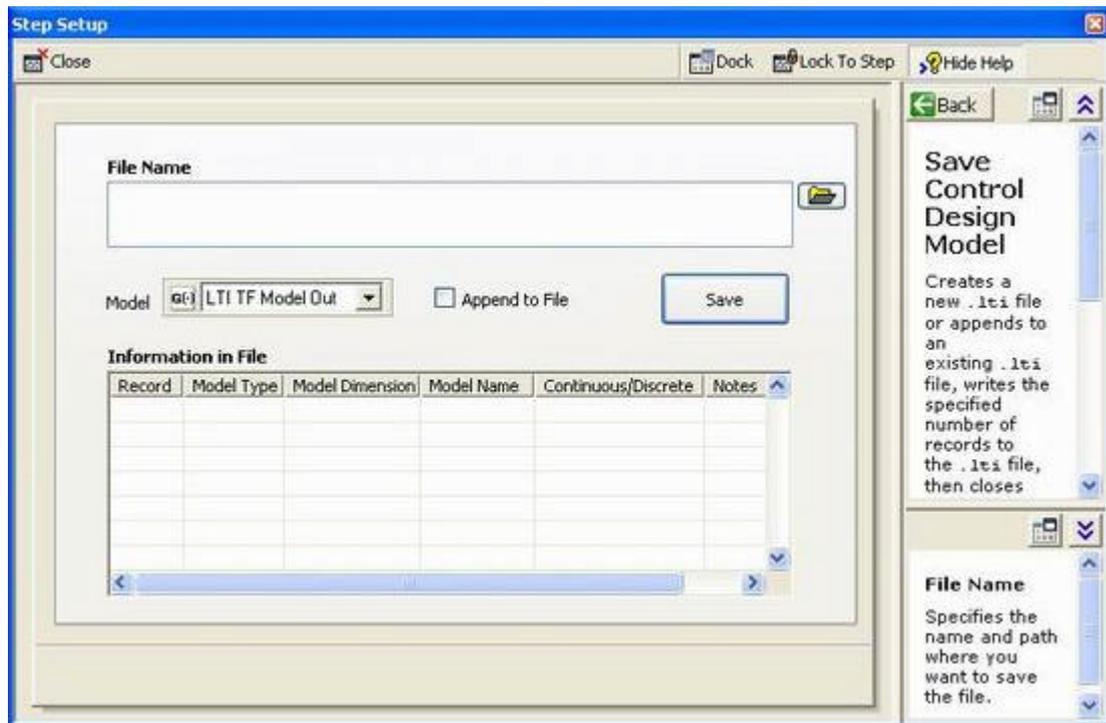


Figure n°16 : sauvegarde du contrôleur PI

5 SIMULATION DU SYSTÈME EN BOUCLE FERMÉE

Maintenant que le système est défini et que le contrôleur est conçu, l'étape suivante consiste à simuler le système en boucle fermée. Cette étape permet de tester le régulateur dans des conditions limites sans risque puisque tout est simulé sur le PC.

Pour ce faire, nous allons utiliser LabVIEW et le Simulation Module. Même si cette partie sort du cadre de l'article qui se consacre essentiellement à SignalExpress, il est intéressant de connaître la méthode de travail. Cela doit, entre autres, permettre de réaliser que les outils d'Instrumentation Virtuelle forme une chaîne et que s'il est très facile (sans programmation) de rapidement commencer sous SignalExpress, LabVIEW permet de récupérer tout ce qui a été fait préalablement et d'aller plus loin.

Dans la figure ci-dessous, on retrouve une interface graphique LabVIEW. Notez que cette dernière a été réalisée par l'ingénieur. En effet, l'Instrumentation Virtuelle parle de ' *systèmes définis par l'utilisateur* '. On y retrouve deux graphes sur lesquels nous avons choisi d'afficher la vitesse effective du moteur (graphe du haut) ainsi que la vitesse du moteur, comparée à la consigne (graphe du bas). Pour le reste, on retrouve un bouton Stop pour arrêter l'application ainsi que trois autres contrôles qui permettent de paramétrer le signal de consigne.

