



LYCÉE LA MARTINIÈRE MONPLAISIR LYON
SCIENCES INDUSTRIELLES POUR L'INGÉNIEUR
CLASSE PRÉPARATOIRE P.S.I.
ANNÉE 2016 - 2017

TIPE

Fiche méthode 1 - Réaliser d'une chaîne structurelle avec Scilab et Arduino(TIPE)

7 décembre 2016

1 Introduction

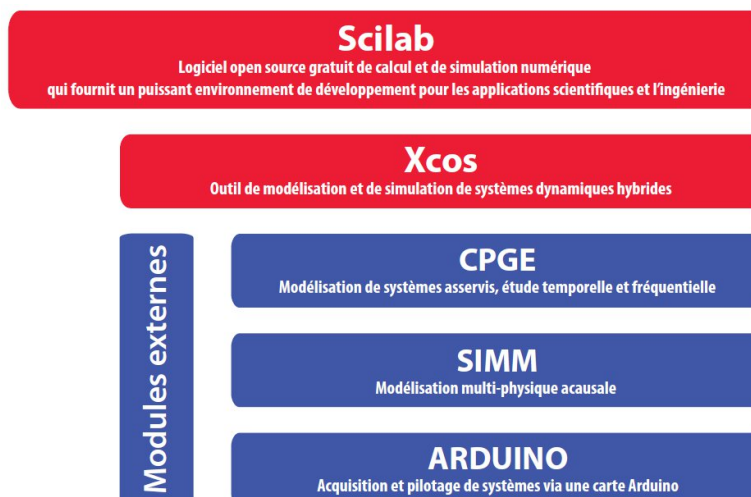
La simulation numérique est aujourd'hui incontournable dans le processus de conception de systèmes. Simuler des phénomènes complexes (physiques, mécaniques, électroniques, etc.) permet d'en étudier les comportements et d'obtenir des résultats sans avoir besoin de recourir à l'expérience réelle. Largement utilisée dans le monde de l'industrie, les ingénieurs et les chercheurs de demain sont formés dès le secondaire aux concepts de modélisation et de simulation. Xcos est l'outil de Scilab dédié à la modélisation et à la simulation de systèmes dynamiques hybrides incluant à la fois des modèles continus et discrets. Il permet aussi de simuler des systèmes régis par des équations explicites (simulation causale) et implicites (simulation acausale). Xcos inclut un éditeur graphique permettant de représenter facilement des modèles sous forme de schémas fonctionnels (diagrammes) en connectant des blocs entre eux. Chaque bloc représente une fonction de base prédéfinie ou une fonction définie par l'utilisateur.

Distribué librement et gratuitement avec Scilab, Xcos est l'outil idéal pour l'enseignement et l'apprentissage des sciences de l'ingénieur en classe comme à la maison.

2 Installation de Scilab

Ce document fait référence à la version 5.4.1

Scilab est un logiciel open source de calcul numérique que chacun peut télécharger gratuitement. Disponible sous Windows, Linux et Mac OS X, Scilab est téléchargeable à l'adresse suivante : <http://www.scilab.org> Une fois Scilab téléchargé (**Bien choisir la version 5.4.1**) et installé, il faut ajouter les modules externes complémentaires, listés ci-dessous :



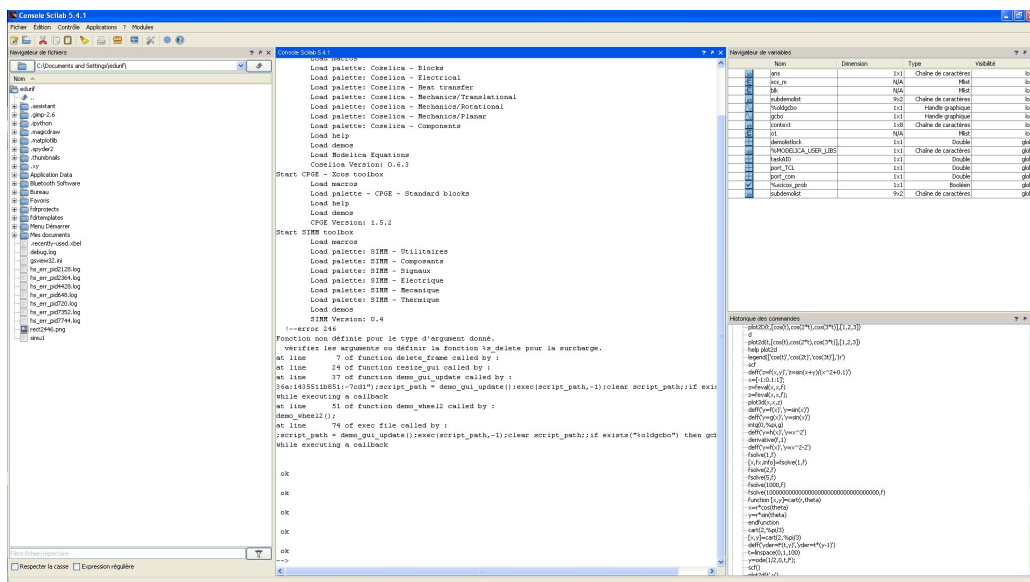
Pour cela, il suffit :

- D'être connecté à Internet
- De lancer Scilab
- De cliquer dans la barre de menus sur **Applications/Gestionnaire de modules - ATOMS** puis d'aller chercher les différents modules dans les catégories indiquées ci-dessous :
 - CPGE (catégorie Éducation),
 - SIMM (catégorie Éducation),
 - Arduino (catégorie Instruments Control ou contrôle d'instrument).

Pour chaque module, cliquez sur le bouton Installer, quittez Scilab et relancez-le. Les modules sont alors installés et se chargeront automatiquement à chaque démarrage de Scilab. Pour supprimer un module ou le mettre à jour, il suffit de retourner dans le **Gestionnaire de modules - ATOMS** et de cliquer cette fois sur le bouton correspondant Supprimer ou mettre à jour.

