
TP n° 01

Mise à niveau pour l'exploitation des boîtes à outils de Matlab Toolbox /Matlab, control et Simulink ...

PARTIE I : COMMANDES AVANCEES :

I.1 REPRESENTAION D'UN POLYNÔME

Dans MATLAB, un polynôme $F(p)$ de degré n est représenté sous la forme codée d'un vecteur ligne F de $n+1$ termes. Ceux-ci sont les coefficients des puissances de p ordonnées par valeurs décroissantes.

I.2 OPERATIONS SUR LES POLYNÔMES :

I.2.1 CALCUL DES RACINES D'UN POLYNÔME

roots() : calcule les racines du polynôme.

I.2.2 DERIVATION DE POLYNÔMES

Polyder() : réalise la dérivation du polynôme

I.2.3 EVALUATION D'UN POLYNÔME POUR UNE VALEUR DONNEE :

Polyval(f,val) : donne la valeur numérique que prend le polynôme lorsqu'on lui applique la valeur numérique val.

I.2.4 MULTIPLICATION DE POLYNÔMES :

z=conv(x,y) : crée le polynôme $z=x \times y$ sous sa forme développée. Le degré du polynôme z est la somme des degrés des polynômes x et y .

I.2.5 LES ZEROS ET LES PÔLES D'UNE FONCTION :

zero(sys) : donne les racines du numérateur.

pole (sys) : donne les racines du dénominateur.

I.2.6 LA TRANSFORMEE DE LAPLACE:

Matlab permet de calculer les transformées de Laplace et les transformées inverses de Laplace.

Syms : définit les symboles "t" et "s"

laplace : calcule la transformée de Laplace de l'expression donnée.

I.2.7 LA TRANSFORMEE INVERSE DE LAPLACE

ilaplace : calcule la transformée inverse de Laplace de l'expression donnée.

PARTIE II : CONTROL SYSTEM TOOLBOX ET SIMULINK

A. CONTROL SYSTEM TOOLBOX

II.1 DEFINITION D'UN SYSTEME LINEAIRE:

II.1.1 DEFINITION D'UN SYSTEME LINEAIRE PAR SA FONCTION DE TRANSFERT:

Sys=tf (num, den)

num : c'est le polynôme numérateur.

den : c'est le polynôme dénominateur.

tf : définit une fonction de transfert

II.1.2 DEFINITION D'UN SYSTEME LINEAIRE PAR SES POLES ET ZEROS:

Sys= zpk (zer,pol,gain)

zer: vecteur ligne qui donne la liste des zéros. (Les zéros sont les racines du numérateur)

pol: vecteur ligne qui donne la liste des pôles. (Les pôles sont les racines du dénominateur)

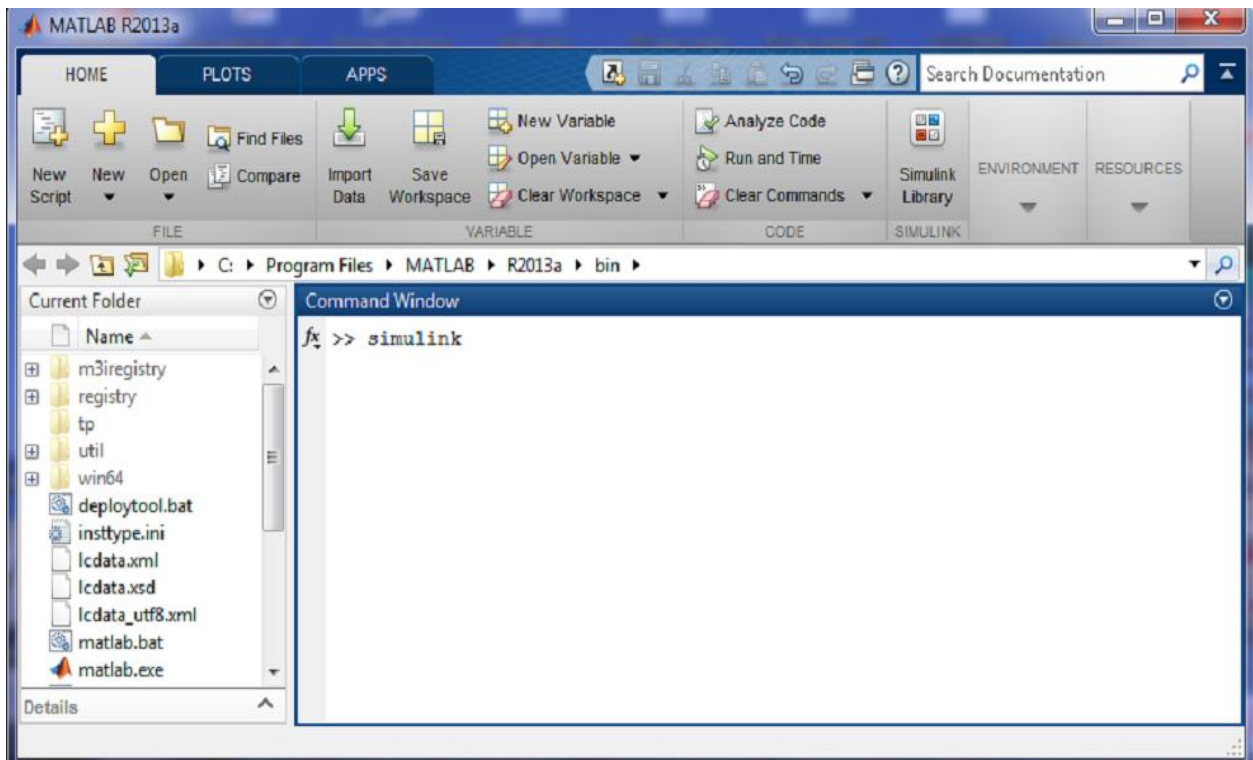
gain: est un scalaire, c'est le gain global que l'on peut mettre en facteur de l'ensemble.