Il faut donc réaliser toute la flexibilité qu'offre l'environnement LabVIEW. Par exemple, l'utilisateur aurait pu choisir de n'avoir qu'un graphe, de n'avoir aucun paramètre pour le signal de consigne, etc.

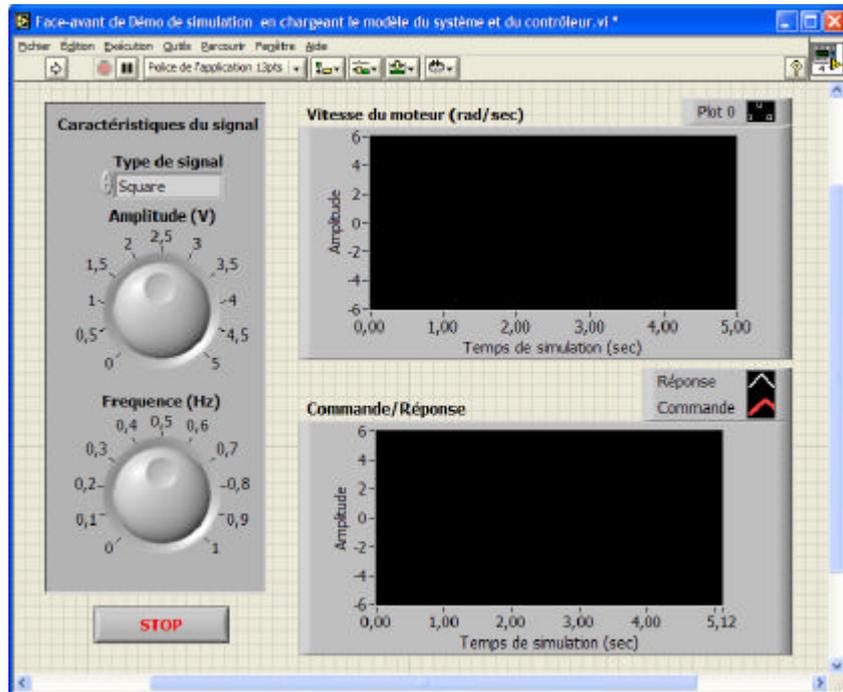


Figure n°17: la face-avant de l'application de simulation en boucle fermée

Passons maintenant à l'étude du code. Ce dernier se trouve dans la figure suivante.