3 Description de Scilab : environnement général

Après avoir lancé Scilab, l'environnement par défaut est constitué d'une console, d'un navigateur de fichiers, d'un navigateur de variables et d'un historique des commandes.



Dans la console, après “-->”, il suffit de saisir une commande et d'appuyer sur la touche entrée du clavier pour obtenir le résultat correspondant. Xcos peut être ouvert :

- Depuis la barre d'outils, via l'icône

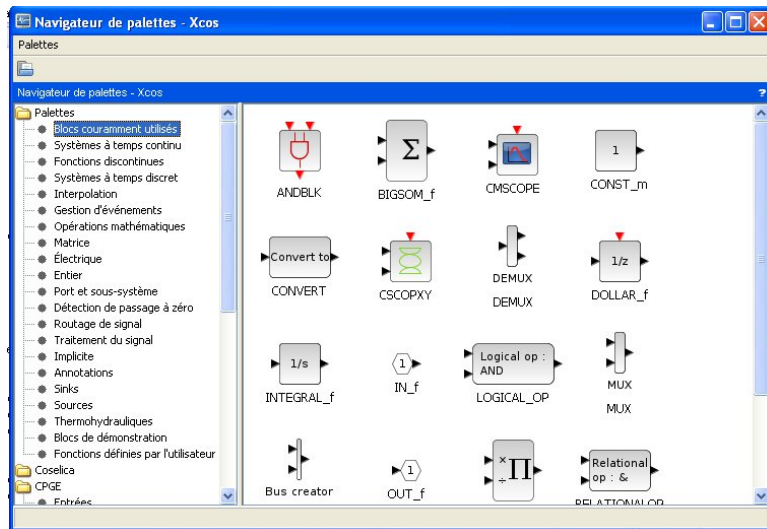


- Depuis la barre de menus, dans applications / Xcos;
- Depuis la console, en tapant :

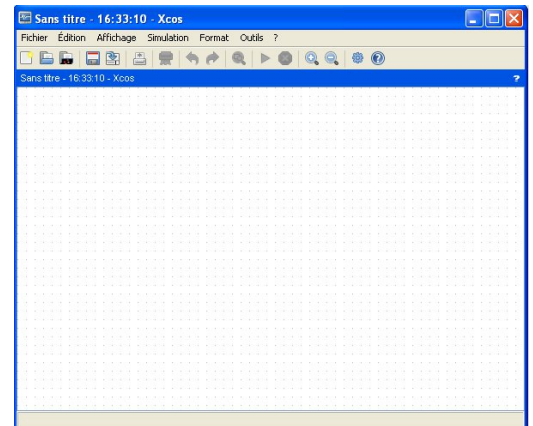
--> xcos

Xcos s'ouvre, par défaut, avec deux fenêtres :

- Le navigateur de palettes qui met à disposition un ensemble de blocs prédéfinis,
- Une fenêtre d'édition qui est la zone de construction d'un diagramme.



Navigateur de palette - Xcos



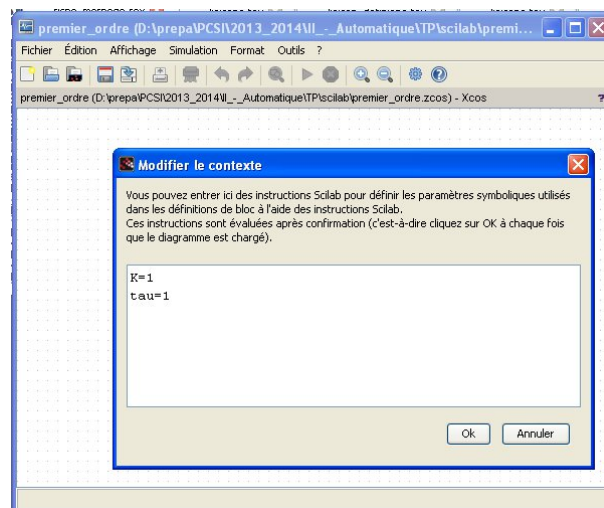
Fenêtre d'édition

Pour construire un diagramme, l'utilisateur sélectionne les blocs dans le navigateur de palettes et les positionne dans la fenêtre d'édition (cliquer / glisser / déposer). Il peut ensuite connecter les blocs entre eux en utilisant leurs différents ports (entrée / sortie / événement) pour pouvoir simuler le modèle créé.

4 Déclaration des variables

Dans Xcos, il est possible de créer des variables qui seront utiles pour faciliter la construction d'un diagramme. Nous utilisons ici un "contexte" pour fixer des valeurs de référence lors de la simulation du diagramme. Pour cela, il suffit de cliquer sur **Simulation / modifier le contexte** dans la barre de menus et de déclarer les variables.

Par exemple si on souhaite simuler le comportement d'un premier ordre, nous serons amenés à définir le gain statique K ainsi que la constante de temps τ :





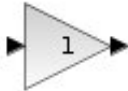
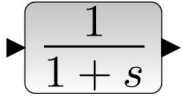

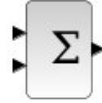
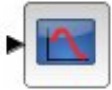


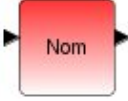

- $K = 1$
- $\tau = 1$

Vous pouvez maintenant utiliser ces variables pour le paramétrage des blocs du diagramme.

5 Exemple de fonctions élémentaires

Pour construire un diagramme de simulation on utilise des blocs qui permettent de créer des entrées (échelon, Dirac, sinus, etc.), des opérateurs linéaires (fonction de transfert) mais aussi des outils d'analyse (générateur de variables, oscilloscope, base de temps, etc...). Le tableau suivant dresse une liste non exhaustive des fonctions que l'on rencontre habituellement.

Une fois que l'on met en place un bloc il ne faut pas hésiter à double-cliquer dessus pour éditer les réglages associés.