II.1.3 DEFINITION D'UN SYSTEME LINEAIRE EN INTRODUISANT SA FONCTION DE TRANSFERT DIRECTEMENT

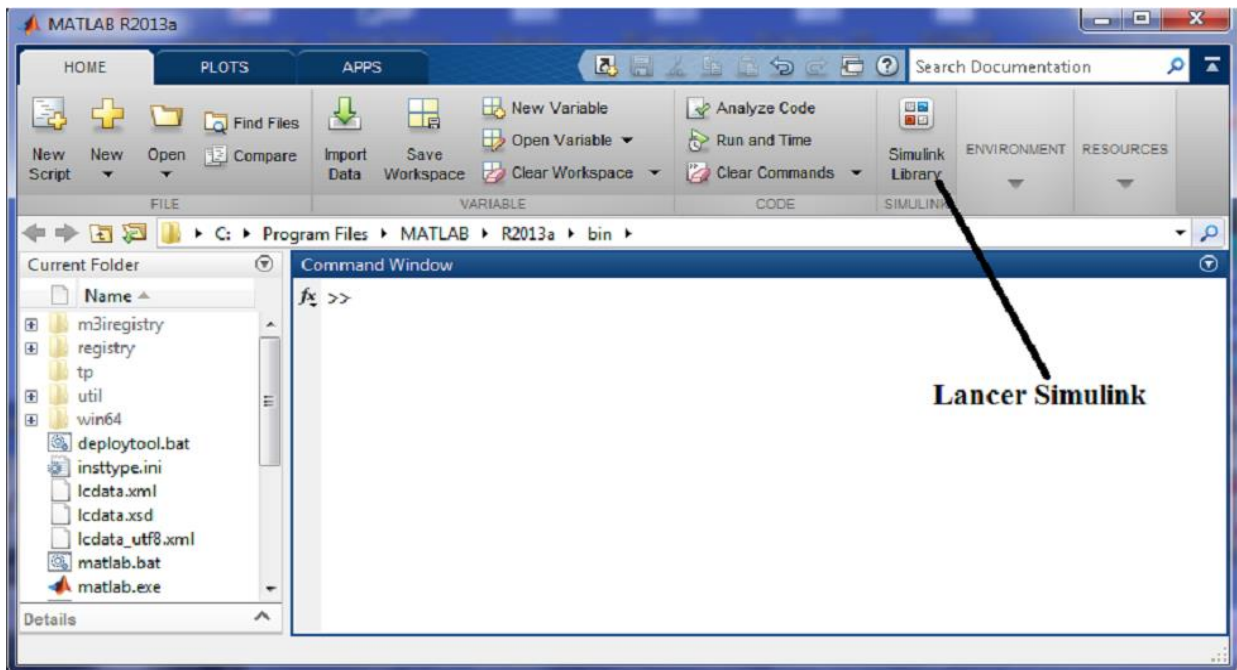
B. SIMULINK :

II.1 PRESENTATION

Simulink est une interface graphique qui facilite l'analyse des systèmes dans le domaine temporel. Les systèmes ne sont pas décrits par des lignes de code Matlab mais simplement définis par des blocs dont tous les éléments sont prédéfinis dans des bibliothèques qu'il suffit d'assembler. Lorsque le schéma bloc du système que l'on étudie est représenté sous Simulink, il est possible d'analyser sa réponse temporelle pour des entrées diverses (échelon, rampe.....). Pour lancer Simulink, on procède à partir de la fenêtre de commande de Matlab : *Soit on tape à la suite du prompt Matlab (>>) tout simplement la commande Simulink.

