
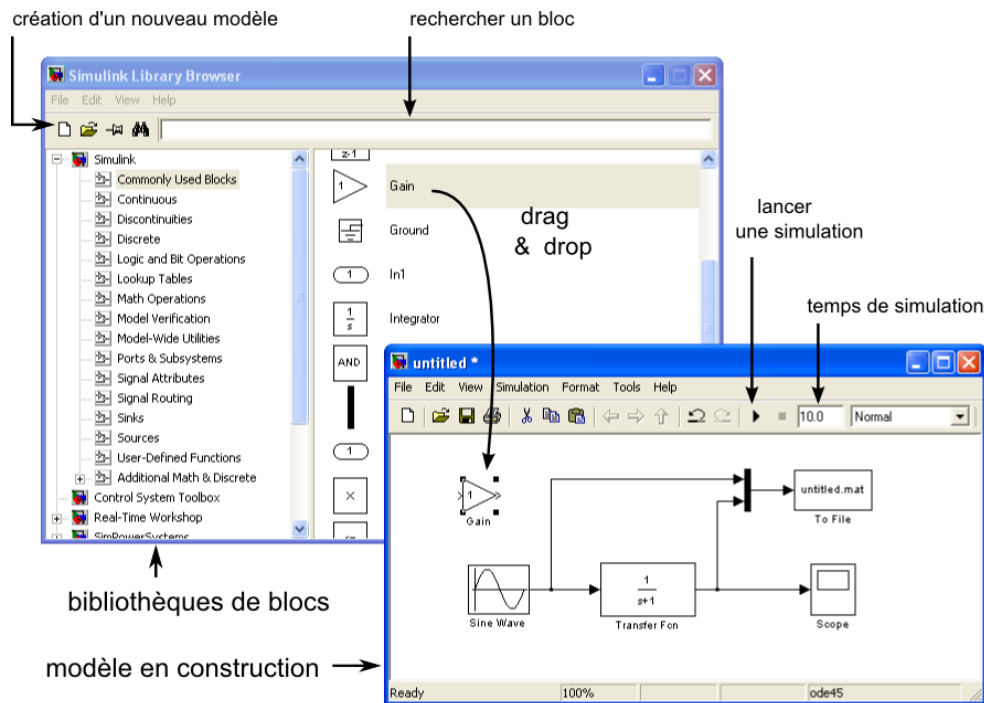


MEMO SIMULINK

Simulink¹ est un logiciel muni d'une interface graphique pour la modélisation, la simulation et l'analyse des systèmes. Etant intégré à MATLAB¹, les deux environnements sont parfaitement compatibles et les différentes fonctionnalités de ce dernier sont alors directement accessibles.

Simulink peut être lancé depuis l'environnement de MATLAB

- en cliquant, dans la barre d'outils, sur le bouton 
- ou en tapant `simulink`

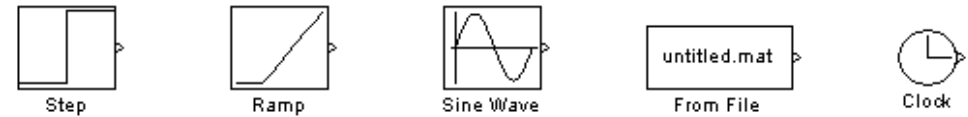


Les modèles, une fois sauvegardés, sont des fichiers d'extension « `.mdl` ».

L'interface est intuitive, un modèle est construit à partir de blocs fonctionnels par glisser-déposer. Ceux-ci sont rangés par « familles » dans différentes bibliothèques. Voici une liste, organisée par bibliothèques, des quelques blocs souvent utilisés.

SOURCES

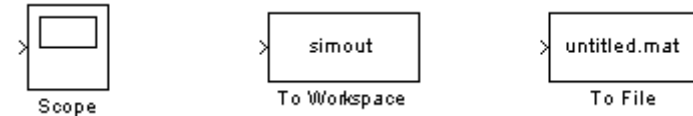
Ensemble de blocs pour la génération de signaux



Signal échelon ; signal rampe ; signal sinusoïdal ; valeurs fournies par un fichier ; temps.

SINKS

Ensemble de blocs pour l'affichage ou la mémorisation de signaux



Affichage type oscilloscope ; stockage dans une variable ; stockage dans un fichier.

