UTILISATION DE MATLAB/SIMULINK POUR L'ÉTUDE DES SYSTÈMES DYNAMIQUES

A. Desbiens et É. Poulin, 17 décembre 2014

1. Commandes Matlab de base 1
Aide via la fenêtre de commande 1
Fonctions utiles liées à l'espace de travail 1
Définition d'une matrice 2
Accès aux éléments d'une matrice 2
Opérations de base sur les matrices ou sur ses éléments 2
Fonctions utiles en algèbre linéaire 2
Fonctions graphiques utiles 3
Création d'un script 3
Création d'une fonction 3
2. Analyse des systèmes linéaires 5
a) Avec Simulink 5
b) Avec LTI Viewer 7
c) Avec des commandes Matlab 12
Réponses temporelles et fréquentielles 12
Décomposition d'une fraction rationnelle en éléments simples 14
Transformation de Laplace 14

1. Commandes Matlab de base

Aide via la fenêtre de commande

doc	Démarre l'aide de Matlab sous forme HTML
help	Présente les différents sujets d'aide regroupés par catégories
help function	Donne de l'aide sur la fonction spécifiée
help help	Donne de l'aide sur l'utilisation de l'aide
lookfor keyword	Recherche le mot-clé dans les différents fichiers de type *.m

Fonctions utiles liées à l'espace de travail

clc	Efface la fenêtre de commande
clear	Efface les variables et les fonctions en mémoire
load	Charge les variables à partir d'un fichier
pack	Réorganise l'espace mémoire
save	Sauvegarde les variables de l'espace de travail

Définition d'une matrice

s = 4	Scalaire (nombre réel)
s = 3 + 4*i	Scalaire (nombre complexe)
v = [1 2 3 4]	Vecteur ligne de dimension (1,4)
v = [1;2;3;4]	Vecteur colonne de dimension (4,1)
v = 0:2:10	Vecteur ligne [0 2 4 6 8 10]
m = [1 2 3;4 5 6;7 8 9]	Matrice de dimension (3,3)
m = [v;4*2 7 6 3*i;8 2 4 6]	Matrice de dimension (3,4), v étant un vecteur ligne de dimension (1,4)
m = zeros(2,3)	Matrice de dimension (2,3) dont tous les éléments sont des zéros
m = ones(2,3)	Matrice remplie de dimension (2,3) dont tous les éléments sont des uns
m = eye(4)	Matrice identité de dimension (4,4)

Accès aux éléments d'une matrice

v(3)	Élément d'un vecteur (troisième élément)
m(2,3)	Élément d'une matrice (élément situé ligne 2, colonne 3)
v(2:3)	Section d'un vecteur (deuxième au troisième élément)
m(2:3,3:4)	Section d'une matrice (lignes 2 à 3 et colonnes 3 à 4)
m(:,2)	Colonne d'une matrice (deuxième colonne)
m(2,:)	Ligne d'une matrice (deuxième ligne)

Opérations de base sur les matrices ou sur ses éléments

m + n	Addition de deux matrices
m - n	Soustraction de deux matrices
m * n	Multiplication de deux matrices
m'	Transposée d'une matrice
m.*n	Multiplication terme à terme des éléments de deux matrices
m./n	Division terme à terme des éléments de deux matrices
m.^2	Mise au carré des éléments d'une matrice

Fonctions utiles en algèbre linéaire

det	Calcule le déterminant d'une matrice
dot	Calcule le produit scalaire de deux vecteurs
eig	Détermine les valeurs propres et les vecteurs propres d'une matrice
inv	Calcule l'inverse d'une matrice
length	Donne la longueur d'un vecteur
lu	Effectue la factorisation LU d'une matrice

null	Retourne la base orthonormée pour l'espace nul d'une matrice
qr	Effectue la factorisation QR d'une matrice
rref	Retourne la forme échelonnée réduite par rapport aux lignes d'une matrice
size	Donne la dimension d'une matrice

Fonctions graphiques utiles

axis	Permet la gestion des axes
figure	Ouvre une fenêtre graphique
grid	Permet la gestion du quadrillage d'un graphique
plot	Trace un graphique
semilogx	Trace un graphique semi-logarithmique
subplot	Fractionne une fenêtre pour insérer plusieurs graphiques
title	Ajoute un titre au graphique courant
xlabel	Ajoute une étiquette à l'axe des x du graphique courant
ylabel	Ajoute une étiquette à l'axe des y du graphique courant

Création d'un script

Il est possible d'exécuter un script à partir de la fenêtre de commande en tapant le nom du fichier (sans l'extension). Cela permet d'éviter l'entrée manuelle de chacune des commandes. Le contenu du fichier nommé « tracedroite.m » est donné ci-dessous à titre d'exemple. La commande « ctrl+c » permet d'arrêter l'exécution d'un script.

```
% Il est possible de mettre des commentaires en mettant le caractère
pourcentage en début de ligne

x = (-10:0.1:10)';
y = 2*x+1;

plot(x,y, 'r');
```

Création d'une fonction

En créant un fichier nommé « addsous.m » dont le contenu est présenté ci-dessous, il est possible d'appeler la fonction à partir de la fenêtre de commande, d'un script ou d'une autre fonction. La commande « ctrl+c » permet d'arrêter l'exécution d'une fonction.