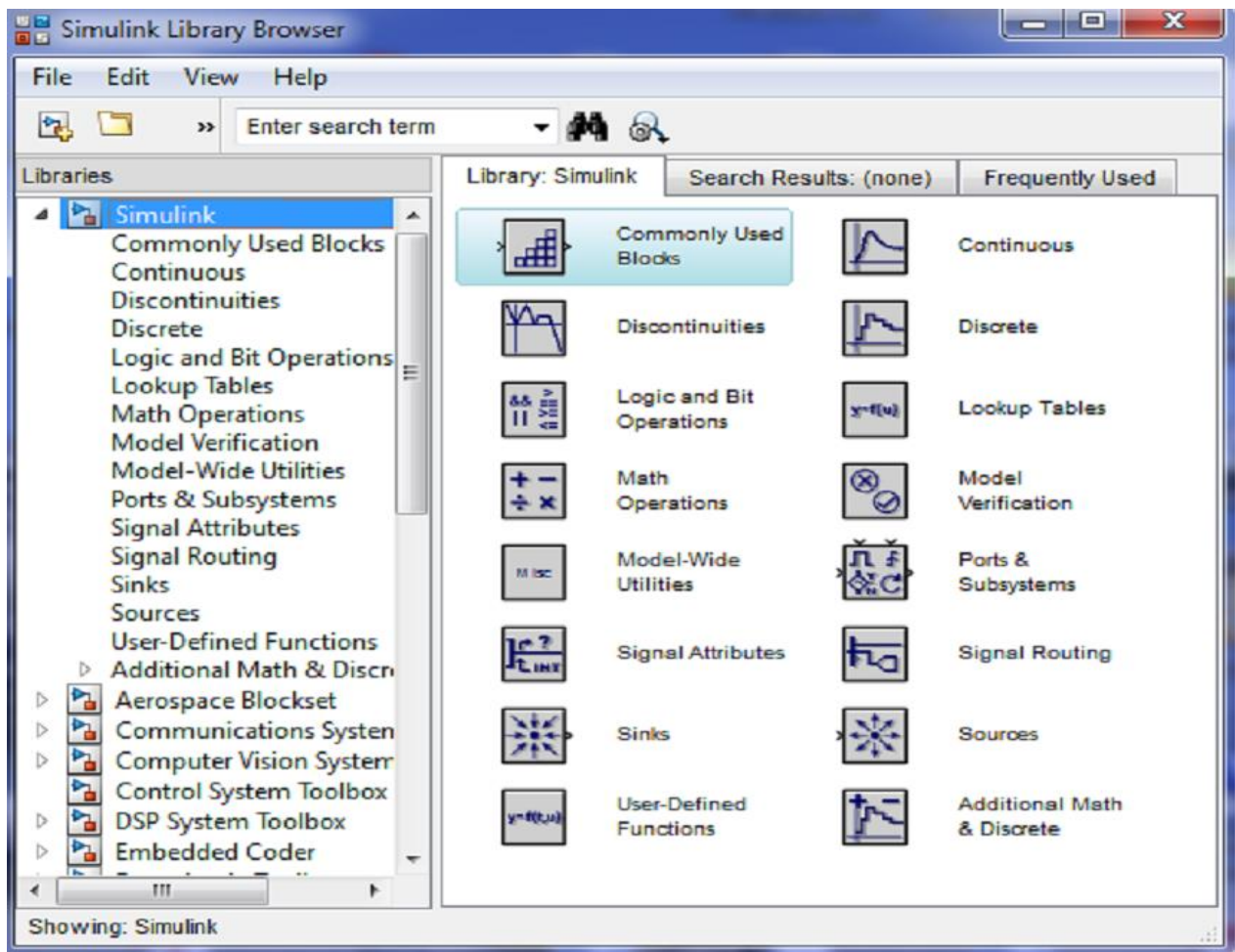


*Soit depuis la barre d'outils de Matlab, on clique sur l'icône dédiée à Matlab

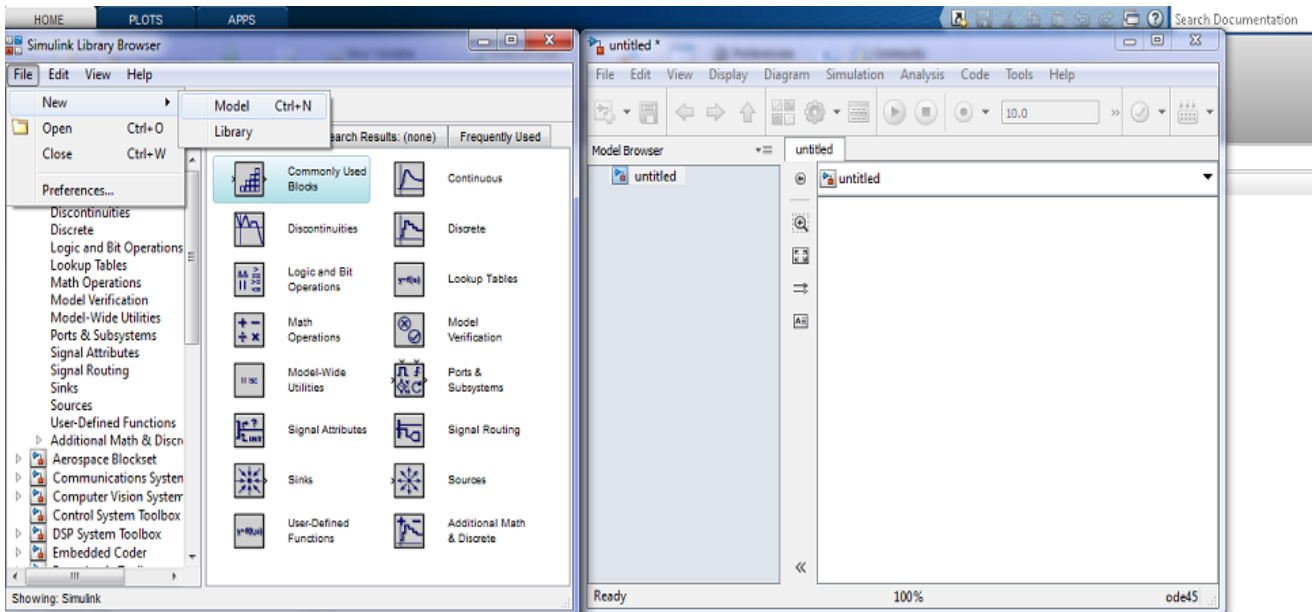


Ces deux actions ouvrent la fenêtre Simulink Library Browser qui permet l'accès à la Bibliothèque Simulink ainsi qu'à d'autres bibliothèques (control system, par exemple). Simulink, comme chaque bibliothèque importante, comporte des compartiments : Continuous, Discrete, Source.... Qui contiennent les blocs.

II.2 QUELQUES BIBLIOTHEQUES :



En cliquant sur File New Model, on ouvre une fenêtre de travail Untitled (sans titre) pour composer le nouveau schéma.



Pour copier un bloc d'une bibliothèque Simulink dans la fenêtre de travail, on le sélectionne en cliquant dessus avec le bouton gauche de la souris. Tout en maintenant le bouton gauche enfoncé, on se déplace avec la souris jusqu'à la fenêtre de travail Simulink. On relâche le bouton de la souris à l'endroit où l'on souhaite positionner son bloc. Le bloc se retrouve ainsi dupliqué dans la fenêtre de travail.

Traçage de courbes : On utilise l'instruction **plot** pour tracer un graphique 2D.