Désignation	Représentation	Sous-palette standard
Echelon		CPGE/entrées/STEP_FUNCTION
Sinus		CPGE/entrées/GENSIN_f
Gain proportionnel		CPGE/Opérateurs linéaires/GAINBLK_f
Fonction de transfert continue		CPGE/Opérateurs linéaires/ CLR
Intégrateur $H(p) = \frac{1}{p}$		CPGE/Opérateurs linéaires/INTEGRAL_f
Sommateur/comparateur		CPGE/Opérateurs linéaires/BIGSOM_f
Visualisation		CPGE/Sorties/SCOPE
Définition de la base de temps		CPGE/Analyses/REP_TEMP
Tracer du diagramme de Bode		CPGE/Analyses/REP_FREQ
Affectation d'une variable		CPGE/Analyses/GRANDEUR_PHYSIQUE
Étude paramétrique		CPGE/Analyses/PARAM_VAR



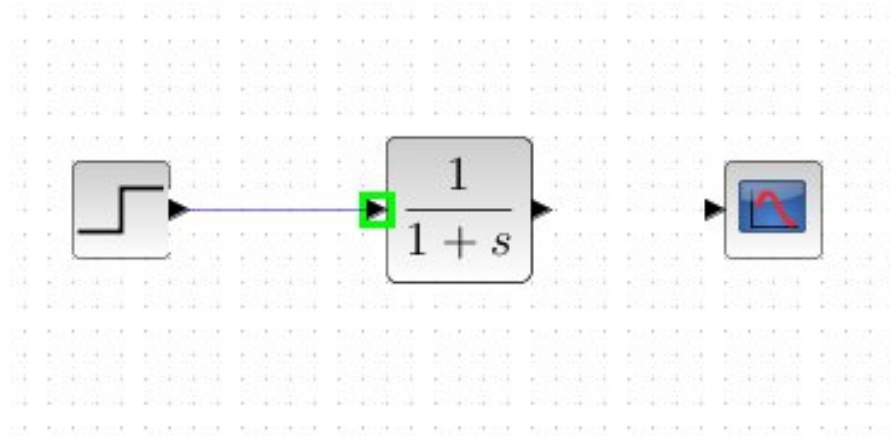
Remarque 1 : Notation de la variable de Laplace dans Xcos

la notation “s” (variable de Laplace) est souvent utilisée dans les pays anglo-saxons alors que la notation “p” est utilisée notamment en France et en Allemagne. Dans les logiciels de simulation, on utilise donc à la place de p , la notation s .

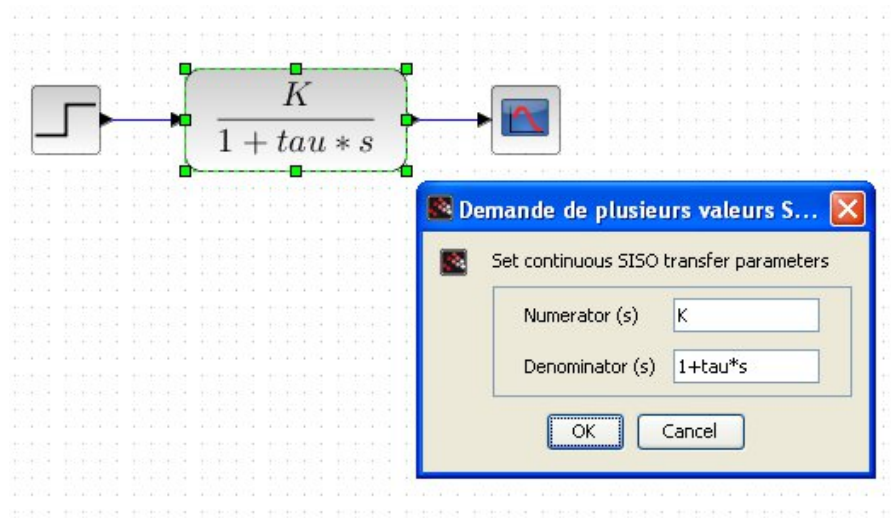
6 Exemple de construction d'une simulation

On se propose ici de créer une simulation d'un système du premier ordre sollicité par un échelon.

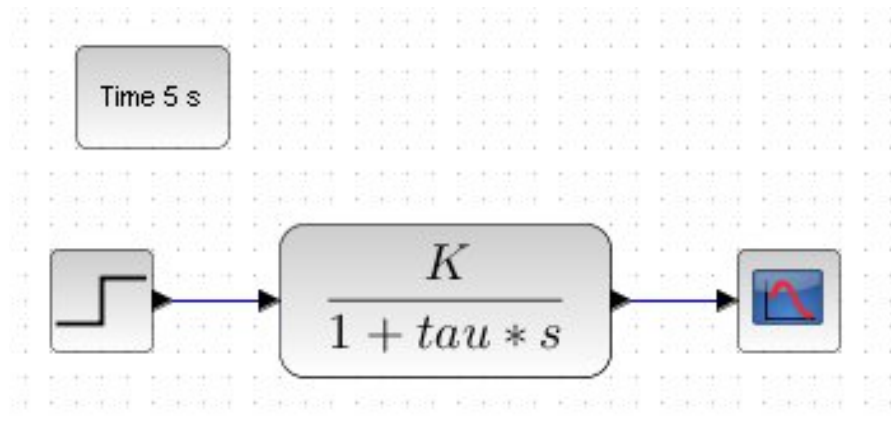
1. Disposez les blocs dans la fenêtre d'édition : échelon, fonction continu et oscilloscope (les faire glisser depuis le navigateur de palette)
2. Relier les ports d'entrée et de sortie entre eux : cliquez sur la sortie (flèche noire) d'un bloc et en maintenant le bouton de la souris appuyé, reliez au port d'entrée du bloc correspondant, un carré vert apparaît en surbrillance pour indiquer que le lien est correct. Relâchez pour finaliser le lien. Complétez ensuite les connexions des blocs entre eux.



