

TP N°1

Introduction au MATLAB et Simulink

1. objectif de TP

Le but de TP est l'initier l'étudiant au logiciel de calcul numérique MATLAB ainsi que l'outil de simulation associé Simulink et de découvrir les fonctions et outils de Matlab relatifs à l'étude des systèmes asservis.

2. Présentation et introduction à Matlab :

Matlab, abréviation de “**MAT**rix **LAB**oratory”, est un logiciel conçu pour fournir un environnement de calcul numérique de haut niveau. Il est particulièrement performant pour le calcul matriciel, et dispose de grandes capacités graphiques pour, par exemple, la visualisation d'objets mathématiques complexes.

Son fonctionnement repose sur un langage de programmation interprété qui permet un développement très rapide. En contrepartie, pour des applications nécessitant des performances plus élevées en temps de calcul, un langage compilé, comme le C++ ou le fortran, est plus adapté. Sous sa forme “graphique”, Matlab dispose d'une interface comprenant l'environnement Matlab à proprement parler, d'où les commandes Matlab peuvent être directement exécutées, ainsi que d'un environnement graphique, pouvant comprendre plusieurs fenêtres : liste des variables en cours d'utilisation, historique des commandes exécutées, ..., et divers menus plus ou moins habituels, “File”, (“New”, “Open”...), “Configuration”, “Help”,...

Dans ce paragraphe d'introduction à Matlab, on ne s'intéressera qu'à l'environnement propre à Matlab, les commandes et syntaxes de base d'instructions Matlab.

3. Commandes générales

Help pour demander l'aide

Cd change/affiche le répertoire courant

who liste des variables de l'espace de travail

clear var supprime la variable var de l'espace de travail

clear all supprime toutes les variables

Close all ferme tous les graphiques

Opérations usuelles

= affectation d'une valeur à une variable (ex. $a = 2$)

+, -, *, / opérations usuelles sur les variables ou valeurs numériques

1i nombre complexe : $(1i)^2 = -1$

Abs, angle, real, imag opérations usuelles sur les nombres complexes

4. Définition et opérations sur les vecteurs et matrices

Définition de matrices et vecteurs :

[, , ; , ,] définition manuelle d'une matrice (ex. $A = [1, 2, 3; 4, 5, 6]$)

deb:pas:fin définition d'un vecteur régulier balayant l'intervalle [deb, fin] avec le pas (ex. $A = 1 : 1 : 6$) ; par défaut le pas est égal à 1 s'il est omis (ex. $A = 1 : 6$)

linspace (deb,fin,N) définition d'un vecteur balayant l'intervalle [deb; fin] avec N valeurs régulièrement espacées

Les opérations usuelles +, -, *, /, ... agissent indifféremment sur les réels, complexes, ou matrices (à conditions que les dimensions de celles-ci le permettent).

– Addition: $A+B$

– Soustraction: $A-B$

– multiplication : $A*B$ et $B*A$

– inversion : $\text{inv}(A)$

– transposition : $\text{transpose}(A)$ ou A'

D'autres opérations sont également disponibles :

.*, ./, .^, ... (Les opérateurs usuels précédés d'un point) opérations sur les matrices effectuées terme à terme (ex. calculer $A = [1 : 6].* [7 : 12]$)

length longueur d'un vecteur

size dimension d'une matrice

<, <=, >, >=, == comparaison des éléments de deux matrices, terme à terme

sum, mean, ... somme, moyenne, ..., des éléments d'un vecteur

sin, cos, exp, log, ... d'une façon générale toutes les fonctions usuelles s'appliquent à des matrices terme à terme (ex. $\log([1, 2, 3]) = [\log 1, \log 2, \log 3]$)

find recherche les éléments non nuls dans une matrice

nnz compte le nombre d'éléments non nuls dans une matrice

Quelques matrices utiles

– matrice identité de dimension n : $\text{eye}(n)$

– matrice de zéros de dimension $m \times n$: $\text{zeros}(m,n)$

– matrice de uns de dimension $m \times n$: $\text{ones}(m,n)$

Etude des matrices

– valeurs propres : $\text{eig}(A)$

– rang : $\text{rank}(A)$

– trace : $\text{trace}(A)$: la somme des éléments diagonaux de la matrice

– déterminant : $\text{det}(A)$

Extraction des éléments d'une matrice

M(i,j) élément de la matrice M situé sur la ligne i et la colonne j

V(end) dernier élément du vecteur V

M(5:9,3) les éléments de la matrice M situés de la ligne 5 à 9, et sur la colonne 3

M(:,j) toutes les lignes de la matrice M, colonne j

M(i,:) toutes les colonnes de la matrice M, ligne i

M(1:5,1:3) les éléments de la matrice M situés sur les lignes 1 à 5, et sur les colonnes 1 à 3

5. Fonctions graphiques :

Figure crée une nouvelle figure ; **figure(n)** (ré) initialise la figure n° n

plot tracé d'un ensemble de points (ex. `plot([0:0.1:2*pi],sin([0:0.1:2*pi]))`)

subplot partitionne la figure courante en plusieurs sous-graphiques

axis permet de sélectionner manuellement l'échelle utilisée

title, xlabel, ylabel, legend : permet d'ajouter un titre général, un titre sur les axes et une légende à une figure

grid on / off affiche une grille sur le graphique courant

hold on /off permet de superposer des graphiques sur une même figure

plot3, semilogx, semilogy, loglog, mesh, surf,... autres fonctions graphiques (voir l'aide...)