plot(x,y) : permet de tracer y en fonction de x : $y=f(x)$ avec x et y deux vecteurs de même longueur. On désire par exemple représenter la fonction $f(t)=t$, pour $0 \leq t \leq 6\pi$ avec un pas de $\pi/10$.

Introduire alors la commande suivante :

`t=[0 :pi/10: 6*pi] ;`

`f = t ;`

`plot(t , f)`

On peut ajouter à plot un troisième paramètre pour indiquer la couleur et le type de tracé (continu ou discontinu) de la courbe (Voir tableau 1).

	Couleur	Marqueur		Style du tracé	
y	jaune	.	point	- (*)	ligne continue
m	magenta	o	cercle	:	ligne pointillée
c	cyan	x	x	- -	tirets
r	red	+	plus	-.	tiret point
g	vert	*	étoiles		
b	bleu	s	carrés		
w	blanc	d	losanges		
k	noir (*)	<, >	triangles		

REMARQUE 1 : Les valeurs notées (*) sont les valeurs par défaut. On peut ajouter à la courbe obtenue les identifiants suivants :

- Le titre de la figure : `title(' ');`
- Le titre de l'axe des abscisses : `xlabel('x');`
- Le titre de l'axe des ordonnées : `ylabel('y');`

Un quadrillage (grille), réalisé par la commande **grid**, donne plus de lisibilité au graphique.