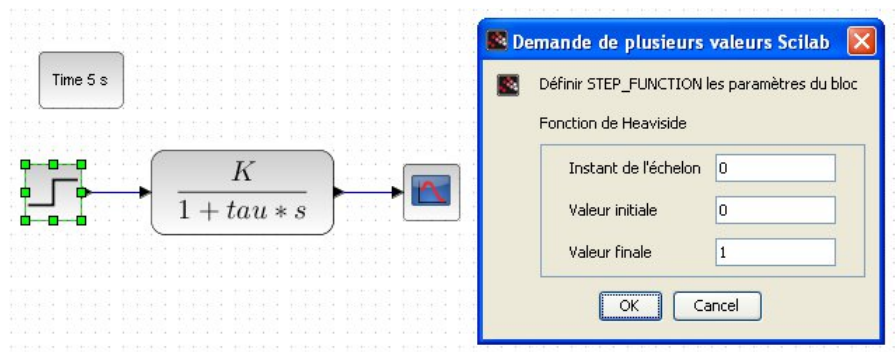
3. Modifier le bloc “Fonction de transfert continue” pour faire apparaître $\frac{K}{1+\tau s}$ en double cliquant dessus.



4. Configurer la durée de simulation en mettant en place un bloc de “Définition de la base de temps”. Dans cet exemple on peut effectuer une simulation durant 5 s



5. Configurer l'échelon en double cliquant sur le bloc associé et en réglant l'instant initial à 0 pour que l'échelon ne soit pas retardé. L'amplitude de l'échelon peut être réglée dans la case valeur finale.



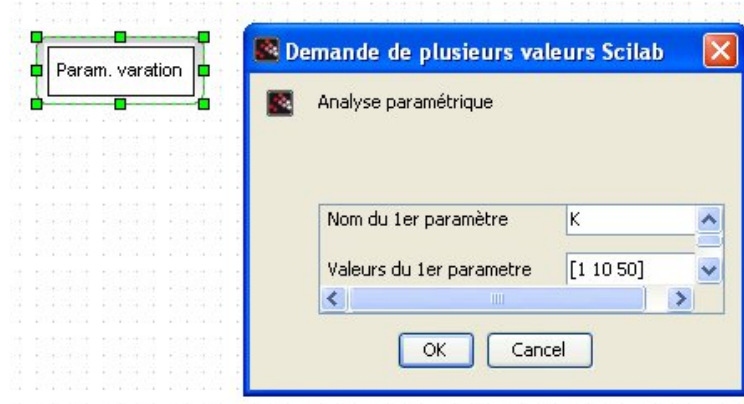
6. Lancer la simulation depuis le menu "Simulation / démarrer" ou en cliquant sur l'icône :



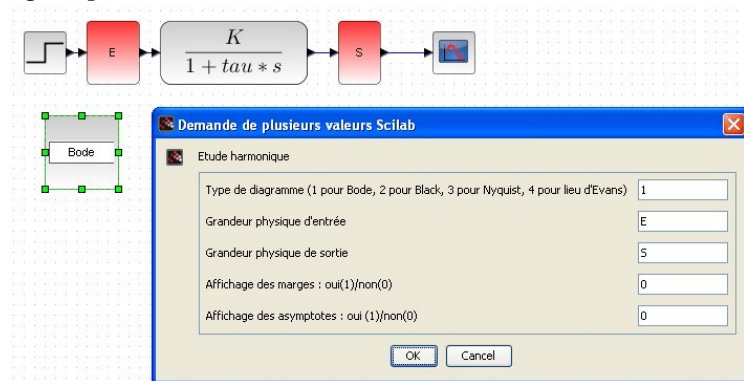
7. Une nouvelle fenêtre (scope) apparaît, montrant l'évolution de la simulation.

Remarque 2 : Compléments de syntaxe

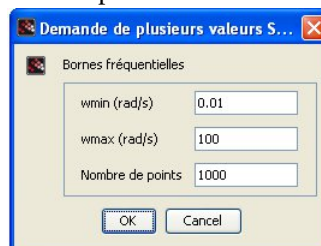
- Pour tracer plusieurs courbes en faisant varier un paramètre, il faut utiliser le bloc d'étude paramétrique ("PARAM_VAR"). L'exemple suivant montre la définition de trois valeurs paramétriques pour la variable K .



- Pour tracer un diagramme de Bode, il faut dans un premier temps associer respectivement à l'entrée et à la sortie les noms de variables E et S à l'aide du bloc d'Affectation d'une variable physique (GRANDEUR_PHYSIQUE).
- Il suffit ensuite d'utiliser le bloc "Bode" (REP_FREQ). En double cliquant dessus, on peut définir la plage de pulsation d'étude (en rad/s).



- On peut alors fixer la plage d'étude des pulsation.



- Enfin on peut lancer la simulation qui aboutira au tracé des diagrammes de Bode.

7 Scilab et arduino

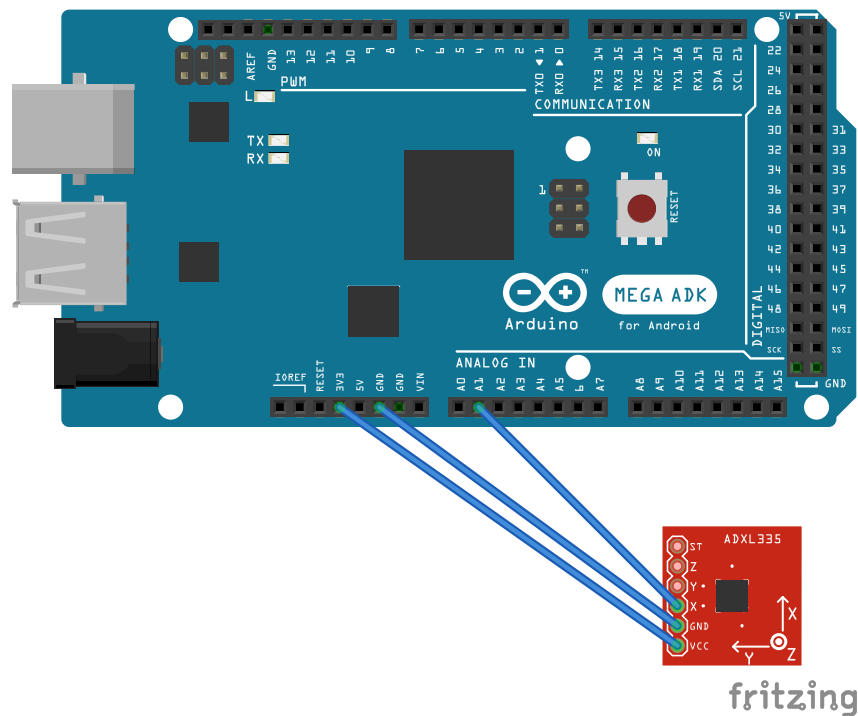
a) Préparation de la carte

- Brancher la carte arduino via un port usb
- Repérer le port de communication : dans le menu Démarrer , Périphérique et imprimante
- Ouvrir le logiciel arduino (icône)
 - Verifier dans outil le type de carte et le port