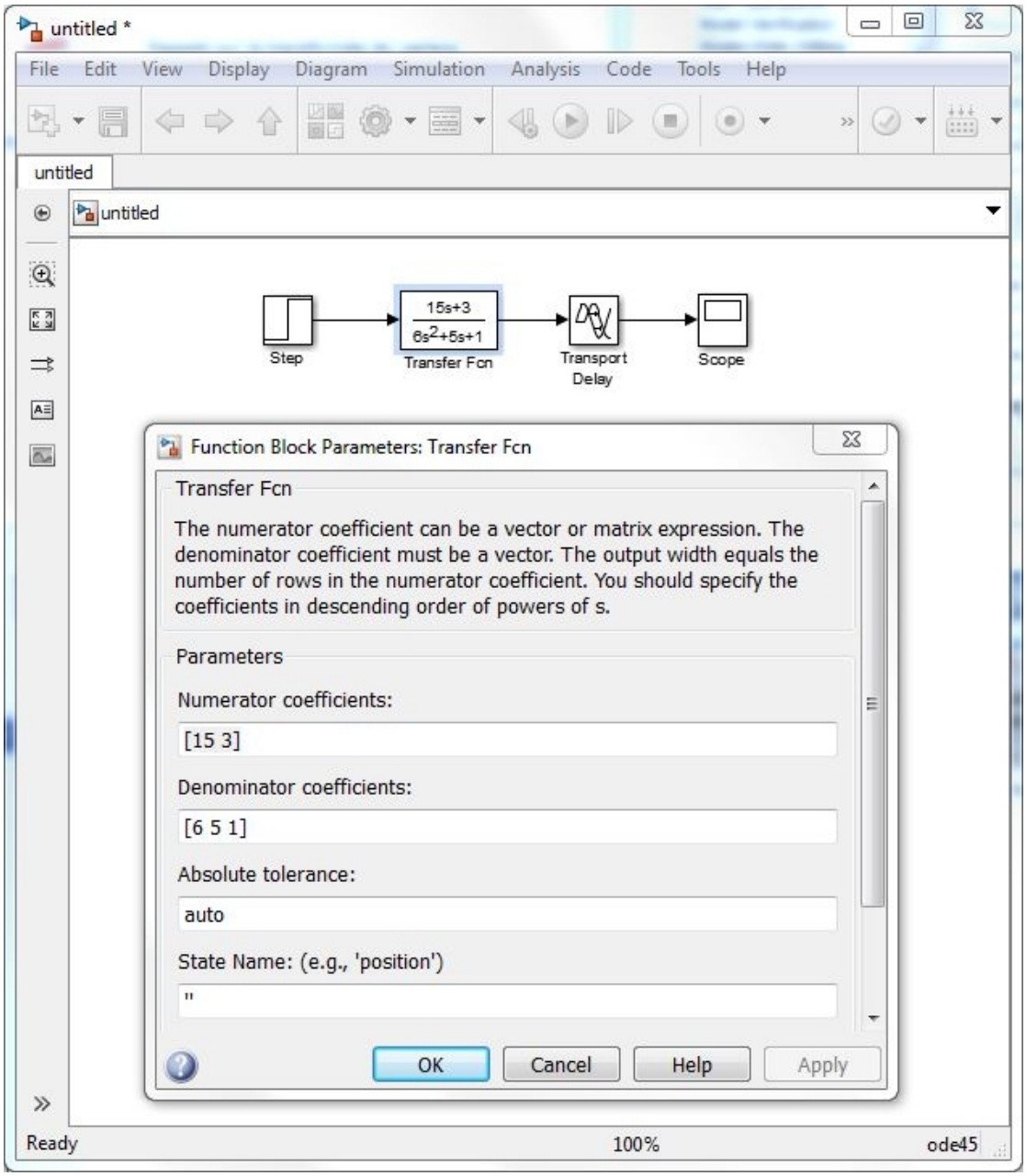
```
function [a, b] = addsous(x,y)

% Les premières lignes de commentaires sont affichées via la commande
% help addsous dans la fenêtre de commande
```

```
a = x + y;
b = x - y;
```