La commande **grid off** supprime le quadrillage du graphique courant.

REMARQUE 2 :

- On peut faire appel à l'éditeur MATLAB en suivant le chemin suivant : **File/New/M-file.**
- On peut également accéder à un fichier déjà enregistré pour le modifier, en suivant : **File/Open/Nom du fichier.**

Exercice 1 :

Ecrire l'équation suivante sous Matlab :

$$y(t) = 5t^4 + 5t^2 + 8t - 1 ; \quad t : \text{étant le temps}$$

Calculer les racines de $y(t)$.

Calculer $y(-1)$, $y(1)$, $y(2)$.

Calculer $dy(t)/dt$

Calculer $z(t) = y(t).x(t)$ avec $x(t) = 9t^2 + t + 5$

Exercice 3 :

Ecrire un programme sous Matlab pour trouver les transformées de Laplace des fonctions suivantes: e^{-at} , t^7 , $\sin wt$, $\cos wt$.

Exercice 4 : Ecrire un programme sous Matlab, pour trouver les transformées inverses de :

$$y1(p) = \frac{3p+1}{p^2+8p+3}, \quad y2(p) = \frac{5}{p(p^2+w^2)}, \quad y3(p) = e^{-\pi p} \quad (w \text{ constante.})$$

Exercice 5 : En utilisant Simulink construisez un modèle pour avoir la réponse à un échelon en boucle ouverte et en boucle fermée d'un système du premier ordre : $k/1+\tau p$, ayant un gain $k=5$ et une constante de temps $\tau=30s$.

Exercice 6 :

Soit un circuit RC attaqué par un échelon d'amplitude 24V,

avec $R=50\Omega$, $C=100\mu F$.

Exercice 2 :

Ecrire sous Matlab, les fonctions de transfert suivantes selon 2 méthodes :

$$F1(p) = \frac{1}{p+6};$$

$$F2(p) = \frac{p+2}{p(p-5)};$$

$$F3(p) = 7 \frac{(p+9)}{p^2+2p+5};$$

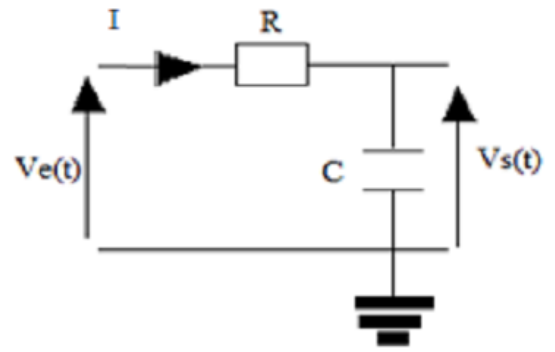
$$F4(p) = 5 \frac{(p-7)}{p^2+0,99p+0,4}$$

Le condensateur est initialement déchargé.

1-Etablissez le modèle simulink de ce montage.

2- Visualisez les courbes en fonction du temps ,de la tension et du courant obtenus au niveau du condensateur.

3-Etablissez l'étude théorique et interprétez les résultats obtenus.



Exercice 7 :

Soit le circuit suivant, on donne : $E=24V$, $R=1.5 K\Omega$, $C=10nF$, $L=10^{-4}H$.

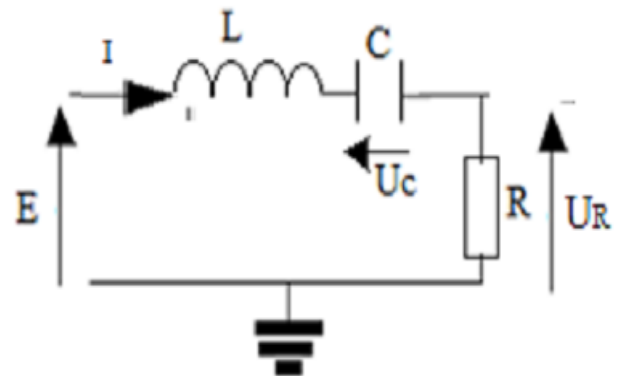
Le condensateur est initialement déchargé.

1-Etablissez le modèle Simulink du circuit.

2-Visualisez les tensions E, UC, UR selon le temps.

3-Faire l'étude théorique et établissez

les diverses équations.



TP n° 02

Etude des systems dans L'espace d'état sous Matlab et Simulink ...

I.1. Introduction

Pendant longtemps, l'automaticien a utilisé exclusivement la notion de relation entrée-sortie. Or, pour les systèmes complexes, ce point de vue se révèle être une représentation incomplète des phénomènes.

La connaissance des variables internes d'un système peut présenter un intérêt certain en ce sens qu'elles permettent de déterminer l'état d'un système à un instant donné.