- ouvrir la fonction *toolbox_arduino_v3*
- téléverser le programme vers la carte.

b) Câblage de la carte

Ce schéma ci-dessous représente le câblage d'un capteur vers une carte Arduino Méga. Les fonctions données ci-dessous sont à aller chercher dans la navigateur de palette et dans la rubrique arduino.

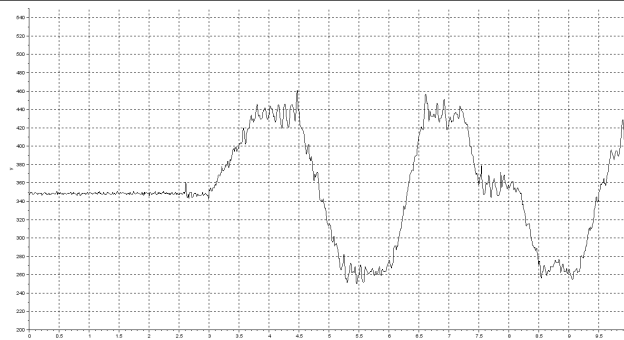
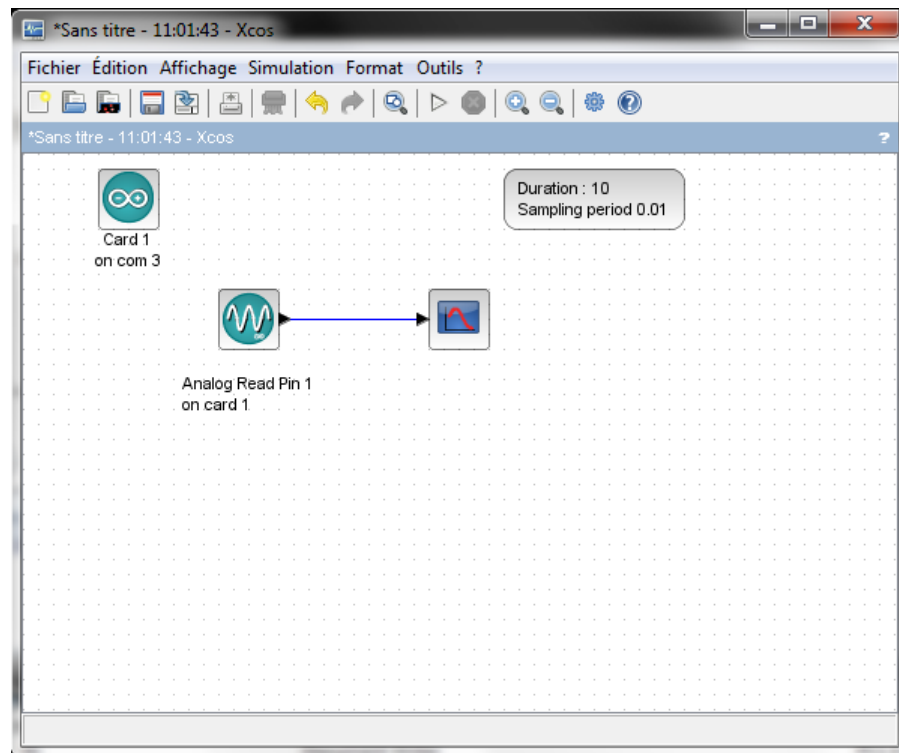


On dispose d'un accéléromètre analogique. Il faut connecter :

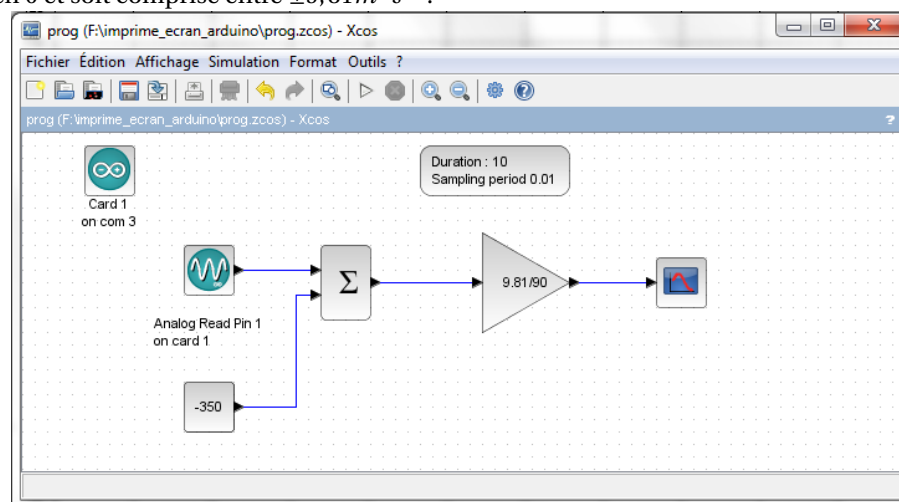
- la borne **VCC ou 3V3** du capteur vers **3V3** de la carte Arduino ;
- La borne **"x"** (par exemple) vers une entrée analogique (**A1** par exemple) ;
- La borne **GND** du capteur vers la **GND** de la carte Arduino ;

