Tutoriel SCILAB pour l'Automatique

April 14, 2020

1 Installation et introduction générale à SCILAB

Scilab est un logiciel libre de calcul numérique d'éveloppé par l'INRIA. Il est disponible pour Windows, Mac OS, et Linux, etc. Une introduction générale à Scilab est disponible sur https://www.scilab.org/.

L'objectif de ce document est de fournir quelques instructions de base permettant de faire de la simulation en automatique avec SCILAB.

2 Définition des données

2.1 Vecteurs

V = [0; 5; 4]; // Vecteur colonne (3*1)
V=[0, 5, 4]; // Vecteur Ligne (1*3)
V_t = V'; // Transposée du vecteur V

Matrices

A = [0, 1 ; 2, 6]; //Matrice (2*2)
A_t= A'; // Transposée de la matrice A

2.2 Scinotes

Les fichiers de type Scinote (".sce") sont des fichiers contenant du code Scilab. Il est recommandé de rédiger son programme dans un fichier ".sce" afin de conserver ses données. Pour créer un nouveau fichier Scinote, on peux utiliser le bouton "Démarrer Scinotes" (voir Figure 1).

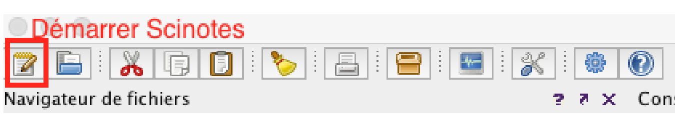

Figure 1: Démarrer Scinotes

Un example de fichier Scinotes est donné sur la Figure 2.

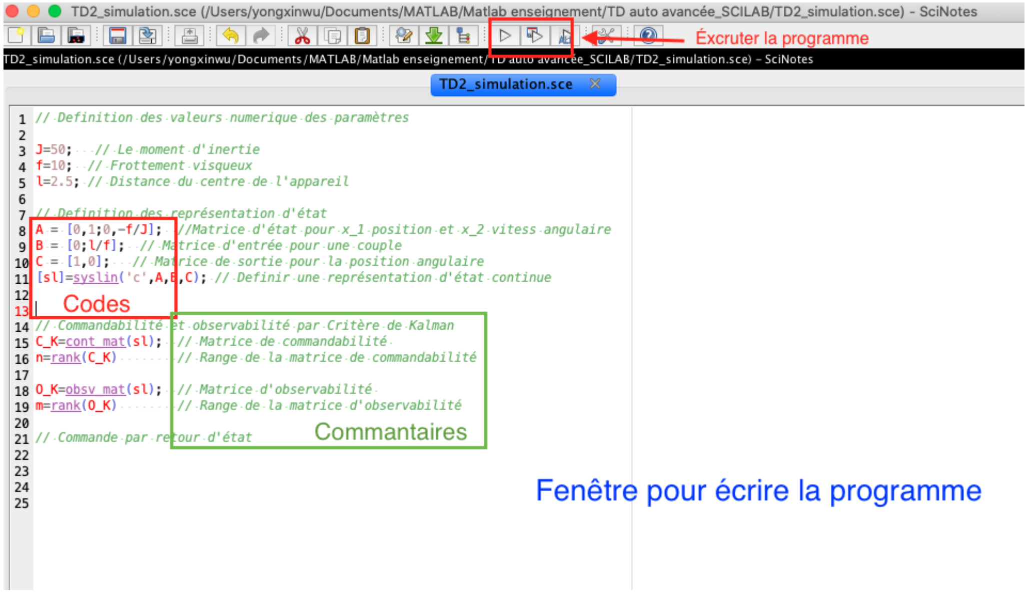

Figure 2: Scinotes

2.3 Systèmes linéaires (LTI, linear time invariant systems)

Il est possible de représenter un système linéaire de deux manières différentes: la représentation externe (fonction de transfert) et la représentation interne (représentation d'état). Dans ce qui suit on explique comment définir un LTI à l'aide de ces deux représentations.

2.3.1 Fonction de transfert

1. Par la commande “zpk”

   La fonction de transfert est définie par ses pôles, ses zéros et son gain.

   --> S=zpk(z,p,k,dt);

   où z, p, k sont les vecteurs des pôles, des zéros et le gain statique reliant l'entrée à la sortie. “dt” est utilisé pour définir si le système est en temps continu ('c') ou en temps discret ('d').