Ces variables d'état sont donc l'ensemble des variables nécessaires à la caractérisation de l'état d'un système. Elles sont regroupées dans un vecteur appelé vecteur d'état.

I.2. Rappels

Les équations d'état d'un système sont constituées de l'ensemble des équations matricielles suivantes :

$$\begin{cases} \dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t) & \text{équation d'état} \\ \underline{y}(t) = C\underline{x}(t) + D\underline{u}(t) & \text{équation d'observation} \end{cases}$$

Avec :

\underline{x} : vecteur des variables d'état

\underline{u} : vecteur des commandes

\underline{y} : vecteur des sorties

A : matrice d'évolution de dimension $(n \times n)$ (n : ordre du système)

B : matrice de commande de dimension $(n \times l)$ (l : nombre d'entrées du système)

C : matrice d'observation de dimension $(m \times n)$ (m : nombre de sorties)

D : matrice de transmission directe de dimension $(m \times l)$

Ces équations sont établies pour les systèmes multivariables, dans le cas particulier des systèmes monovariante qui nous intéresse, on peut réécrire ces équations de la façon suivante :

$$\begin{cases} \dot{\underline{x}}(t) = A\underline{x}(t) + Bu(t) \\ \underline{y}(t) = C\underline{x}(t) + Du(t) \end{cases}$$

Dans la plupart des systèmes rencontrés en automatique, il n'existe pas de lien direct entre l'entrée et la sortie du système. Par conséquent, dans la représentation d'état de tels systèmes, la matrice D de transmission directe est nulle.

Le choix des variables d'état conduit à des formes particulières pour les matrices, ces structures spécifiques sont appelées forme canonique, qu'on peut classer en trois formes :

Forme canonique diagonale

La matrice d'état est diagonale et les éléments de la diagonale sont les valeurs propres.

Forme canonique commandable

Une ligne de la matrice d'état correspond aux coefficients du polynôme caractéristique de la fonction de transfert

Forme canonique observable

Une colonne de la matrice d'état correspond aux coefficients du polynôme caractéristique de la fonction de transfert

I.3. Présentation du logiciel

Matlab (MATrix LABoratory) est l'outil de référence pour la simulation numérique, notamment en ce qui concerne l'Automatique.

Matlab est une application Windows. En cliquant deux fois sur l'icône **Matlab**, s'ouvre alors la fenêtre principale. Cette fenêtre est divisée en plusieurs parties, comme le montre la Figure I.1 ci-dessous.

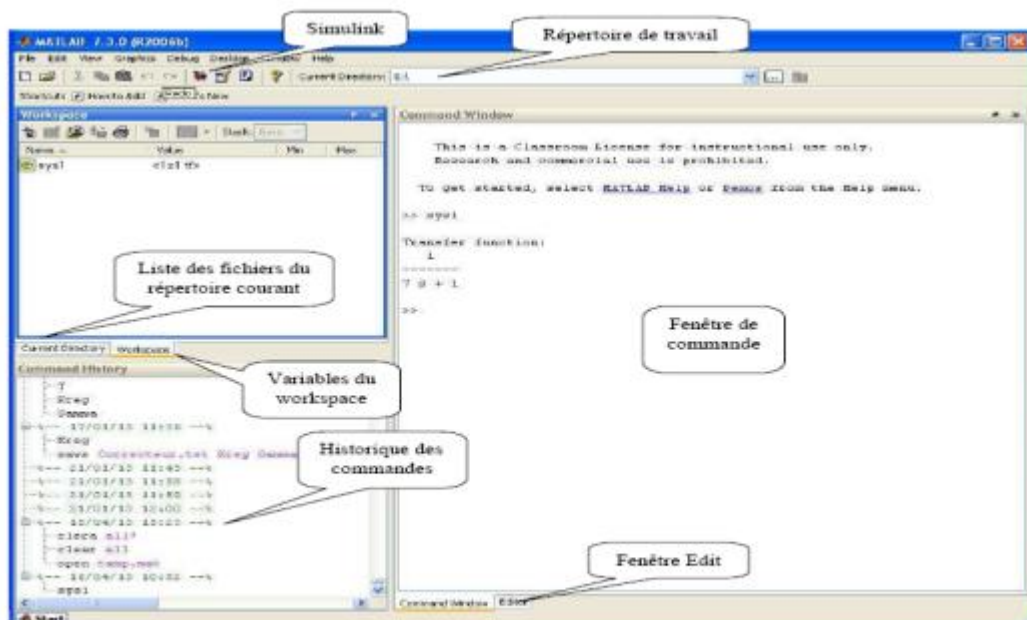



Figure I.1 : IHM (Interface Homme Machine) de **Matlab**

Programmation sous Matlab :

- Lancer Matlab sous windows, une fenêtre (command window) est alors affichée.
- Cliquer sur File et choisir “New” ensuite “Mfile”, une nouvelle fenêtre est ouverte dans laquelle on peut écrire notre programme.
- Donner un nom au programme à enregistrer.
- Pour l’exécution du programme, taper son nom dans la fenêtre (Command window) ou cliquer sur l’icône **run**  dans la barre d’outils de Matlab.