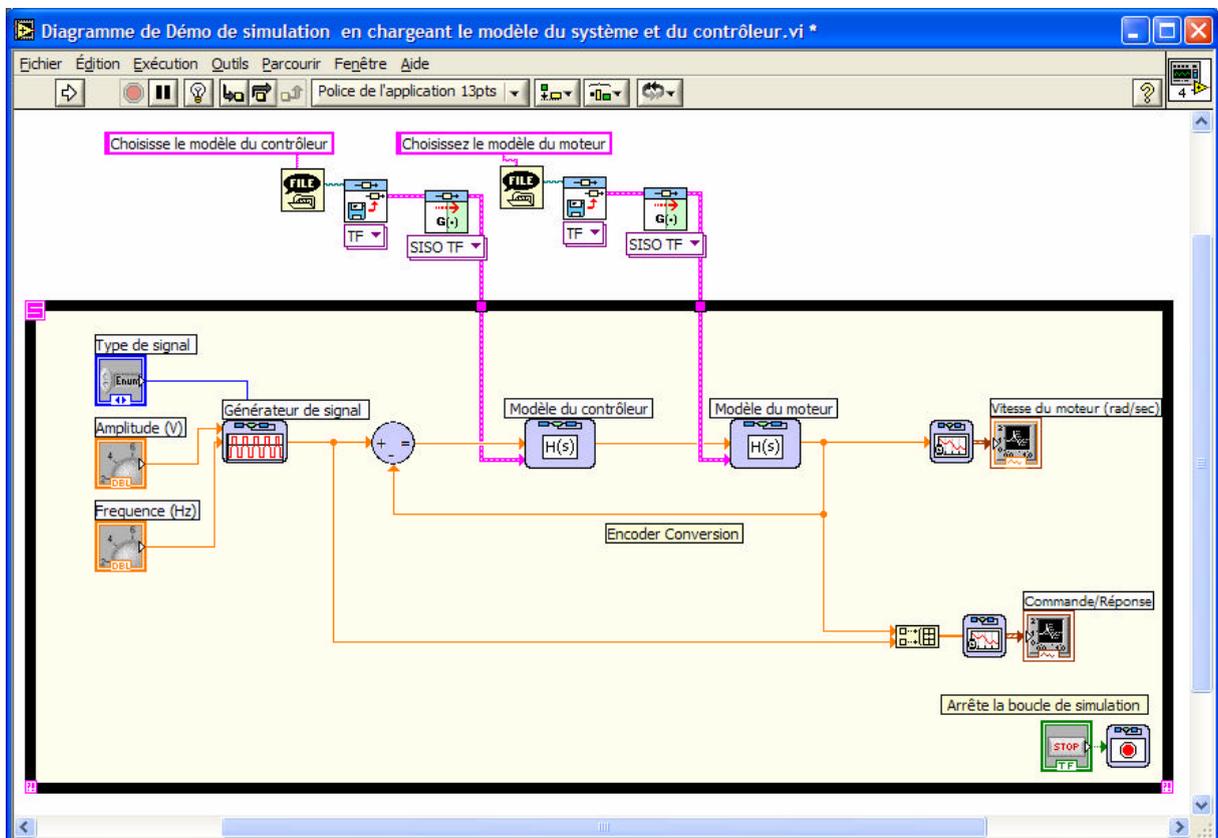


Figure n°18 : le code de l'application de simulation en boucle fermée

Au cœur de l'application se trouve une boucle de simulation (le grand rectangle noir) dans laquelle le programmeur peut coder des boucles contre réaction, insérer des sommateurs, des fonctions de transfert, etc.

En bas à droite de la boucle de simulation, on observe que la condition d'arrêt est connectée au bouton Stop ce qui signifie que le programme s'arrête simplement dès que l'on appuie sur le bouton.

À gauche du diagramme, on retrouve, connectés à un bloc fonctionnel de haut niveau (un générateur de signaux) les trois contrôles qui permettent de paramétrer le signal de consigne (fréquence, amplitude, type de signaux).

En continuant la lecture du diagramme de gauche à droite, on trouve successivement un sommateur, la fonction de transfert du régulateur et la fonction de transfert du moteur à courant continu. À propos des fonctions de transfert, il est intéressant de noter qu'au lieu de taper la formule en "s", ici, l'ingénieur a décidé d'aller rechercher sur le disque dur les fonctions de transfert qu'il avait déterminées à l'aide de SignalExpress. La lecture des fonctions de transfert se fait via les deux groupes d'icônes (tout en haut du diagramme). Dans chaque groupe, on affiche une boîte de dialogue Windows classique pour permettre à l'utilisateur de choisir tel ou tel modèle, puis on effectue une lecture du fichier sélectionné et enfin on convertit le modèle Control Design en un modèle utilisable par le module de simulation.

On notera pour finir la boucle de contre-réaction et la façon dont les graphes sont alimentés par des fils. Le graphe du haut n'affichera que la vitesse effective du moteur alors que le graphe du bas permettra de comparer la vitesse effective du moteur à la valeur de la consigne.

6 CONCLUSION

Nous avons donc rapidement démontré une partie de ce qu'il était possible de faire dans le monde de la conception à l'aide de SignalExpress et de LabVIEW. Ceci dit, l'affaire ne s'arrête pas là...

En effet, SignalExpress est une application qui permet tout aussi simplement de faire de l'automatisation de prise de mesures et de l'analyse. Par exemple, avec SignalExpress, il est possible, sans programmer, de comparer des mesures réelles avec celles issues d'un simulateur SPICE, de même, il est tout aussi facile de "glisser-déposer" des données dans Excel ou d'exporter ses scripts dans LabVIEW, etc.

Pour beaucoup, SignalExpress est LE produit qui permet de faire bénéficier les non-programmeurs de tous les avantages qu'offre l'Instrumentation Virtuelle.