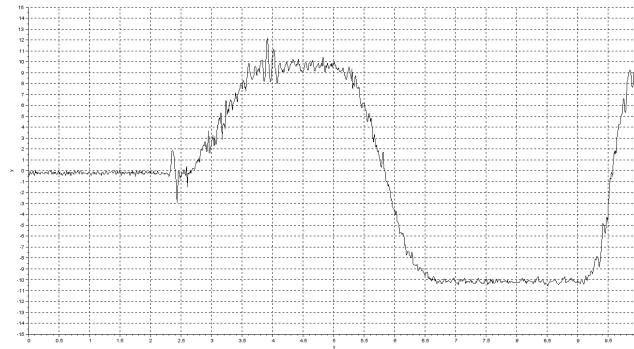
c) Câblage de la carte et Mise en place du programme

- Lecture d'une entrée analogique :



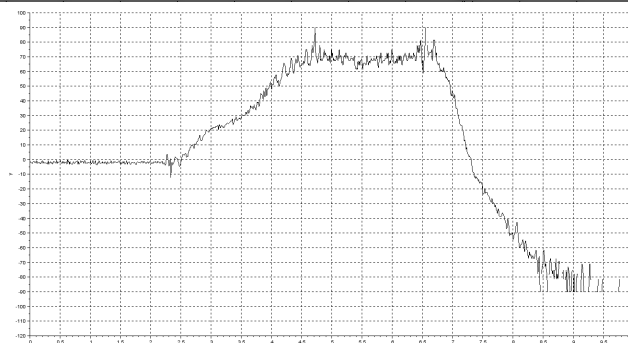
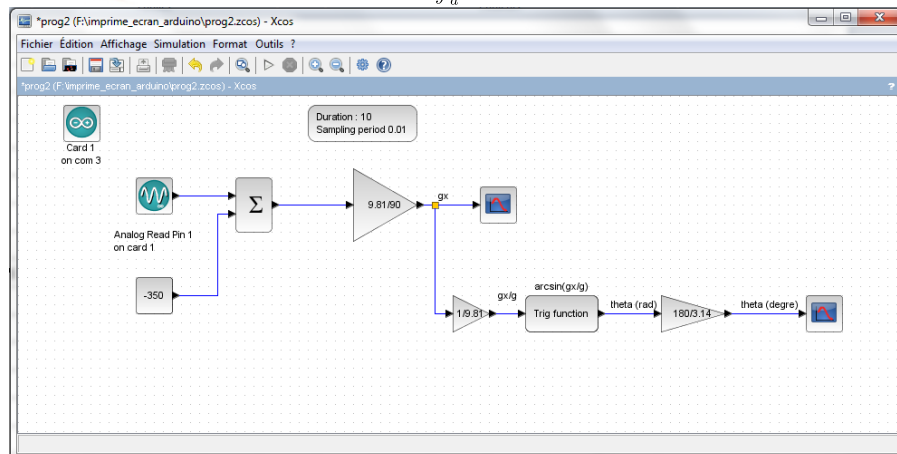
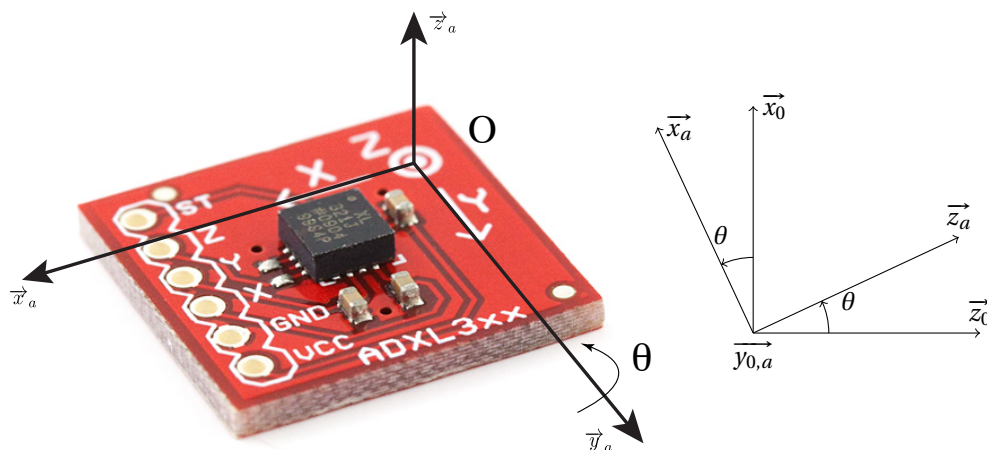
- Réglage de l'offset et d'un gain d'une entrée analogique : ici on souhaite que l'accélération mesurée selon \vec{x} soit centrée en 0 et soit comprise entre $\pm 9,81 m \cdot s^{-2}$.