Une boîte à outils de Matlab correspond à un ensemble de fonctions disponibles dans la fenêtre de travail que l’on peut appeler à l’invite >>

De même, pour chaque commande particulière, vous avez accès à deux niveaux d’aide :

1. Un 1^{er} niveau d’aide par : >> help ‘nom de la commande’
2. Un 2^{ème} niveau d’aide, avec navigation hypertexte par : >>helpwin ‘nom de la commande’ .

Exemple: >>help ss ou >>helpwin ss

Matlab possède une boîte à outils réservée à l’étude des systèmes de commande « control system Toolbox », ainsi qu’un environnement graphique de simulation des systèmes dynamiques **Simulink** sous forme de schémas en blocs.

Pour lancer Simulink, on clique sur l’icône  dans la barre d’outils de Matlab, ou bien par la commande >> simulink à l’invite, dans la fenêtre de travail de Matlab.

Ceci nous donnera accès à différents menus ou boîtes de Simulink ; ainsi on ouvrira une nouvelle fenêtre Simulink à partir de laquelle on pourra créer un nouveau fichier modèle et placer les éléments constitutifs du système à étudier.

La librairie de Simulink comprend de différentes sections permettant d’aborder de nombreux aspects de la commande des systèmes.

Dans notre polycopié on portera notre attention sur les éléments suivants comme le montre la Figure I.2 :

- **Step** : se trouve dans **Sources**, générateur de l'échelon.
- **Transfert fcn** : se trouve dans **Continuous**, permet de définir une fonction de transfert. Changez les paramètres pour qu'elle corresponde à la fonction de transfert demandée.
- **State-space** : se trouve dans **Continuous**, permet de définir une représentation d'état. changer les matrices A, B, C, D pour qu'elle corresponde à la représentation d'état demandée.
- **Gain** : se trouve dans **Math Operations**, permet de multiplier l'entrée par un gain k (scalaire ou vecteur)
- **Mux** : se trouve dans **Signal routing**, permet de multiplexer plusieurs signaux dans un fil.
- **Sum** : se trouve dans **Math Operations**, permet de réaliser le comparateur. Il faut choisir les signes $+$ et $-$.
- **Scope** : se trouve dans **Sinks**, c'est un scope rudimentaire pour avoir rapidement un tracé des courbes.
- **ToWorkspace**: il se trouve dans **Sinks**, permet de récupérer le résultat de la simulation dans une variable exploitable sur Matlab (ligne de commande).