2. Analyse des systèmes linéaires

a) Avec Simulink

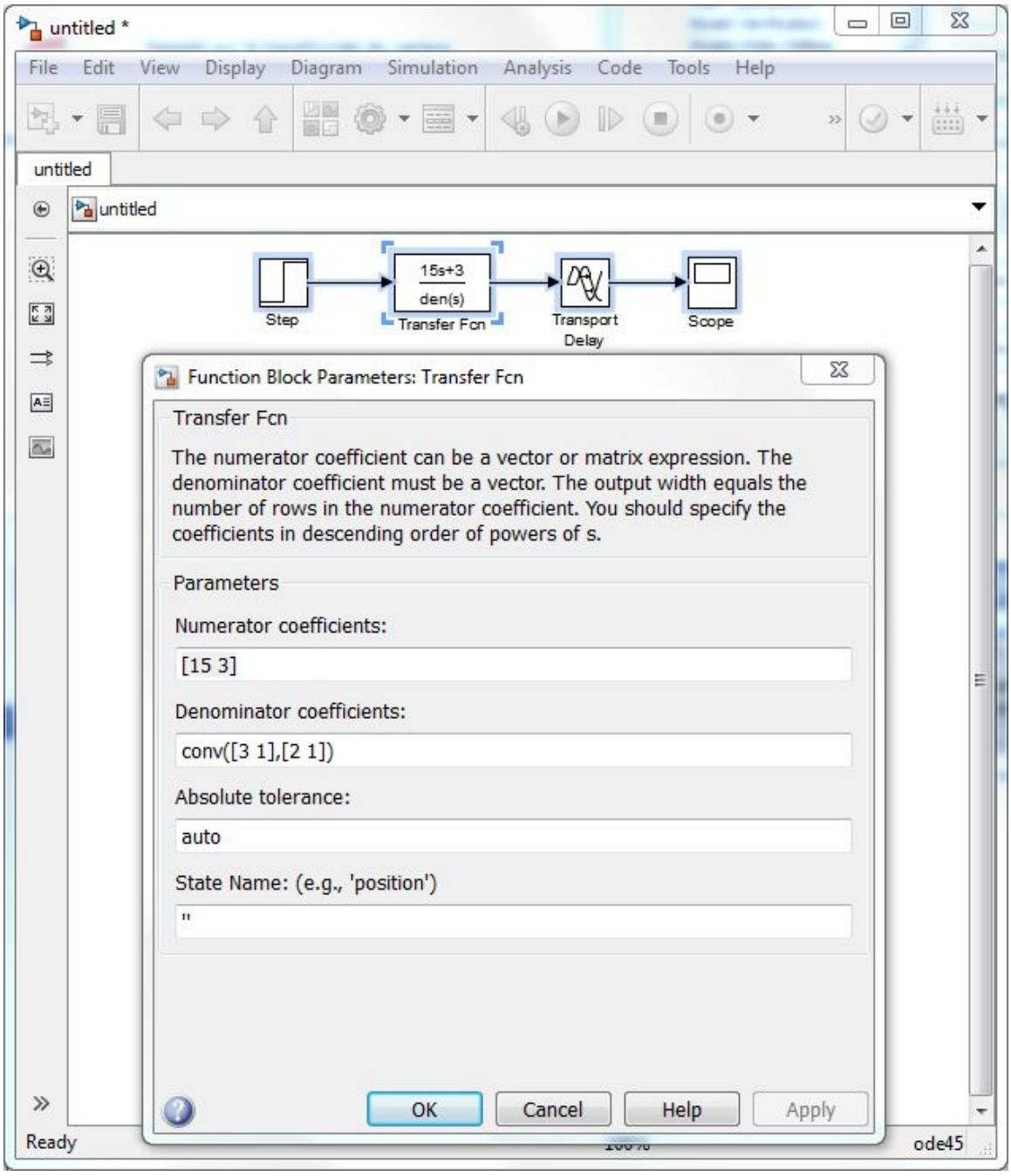
Exemple 1:



Note 1: Remarquez comment les paramètres du bloc "Transfer Fcn" sont entrés. Les polynômes du numérateur et du dénominateur apparaissent comme des vecteurs dont les éléments sont les chiffres multipliant les puissances décroissantes de "s".

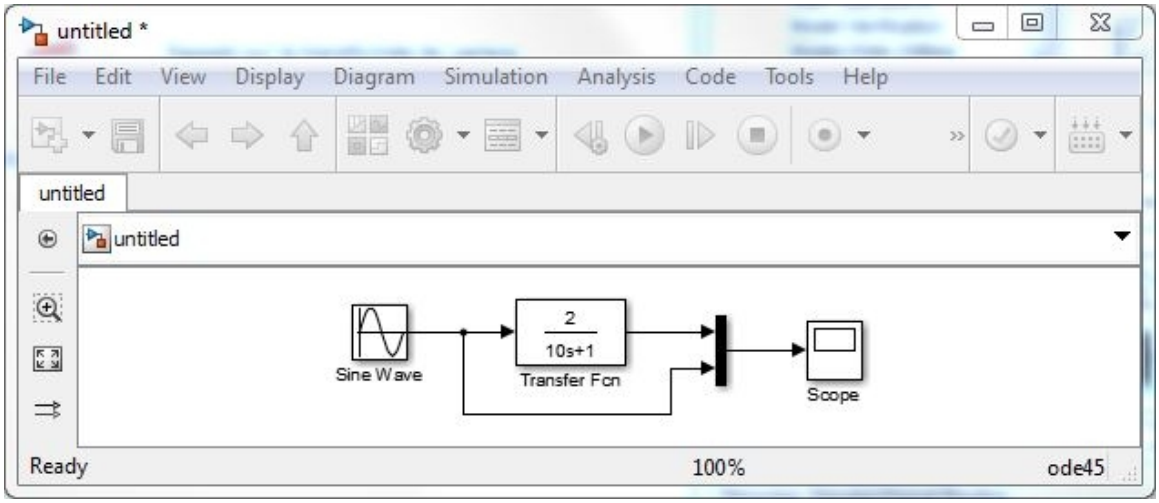
Note 2: Le bloc "Transport Delay" simule un retard.

Exemple 2:



Note: Il s'agit du même système. Le numérateur est entré en utilisant la fonction "conv" qui permet de multiplier les deux polynômes "3s+1" et "2s+1".

Exemple 3:



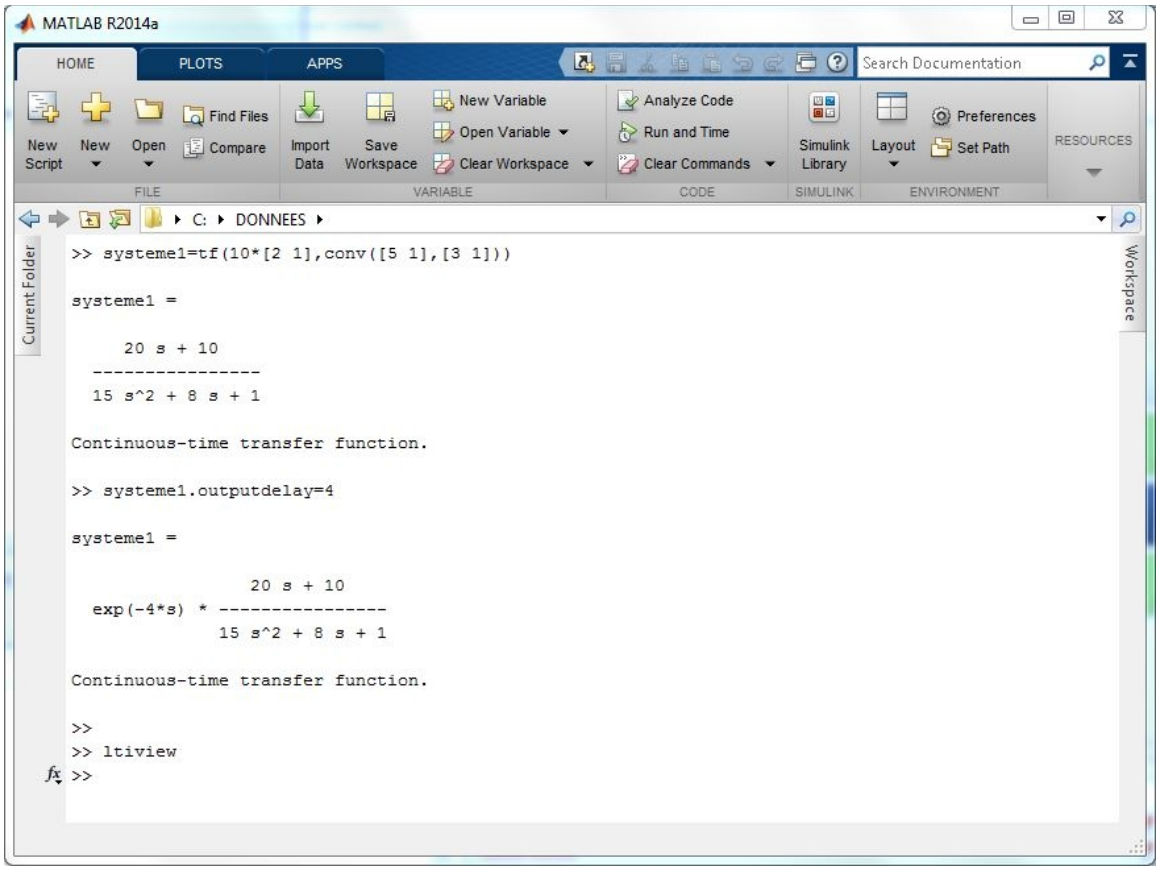
Note 1: Ce système permet de visualiser sur le même graphe l'entrée et la sortie du système et donc de déduire, par exemple, le rapport d'amplitude et la phase à la fréquence d'excitation.

Note2: Le bloc qui réunit l'entrée et la sortie de la fonction de transfert est un "Mux" (multiplexeur).

b) Avec LTI Viewer

LTI Viewer (LTI : linear time invariant) est un environnement qui permet de visualiser le comportement des systèmes linéaires : réponses à l'échelon et à l'impulsion, diagrammes de Bode, lieux de Nyquist et de Black, plan de Laplace, etc.

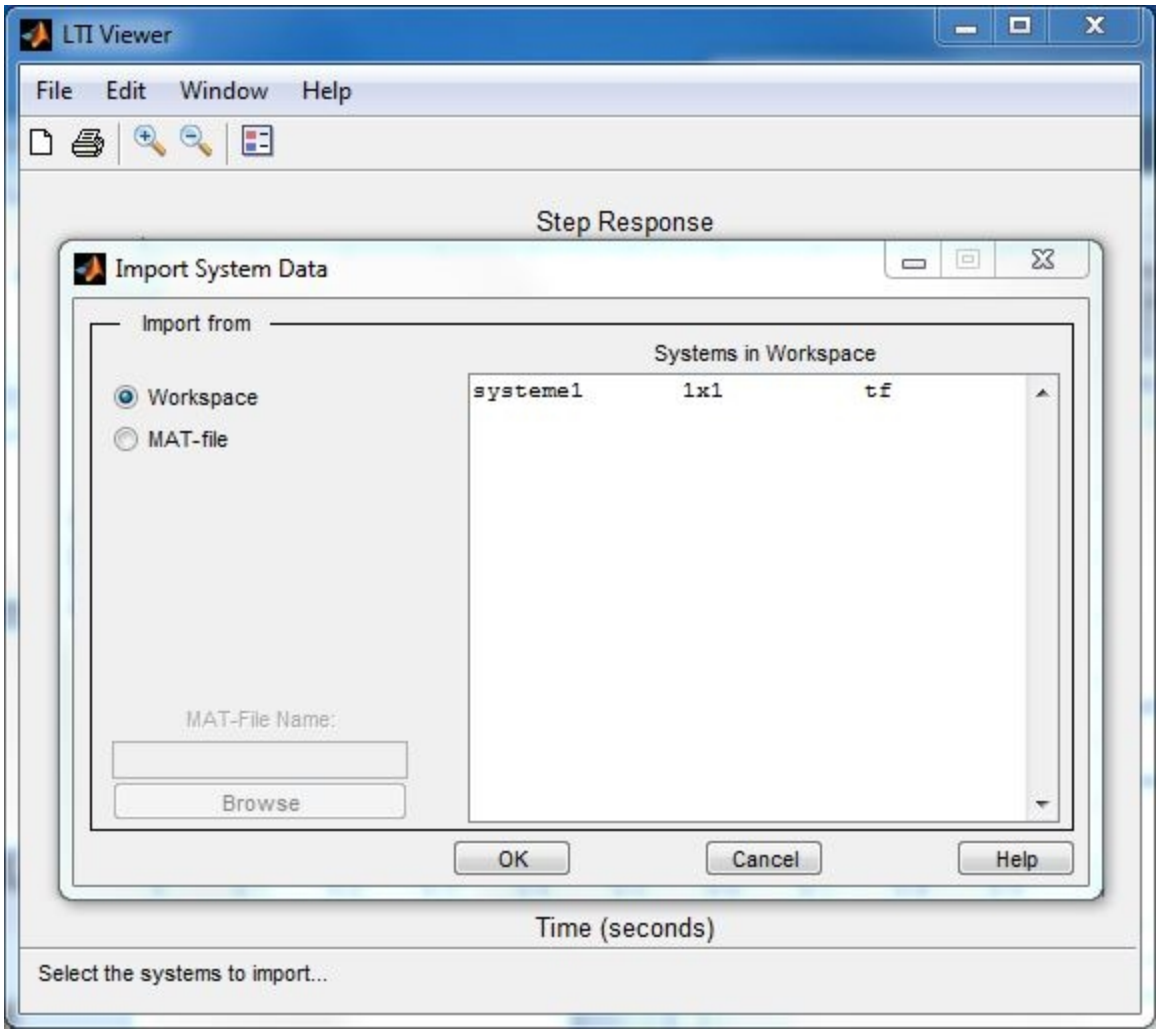
Il faut d'abord définir le système à étudier et démarrer LTI Viewer:



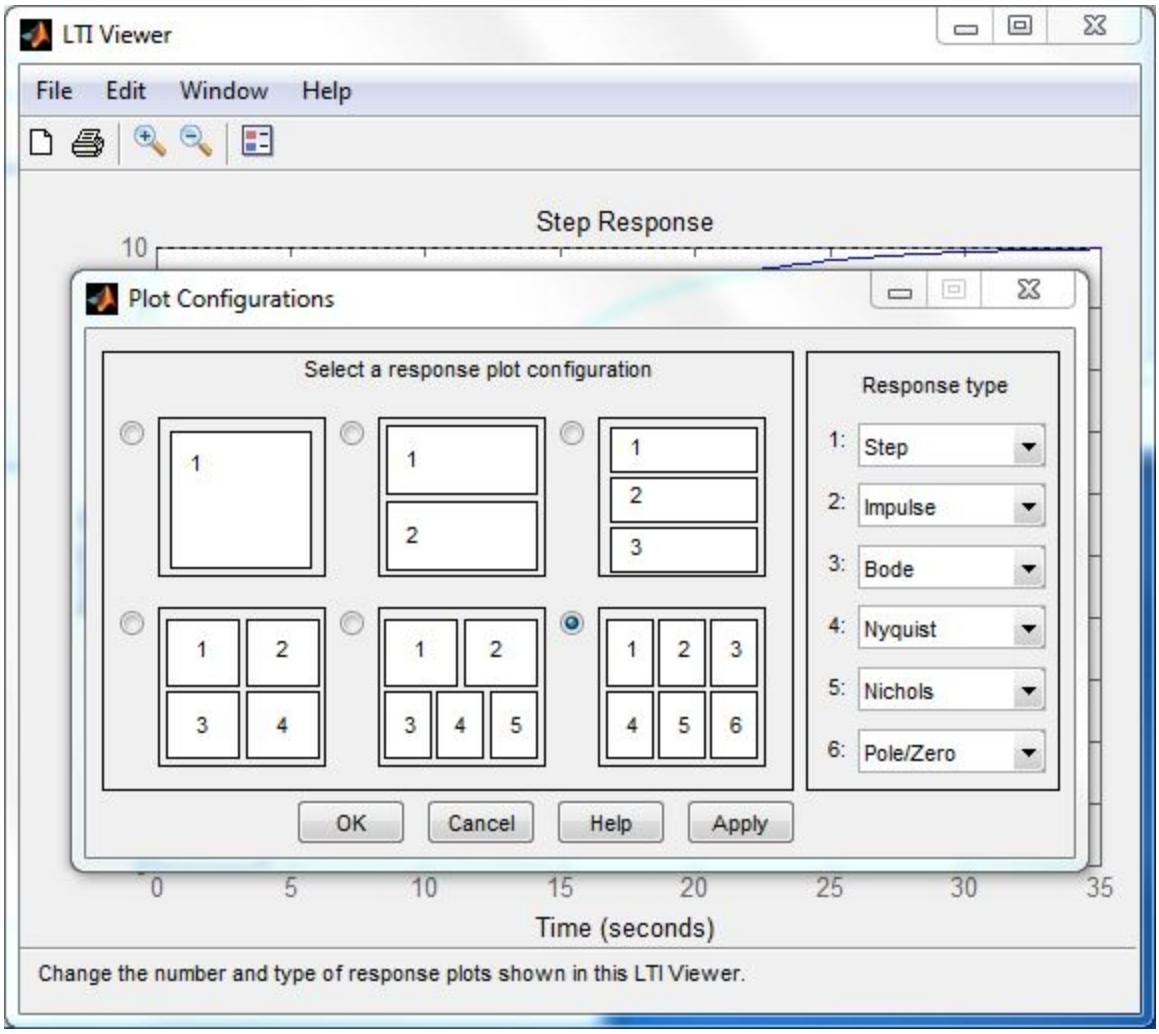
La fonction "tf" définit une fonction de transfert. Les deux arguments à la fonction sont dans l'ordre le numérateur et le dénominateur, tous deux en puissance décroissante de "s".