   Exemple : On considère le système continu défini par la fonction de transfert

   $$G(p) = \frac{5(p+2)(p+3)}{(p+1)(p+5)} \quad (1)$$

   Les pôles du système sont $p = [-1, -5]$, les zéros sont $z = [-2, -3]$, et $k = 5$. Pour définir le système $G(p)$, on peut donc utiliser le code suivant:

   --> p=[-1,-5]; //définit les pôles

   --> z=[-2,-3]; //définit les zéros

   --> k=5; // définit le gain statique entrée/sortie

   --> S=zpk(z,p,k,'c'); //définit que le système est continu ('c')

2. Par la commande “syslin”

   La fonction de transfert est définie à l'aide de polynômes en utilisant les syntaxes suivantes.

   --> S=syslin(dt,N,D)

   --> S=syslin(dt,H)

   où N, D sont les numérateur et dénominateur; H représente la fraction rationnelle associée à la fonction de transfert.

   Exemple: On considère le système continu défini par la fonction de transfert

$$G(p) = \frac{2p+1}{p^2+5p+6} \tag{2}$$

pour définir ce système sous Scilab, on peux utiliser les codes suivants:

--> s = poly(0,'s'); // Définit la variable de Laplace

--> H = (2*s + 1) / (s^2 +5*s +6); //Définit le polynôme

--> S=syslin('c',H); //Définit la fonction de transfert liée à H

2.3.2 Représentation d'état

1. Par la commande "syslin"

   Un système LTI peut être défini sous forme d'état par le biais de la commande "syslin" de la manière suivante:

   -->SL=syslin(dt,A,B,C,[D[,x0]])

   où A est la matrice d'état, B est la matrice d'entrée, C est la matrice de sortie, D est la matrice de transmission directe entre l'entrée et la sortie, x0 est la condition initiale du système. Les deux derniers termes ne sont pas obligatoires d si D=0 et si la condition initiale est nulle.

   Exemple: Le système

$$\left\{ \begin{array}{rcl} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} & = & \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y & = & \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{array} \right. \tag{3}$$

   peut être défini par

   --> A = [-1,1;0,-2]; //Matrice d'état
   --> B = [0;1]; // Matrice d'entrée
   --> C = [1,0]; // Matrice de sortie
   --> SL=syslin('c',A,B,C); // Definit une représentation d'état continue avec D=0 et x0=0

3 Les commande utiles

3.1 Analyse du système

1. Diagramme de bode : bode

   -->bode(nom du système)

2. Diagramme de Nyquist

   -->nyquist(nom du système)

3. Diagramme black

   --> black(nom du système)

4. Stabilité

   Pour analyser la stabilité du système, on cherche les pôles du système.

   --> plzr(nom du système)

   Cette fonction permet d'afficher les pôles et les zéros du système

   --> roots(Numérateur ou dénominateur de la fonction de transfert)

   Cette fonction calcule les racines d'un polynôme.

3.2 Réponse du système

Pour tracer la réponse du système à différentes consignes, on utilise la commande "csim" de la manière suivante:

--> [y [,x]]=csim(u,t,sl,[x0 [,tol]])

où, y est le vecteur de sortie, x est le vecteur d'état, u est l'entrée, t est le vecteur temps, sl est le système défini par avance, x0 et tol sont les condition initiale du système et la tolérance pour le solveur. Les paramètres entre crochets ne sont pas obligatoires.

Exemple : Tracer la réponse indicielle du système défini dans l'exemple précédent.

-->t=0:0.05:20; // Définit le vecteur de temps

-->y=csim('step',t,SL) ; //Calcule la réponse indicielle

-->plot(t,y); // Trace la réponse

La réponse est donnée par la courbe suivante

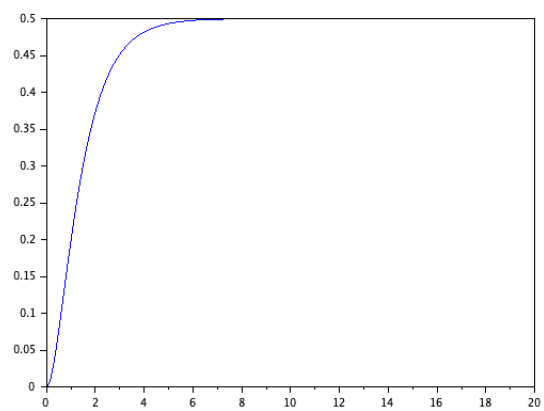

Figure 3: Réponse indicielle

Tips : L'entrée u peut être définie comme une function ou une chaîne de caractères (string). On peut utiliser la commande 'step' pour la réponse indicielle ou 'impulse' pour la réponse à une impulsion.