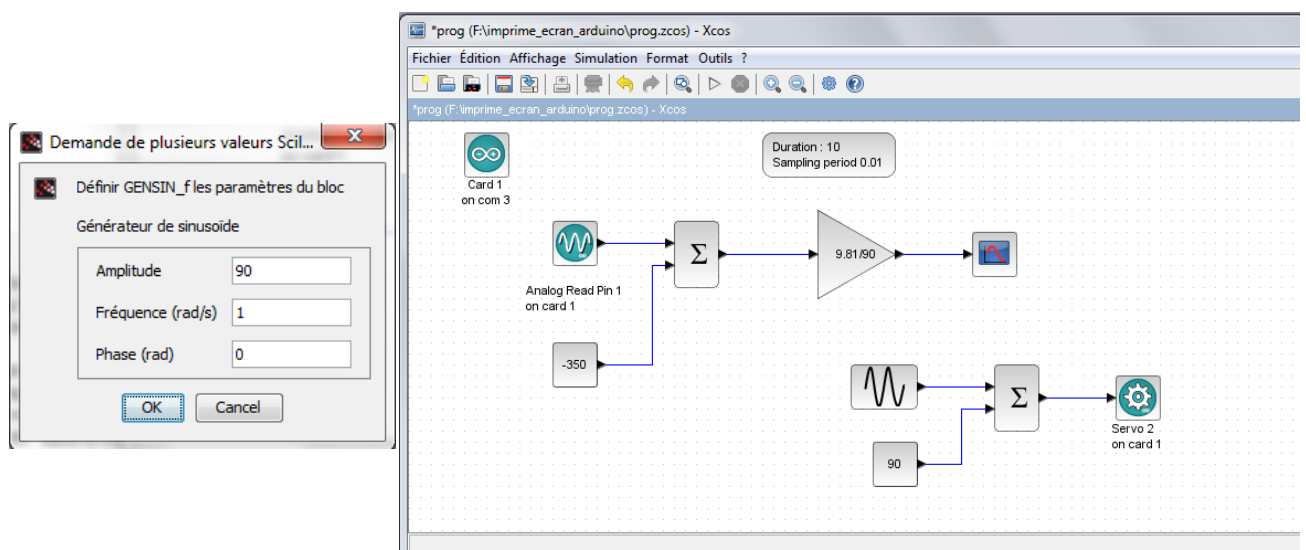
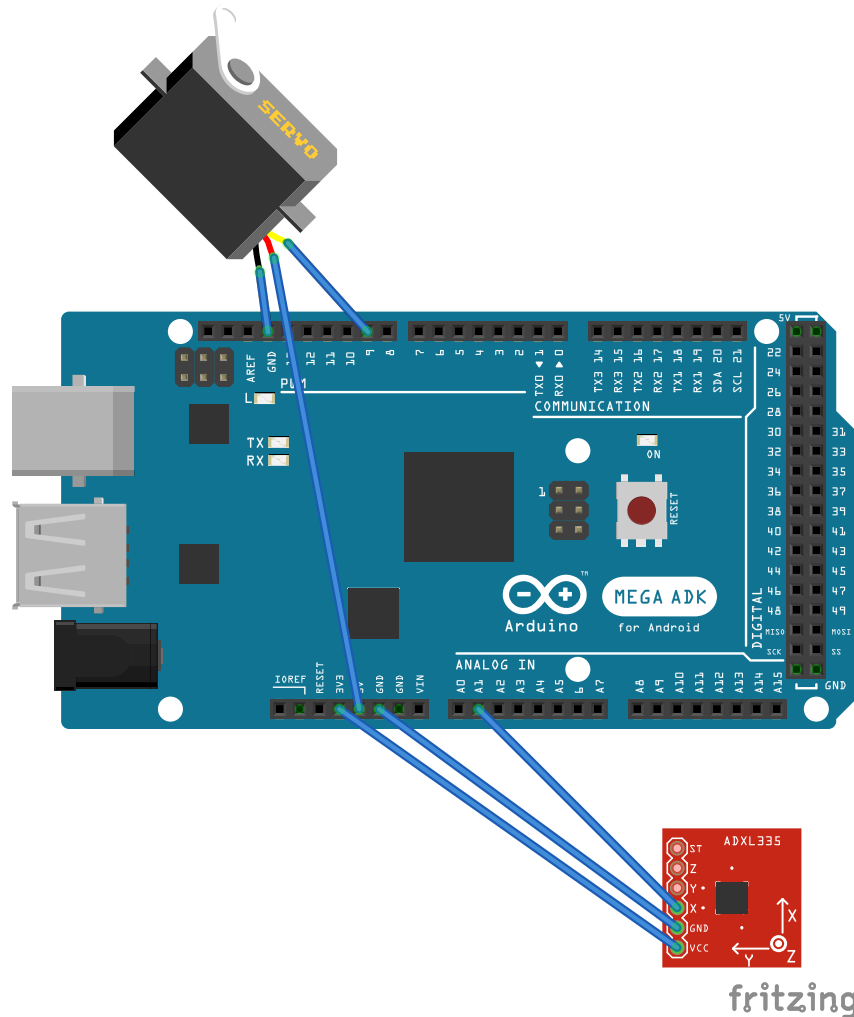


- Conversion d'une accélération en angle : La manipulation consiste à faire pivoter l'accéléromètre selon son axe (O, \vec{y}_a) . La borne X du capteur recueille en réalité la projection selon \vec{x}_a (direction horizontale associée au capteur) du champ d'accélération (notée g_x). En régime quasi-statique on mesure alors la projection de l'accélération de la pesanteur \vec{g} . On se donne un repère $R_0 O \vec{x}_0 \vec{y}_0 \vec{z}_0$ associé au référentiel terrestre supposé galiléen avec $\vec{g} = -g \cdot \vec{z}_0$. On se donne un autre repère $R_a O \vec{x}_a \vec{y}_{0,a} \vec{z}_a$. Pour obtenir l'angle d'inclinaison du capteur on a alors :