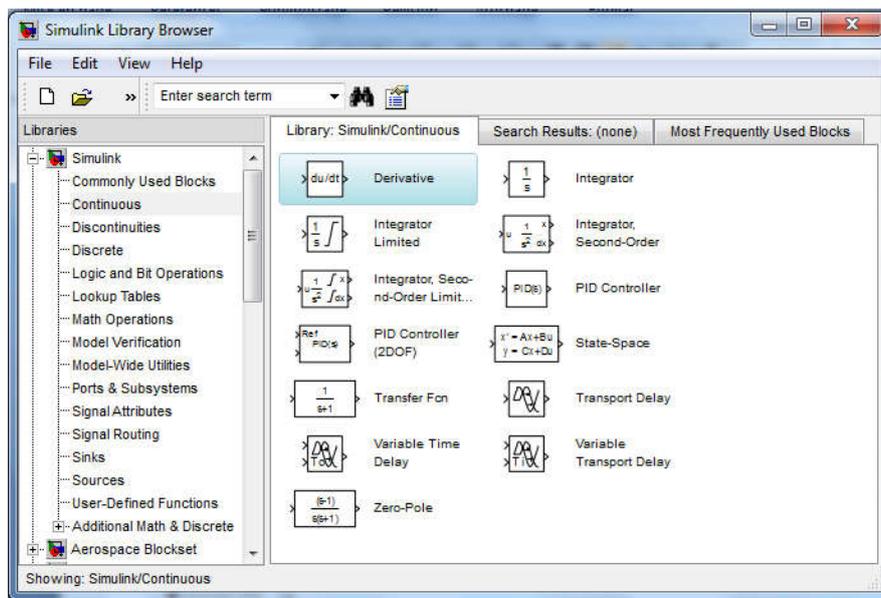
6. Introduction à Simulink :

Simulink est l'extension graphique de MATLAB permettant de représenter les fonctions mathématiques et les systèmes sous forme de diagramme en blocs, et de simuler le fonctionnement de ces systèmes.

POUR DÉMARRER SIMULINK

Dans la fenêtre Commande de MATLAB, taper **Simulink**. La fenêtre Simulink va s'ouvrir.

La librairie de **Simulink** comprend de différentes sections permettant d'aborder de nombreux aspects de la commande des systèmes. Dans un premier temps, on portera notre attention sur quelques sections montrées ci-dessous :



➔ « **Continuous** » : permettant de sélectionner des blocs fonctionnels modélisant des éléments linéaires et notamment les blocs fonctions de transferts continus.

- ➔ « **Discrete** » : contient des modèles discrets pour construire des systèmes à temps discret.
- ➔ « **Math Operations** » : permettant de sélectionner des blocs réalisant des opérations mathématiques et notamment le bloc **Add**.
- ➔ « **Signal Routing** » : permettant différents aiguillages des données sur les fils du modèle ; on y trouve notamment le bloc **Mux** qui permet de multiplexer plusieurs données dans un seul fil.
- ➔ « **Sources** » : générateurs de signaux (échelons, carré, sinus, etc...).

« **Sinks** »: qui permet de trouver des blocs fonctionnels de sorties possibles du fichier modèle que l'on simule. On y trouve notamment le bloc fonctionnel « **To Workspace** » qui permet d'accéder aux données dans l'espace de travail de Matlab.

CONSTRUCTION D'UN DIAGRAMME SIMULINK

Pour commencer, dans le menu File, on choisit **New - Model**. Une fenêtre de travail Untitled s'ouvrira.

Ouvrir les collections de blocs en cliquant dessus (double). Faire glisser dans la fenêtre de travailles blocs dont on a besoin pour construire le diagramme. Faire des liaisons entre les blocs à l'aide de la souris. Lorsqu'on clique (double) sur un bloc, une fenêtre de dialogue s'ouvrira. On peut alors changer les paramètres de ce bloc. Une fois terminé, on ferme la fenêtre de dialogue. Une fois le diagramme terminé, on peut l'enregistrer dans un fichier: dans le menu File, choisir Save As et donner un nom (*.mdl) au fichier.

Exemple de diagrammes Simulink:

Reproduisez le schéma de la figure 1 pour comprendre le fonctionnement de cet outil.

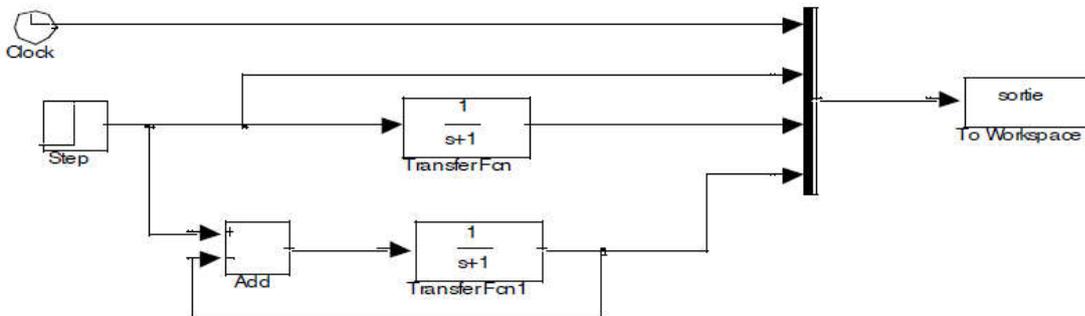


Figure 1: Exemple de modèle Simulink.

Dans, l'exemple proposé ci-dessus et après simulation effective du fichier modèle, on disposera sous Matlab d'une variable nommée sortie, comprenant 4 colonnes :

- sortie(:,1) : les instants pour lesquels la simulation a été effectuée.
- sortie(:,2) : l'entrée indicielle appliquée
- sortie(:,3) : la sortie du système qui a pour fonction de transfert $1/(s+1)$
- sortie(:,4) : la sortie du système bouclé, retour unitaire correspondant.

Les paramètres de la simulation sont accessibles dans la rubrique simulation de la barre d'outils.