D'un autre côté, une fois la phase de simulation validée dans LabVIEW, il est tout à fait envisageable d'aller plus loin. Beaucoup plus loin même... En effet, LabVIEW permet de déployer du code sur des cibles temps réel ou FPGA. Autrement dit, parce qu'il a fait le choix de LabVIEW, l'ingénieur peut facilement prototyper son régulateur en déployant ce dernier sur une cible RT connectée au système à asservir.

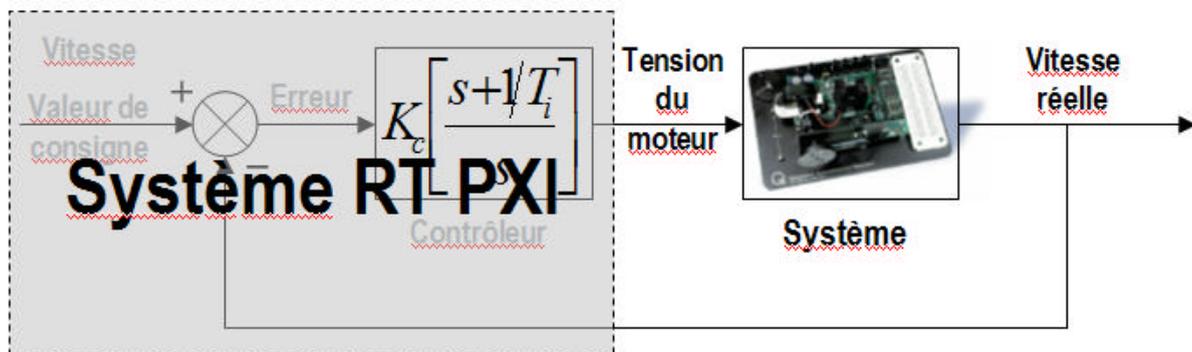


Figure n°19 : prototypage rapide avec LabVIEW Real-Time

Ceci fait, le code du contrôleur sera sans doute embarqué dans un boîtier final. Or, comme l'ingénieur possède déjà le modèle du système à contrôler, il pourra en déployant ce dernier sur une cible RT faire ce que l'on appelle du HIL (Hardware-In-the-Loop).

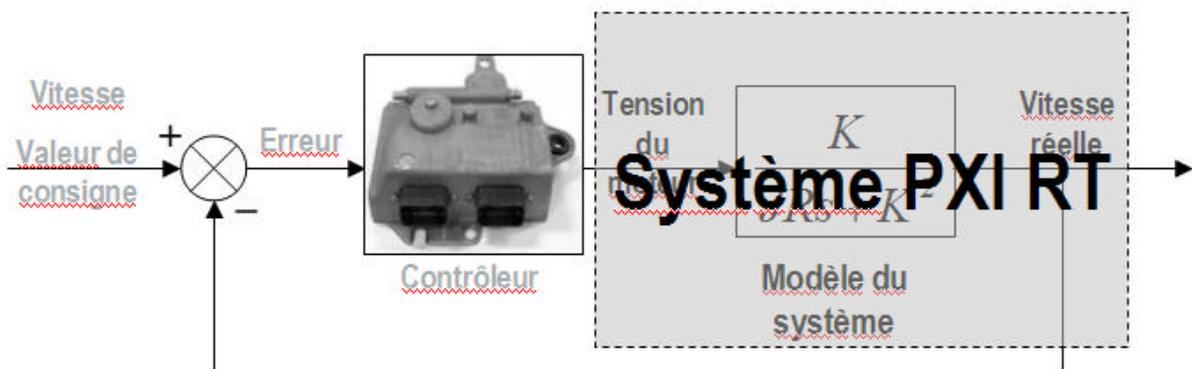


Figure n°20 : application HIL avec LabVIEW Real-Time

Pour finir, quand le régulateur aura été validé, il "suffira" d'utiliser LabVIEW pour faire des tests de production ou embarqués classiques.

Le plus important à noter, c'est sans doute le fait qu'aujourd'hui, l'Instrumentation Virtuelle et les outils qui en découlent, offre une véritable plate-forme sur laquelle il est possible de s'appuyer pour concevoir mieux, plus facilement et plus efficacement.