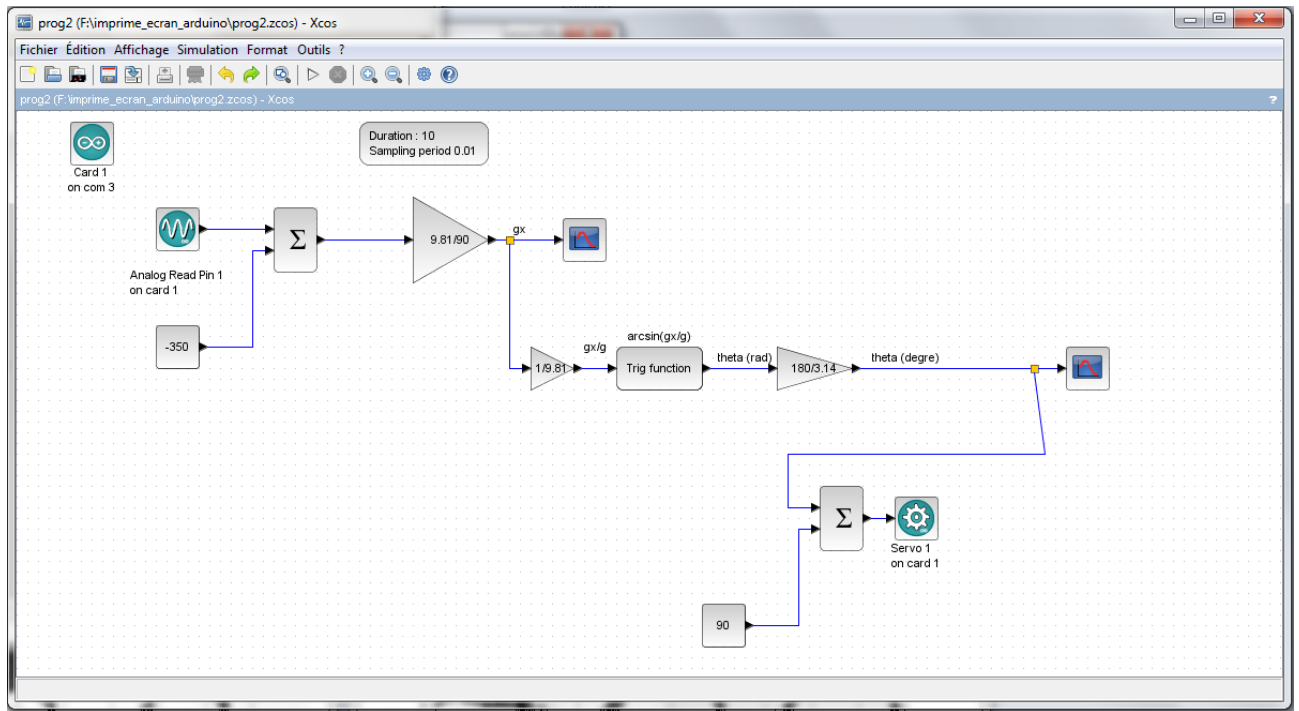
$$\theta = \arcsin\left(\frac{g_x}{g}\right)$$



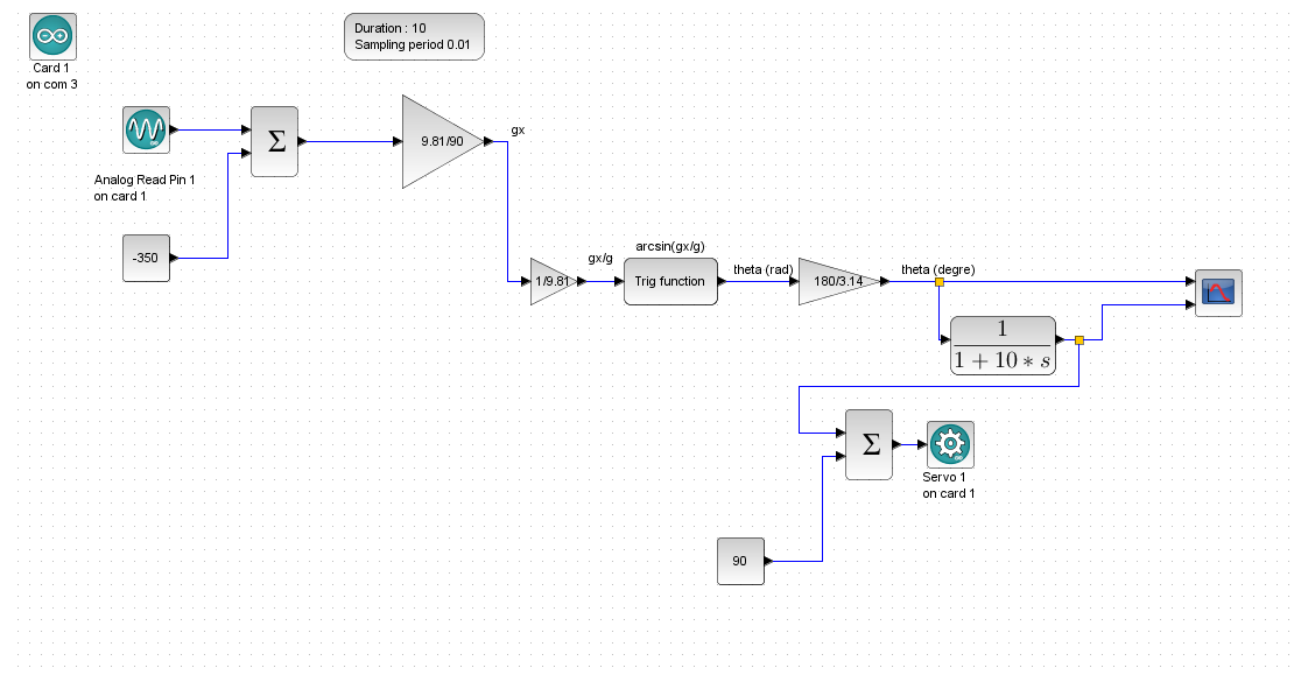
- Test de la commande d'un moteur avec une commande sinusoïdale : On peut câbler le servomoteur sur la carte arduino pour le commander :
 - fil noir du servo-moteur vers GND de la carte;
 - fil rouge du servo-moteur vers 5V de la carte;
 - fil jaune ou blanc (commande) du servo moteur vers pin 9 ou 10 (associés respectivement aux servo1 et servo2 dans scilab)
 - le servo moteur est commandé avec une entrée comprise entre 0 et 180.



- Commande d'un moteur avec accéléromètre :



- Mise en place d'un filtre (avec un premier ordre) :



8 Référence

- Le site Internet de Scilab dispose d'une rubrique consacrée à l'utilisation de Scilab pour l'enseignement [?] (<http://www.scilab.org/fr/community/education>), avec des liens et des documents utiles, dont le présent livret au format PDF, un livret destiné à l'enseignement des mathématiques, des exercices et des corrigés d'épreuves pratiques, le tout pouvant être téléchargé et imprimé librement.
- Le site de professeurs ayant largement contribué à l'écriture du livret d'utilisation de **Scilab/Xcos** [?] est également une source riche d'informations et d'exemples d'utilisation : <http://www.demosciences.fr/> [?]