MATH OPERATIONS

Ensemble de blocs réalisant une fonction mathématique appliquée aux signaux entrants.

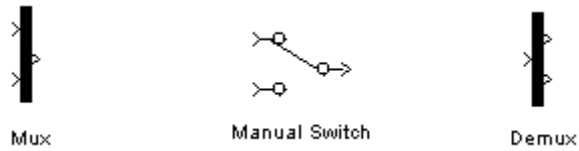


Somme/soustrait deux signaux ; multiplie un signal ; fournit le module et la phase ; applique une fonction standard.

¹ Matlab et Simulink sont des produits développés par la société The MathWorks, Inc.. Matlab® et Simulink® sont des marques déposés par cette même société.

SIGNAL ROUTING

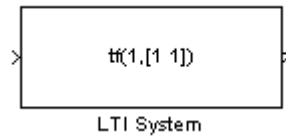
Ensemble de blocs pour l'aiguillage de signaux.



Multiplexeur: combine deux entrées en une seule (vecteur) ; switch: permet de sélectionner manuellement une entrée; démultiplexeur: sépare une entrée (vecteur) en plusieurs composantes.

CONTROL SYSTEM TOOLBOX

Bloc « LTI system » pour la définition de fonctions de transfert.



Une fonction de transfert se définit par l'instruction $\text{tf}(a, b)$, où a et b sont des vecteurs composés respectivement des coefficients du numérateur et du dénominateur. Par exemple, définissons la fonction de transfert

$$G(p) = \frac{p+2}{3p^2+4p+5} \implies \text{tf}([1 \ 2], [3 \ 4 \ 5]).$$

Il est conseillé de définir la FT dans MATLAB : $G=\text{tf}([1 \ 2], [3 \ 4 \ 5])$, puis de spécifier son nom G dans le bloc « LTI system ». Enfin, pour définir une fonction de transfert en Z, un 3^e argument doit être ajouté: T_e la période d'échantillonnage. Par exemple

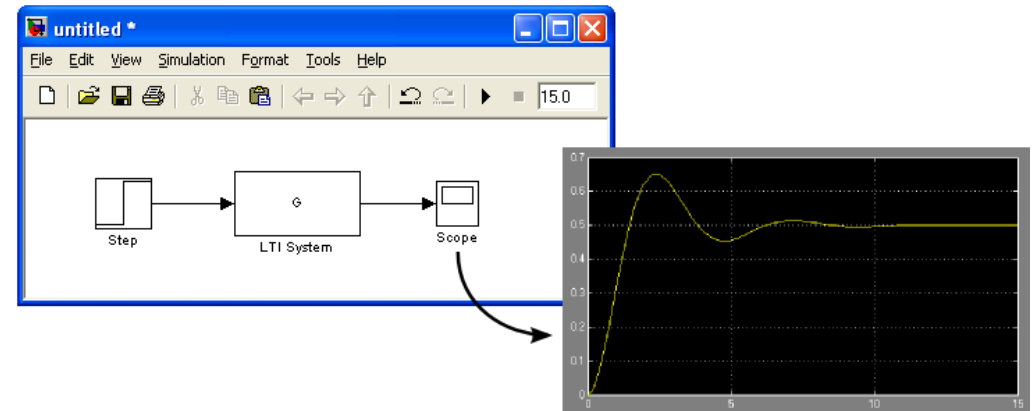
$$G(z) = \frac{z}{z-0.5} \implies \text{tf}([1 \ 0], [1 \ -0.5], 2)$$

pour une période $T_e=2s$.

Exemple

Considérons un système modélisé par la fonction de transfert suivante : $G(p) = \frac{1}{p^2+p+2}$

Définissons celle-ci dans MATLAB: $G = \text{tf}(1, [1 \ 1 \ 2])$. Construisons un modèle Simulink afin d'observer la réponse du système à une entrée de type échelon unitaire. La simulation est effectuée sur un horizon temporel de 15s. Après l'exécution de celle-ci, la réponse peut être observée en double-cliquant sur l'oscilloscope.



Réalisons maintenant un asservissement avec une commande proportionnelle de gain K . Notre nouveau modèle de simulation nous permettra de tester les performances de l'asservissement pour différentes valeurs de K . Il peut être fixé directement depuis le bloc ou via MATLAB. Ici le multiplexeur permet d'afficher sur un seul graphe le signal de consigne et le signal de sortie.