4 Commande du système décrit par la fonction de transfert

Pour commander le système linéaire décrit par une fonction de transfert, on peut utiliser l'interface visuelle proposé par Scilab –Xcos. Xcos peut être lancée à l'aide du boton sur Scilab, Cf. Figure 4.

Figure 4: Lancer Xcos

La Figure 5 contient un exemple d'asservissement d'un système 2nd ordre (TD2) à l'aide d'un correcteur PID.

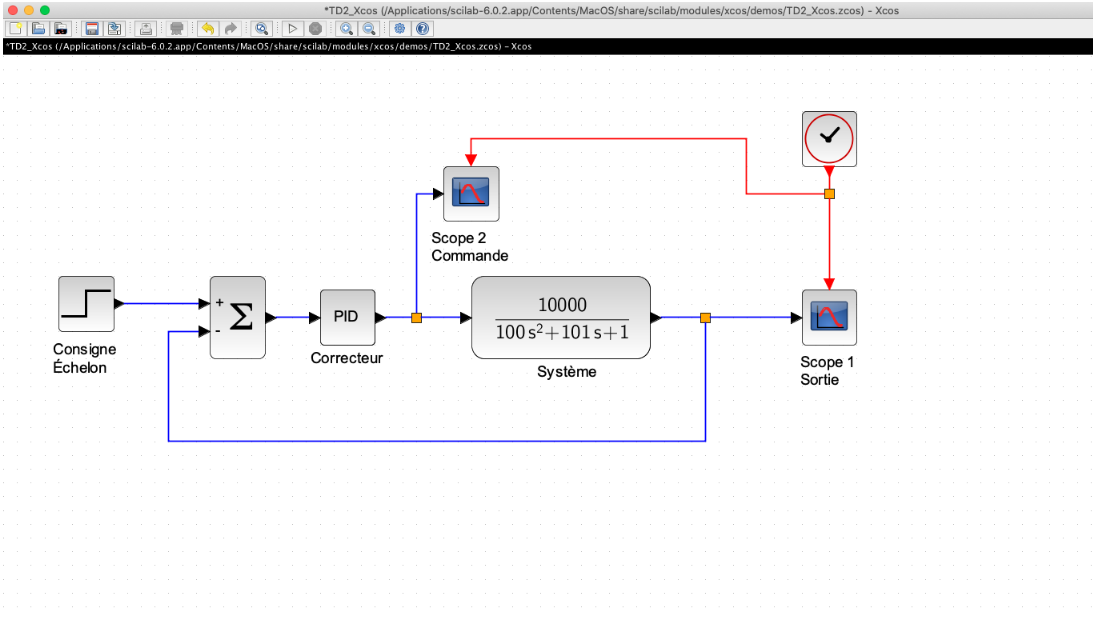

Figure 5: Xcos interface

On peut trouver tous les blocs permettant de construire le schéme bloc dans le navigateur de Xcos (cf Figure 6).

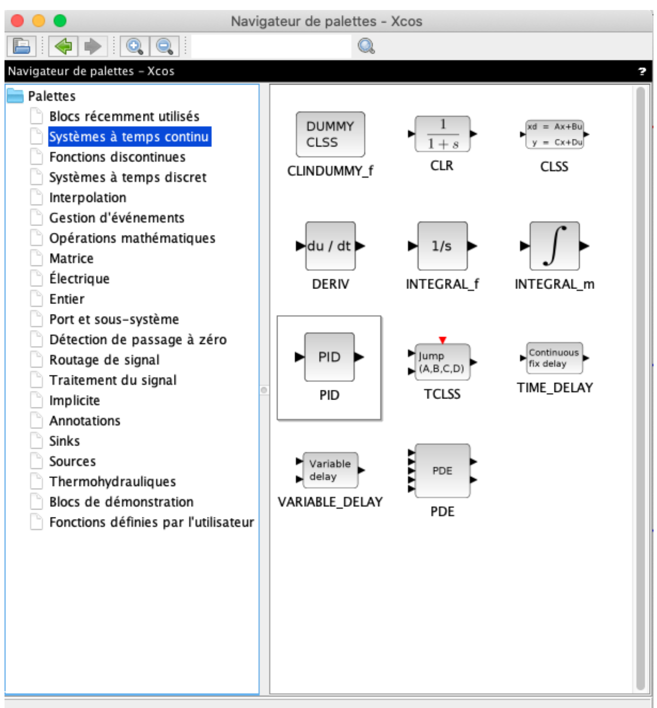

Figure 6: Navigateur Xcos

Tips: On peut utiliser clic droit pour changer les paramètres de chaque bloc. Clic droit sur un endroit vide dans la fenêtre de Xcos et puis choisir “Configurer” permet de changer le temps d'intégration final (le temps de simulation).