On peut ajouter un retard par la suite de la façon illustrée dans l'exemple.

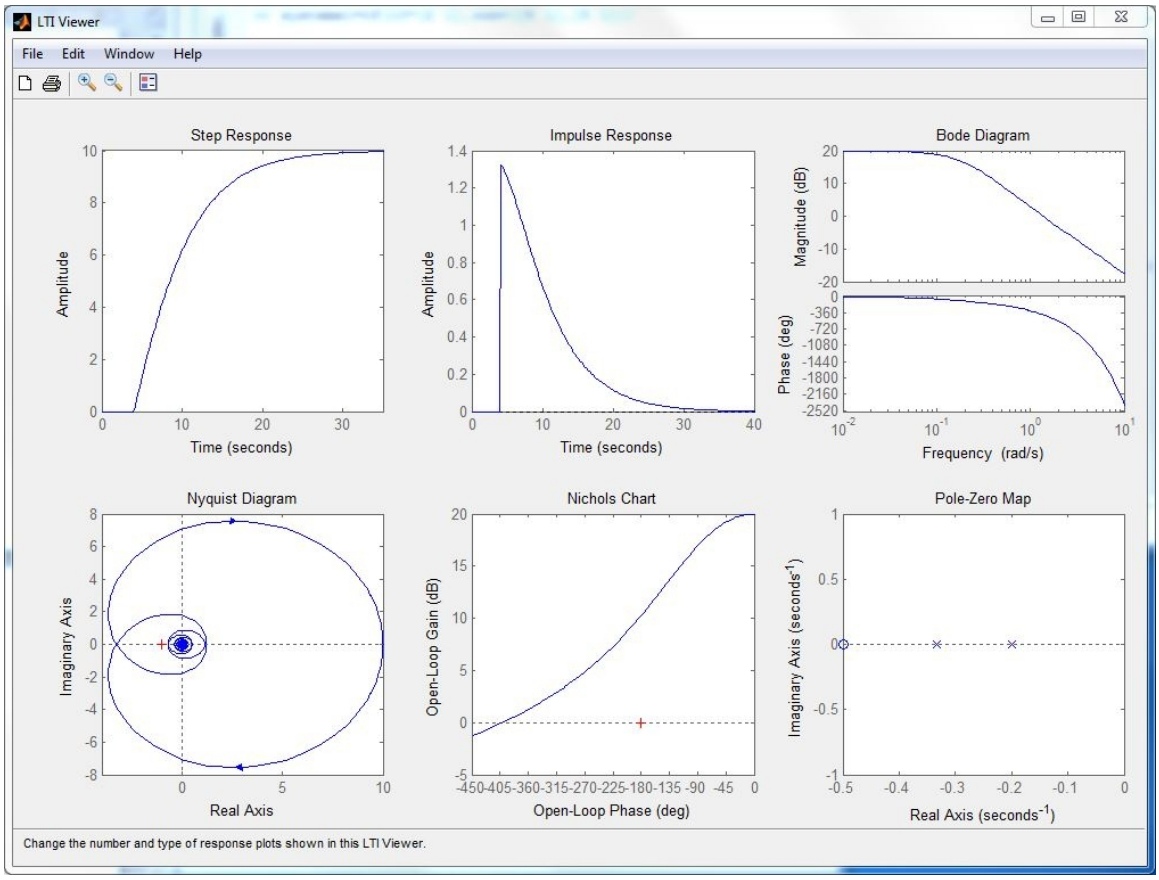
La commande " ltiview" fait apparaître l'interface usager LTI Viewer. En cliquant sur "File" puis "Import", on voit apparaître les systèmes qui sont définis en mémoire. Dans ce cas-ci, seul "systeme1" existe. On le sélectionne puis on pèse sur OK pour que LTI Viewer travaille avec ce modèle.



Pour choisir ce que nous désirons visualiser, il faut cliquer sur "Edit" puis sur "Plot Configurations". Dans ce cas-ci, on indique que nous voulons l'écran séparé en 6 afin d'y voir la réponse à l'échelon ("Step"), à l'impulsion ("Impulse"), le diagramme de Bode ("Bode"), le lieu de Nyquist ("Nyquist"), le lieu de Black ("Nichols") et le plan de Laplace ("Pole/Zero").



En cliquant sur OK on obtient les graphes désirés.



En cliquant sur la courbe avec le bouton gauche de la souris, on peut obtenir la valeur exacte d'un point sur la courbe.

En cliquant sur un graphe (à coté de la courbe) avec le bouton droit, on fait apparaître un menu qui nous offre plusieurs options telles que: faire un zoom, ajouter une grille en arrière-plan, changer quelques propriétés du graphique, etc.

Dans le menu "Edit", l'option "Viewer Preferences" permet entre autres de modifier les temps auxquels les réponses temporelles (échelon, impulsion) et les fréquences auxquelles les réponses en fréquences (Bode, Nyquist, Black) sont calculées.

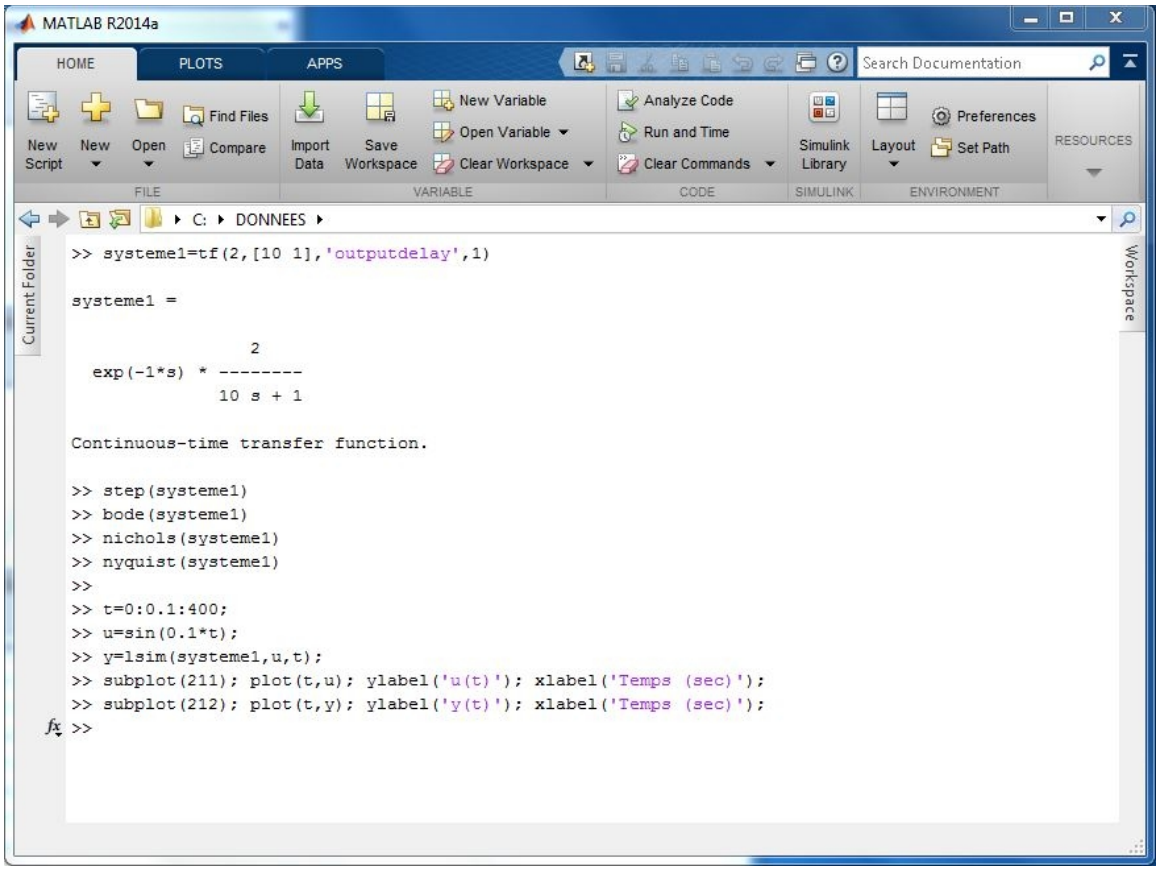
c) Avec des commandes Matlab

Réponses temporelles et fréquentielles

Voici quelques commandes de base qui permettent d'étudier le comportement des systèmes linéaires :

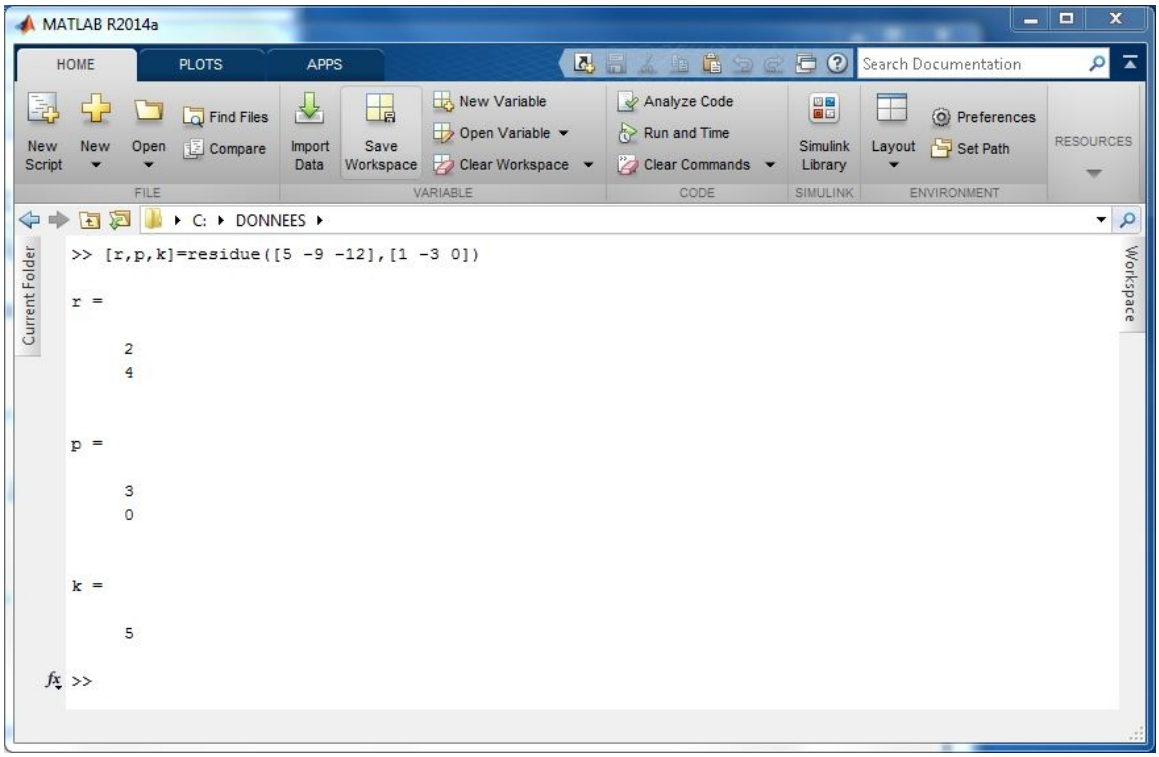
help control	Liste les fonctions du *Control System Toolbox*
tf	Définit un système par sa fonction de transfert
step	Trace la réponse à l'échelon d'un système, ou retourne les valeurs de la réponse temporelle
bode	Trace le diagramme de Bode d'un système, ou retourne les valeurs de la réponse fréquentielle
margin	Comme la fonction précédente mais donne en plus les marges de gain et de phase
nichols	Trace le lieu de Black d'un système, ou retourne les valeurs de la réponse fréquentielle
ngrid	Ajoute l'abaque de Black au lieu de Black
nyquist	Trace le lieu de Nyquist d'un système, ou retourne les valeurs de la réponse fréquentielle
lsim	Calcule la réponse d'un système à une entrée quelconque