5 Commande par retour d'état

On considère le système commandable décrit par la représentation d'état

$$\dot{x} = Ax + Bu \tag{4}$$

On veut utiliser la commande par retour d'état

$$u = -Kx + Hy_c \tag{5}$$

afin de placer les pôles en P, P représentant les pôles associés à la dynamique désiré (cf. cahier des charges) du système en boucle fermée.

La fonction Scilab pour calculer K permettannt de placer ces pôles est "ppol"

-->K=ppol(A,B,K)

La matrice d'état du système en boucle fermée est

$$A_{BF} = A - BK \tag{6}$$

Ensuite, on doit calculer la pré-commande statique H telle que le gain statique du système en boucle fermée est égal à 1 ou l'erreur statique est nulle. Par application du Théorème de la Valeur Final, on sait que

$$H = \frac{1}{C\left(-A_{BF}\right)^{-1}B} \tag{7}$$

Donc, sous Scilab, on peut calculer H par

-->H=1/(C*inv(-Abf)*B)

où "inv" est la fonction Scilab qui calcule l'inverse d'une matrice (c'est la même fonction sous MATLAB). On peut maintenant définir le système en boucle fermée par le biais des matrices $[A - BK, BH, C, D]$ sous Scilab.

Il est aussi possible de simuler le système en boucle fermée à l'aide de Xcos. Une fois que l'on a défini toutes les matrices : A, B, C, D, K, H, on peut passer à l'interface Xcos et construire un schème de simulation comme celui de la Figure 7.

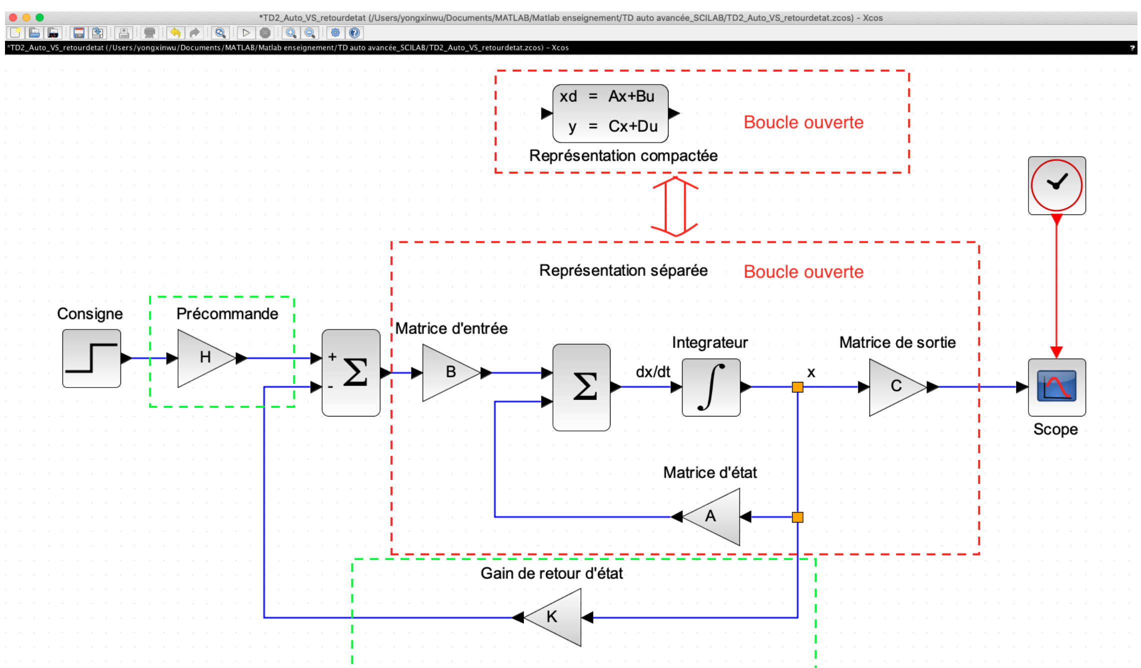

Figure 7: Xcos Commande par retour d'état

Dans la Figure 7 apparaissent deux boucles ouvertes. C'est juste pour illustrer qu'il y a différentes façon de construire la représentation d'état du système.

À réfléchir : Est-il possible d'utiliser le bloc boucle ouverte compacté pour faire la commande par retour d'état ? Si oui, comment faire? Si non, pourquoi?