Par exemple :


MATLAB R2014a
HOME
PLOTS
APPS
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink Library
Layout
Preferences
Set Path
RESOURCES
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
C: ▸ DONNEES ▸
Current Folder
Workspace
>> systeme1=tf(2,[10 1],'outputdelay',1)
systeme1 =
2
exp(-1*s) * --------
10 s + 1
Continuous-time transfer function.
>> step(systeme1)
>> bode(systeme1)
>> nichols(systeme1)
>> nyquist(systeme1)
>>
>> t=0:0.1:400;
>> u=sin(0.1*t);
>> y=lsim(systeme1,u,t);
>> subplot(211); plot(t,u); ylabel('u(t)'); xlabel('Temps (sec)');
>> subplot(212); plot(t,y); ylabel('y(t)'); xlabel('Temps (sec)');
>>

Décomposition d'une fraction rationnelle en éléments simples

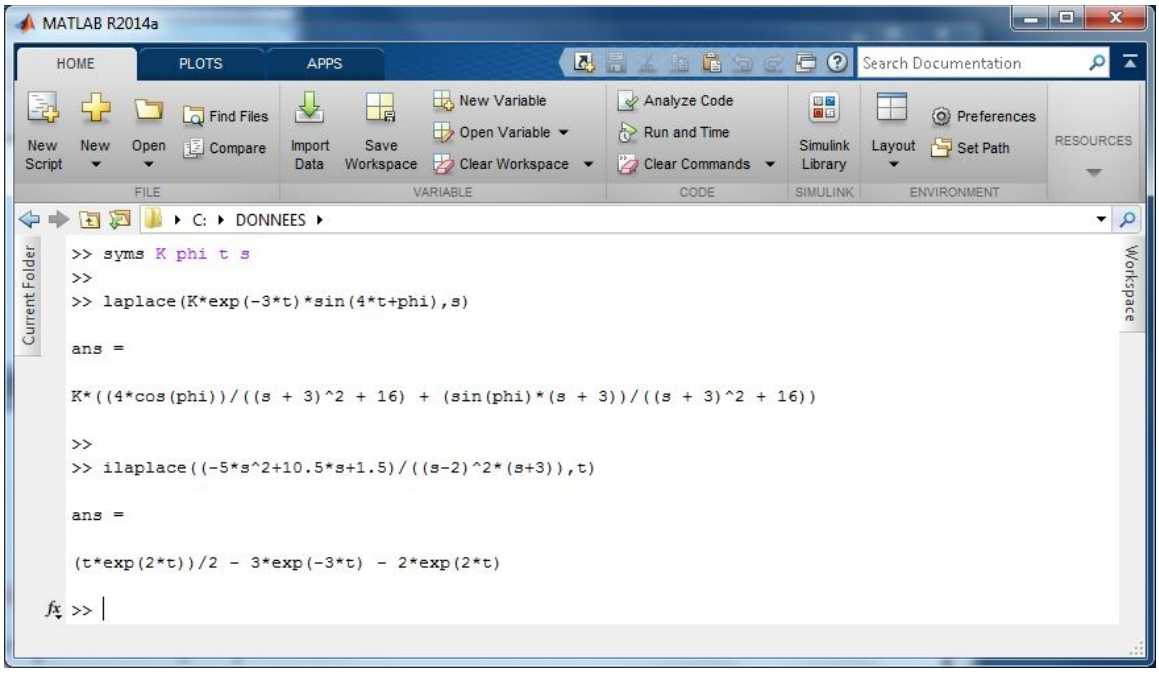
Le calcul qui suit correspond à l'exemple A.2 (équation A.20).



On passe le numérateur et le dénominateur de la fraction rationnelle (en puissance décroissante de "s") à la fonction "residue". On obtient alors les résidus "r", les pôles "p" correspondant aux résidus et la partie entière "k".

Transformation de Laplace

Le calcul symbolique de Matlab permet de calculer les transformées de Laplace et les transformées inverses de Laplace.



La commande "syms" définit les symboles "K", "phi", "t" et "s".

La commande "laplace" calcule la transformée de Laplace de l'expression donnée dans le premier argument et le résultat est fonction du second argument. Le calcul correspond à l'exemple 3.2.

La commande "ilaplace" calcule la transformée inverse de Laplace de l'expression donnée dans le premier argument et le résultat est fonction du second argument. Le calcul correspond à l'exemple 3.4.