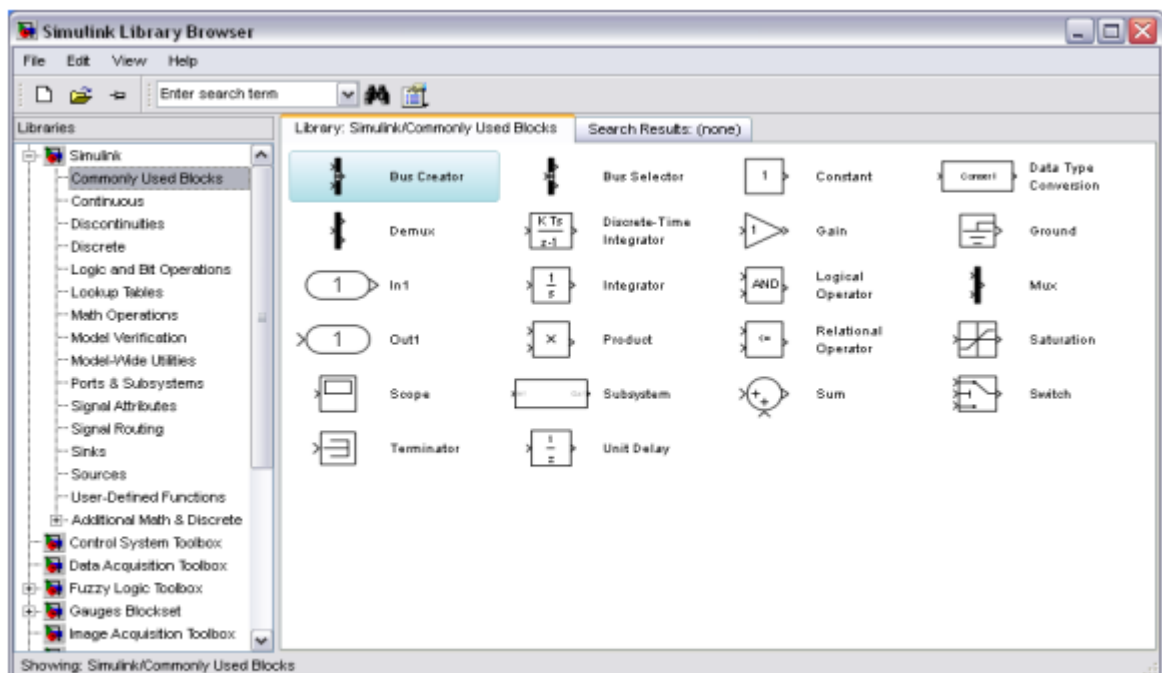


Figure I.2 : interface Simulink

I.4. Descriptions des systèmes

Dans toute la suite de ce TP, les systèmes considérés n'auront qu'une entrée et qu'une sortie.

I.4.1. Exemple 1 : système électrique

Considérons un circuit électrique alimenté par une tension $u(t)$, constitué des éléments suivants : résistance R , inductance L et capacité C en série (Figure I.3), en notant q la charge aux bornes du condensateur et i l'intensité du courant qui traverse le circuit.

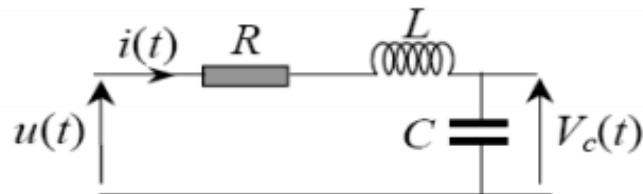


Figure I.3: Schéma d'un circuit RLC

La loi des mailles conduit à :

$$u(t) = Ri(t) + L \frac{di(t)}{dt} + V_c(t)$$

La formule suivante relie l'intensité du circuit à la charge et donc à la tension aux bornes du condensateur :

$$i(t) = C \frac{dV_c(t)}{dt} = \frac{dq(t)}{dt}$$

On arrive aisément à l'équation différentielle temporelle comportant une dérivée d'ordre 2

$$LC \cdot \frac{d^2V_c(t)}{dt^2} + RC \cdot \frac{dV_c(t)}{dt} + V_c(t) = u(t)$$

Données numériques du système : $R = 400\Omega$, $C = 100\mu F$, $L = 10H$

Pour la modélisation dans l'espace d'état du système, on précise que les variables $i(t)$ et $q(t)$ sont des variables d'états importantes et donc à prendre en compte.

D'où le choix suivant pour le vecteur d'état :

$$\text{Le vecteur d'état : } \begin{cases} x_1(t) = i(t) \\ x_2(t) = q(t) \\ x_3(t) = V_c(t) \end{cases}, \text{ l'entrée est } u(t), \text{ la sortie } y(t) = x_3(t)$$

I.4.2. Exemple 2 : Système mécanique

Une masse m , retenue par un système ressort +amortisseur, se déplace sur un plan sous l'effet d'une force extérieure f (Figure I.4)

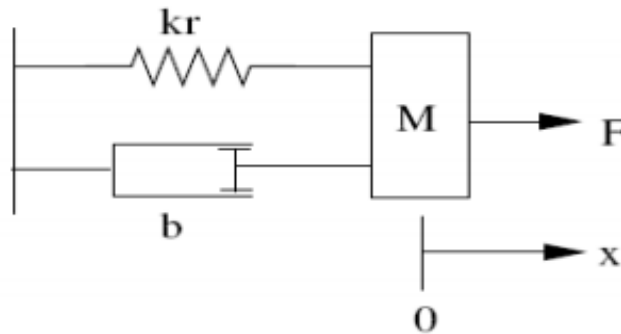


Figure I.4 : Schéma d'une masse en translation

La loi de Newton permet d'écrire l'équation du système :

$$M \cdot \ddot{x}(t) + k_r \cdot x(t) + b \cdot \dot{x}(t) = F(t) , \text{ équation du second ordre.}$$

$$\text{Données numériques du système : } k_r = 2, b = 8, M = 3$$

On considérera comme état, le vecteur d'état suivant : $\underline{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}$

L'entrée est la force $F(t)$ et la sortie est la position $x(t)$

I.4.3. Exemple 3 : Système électromécanique "moteur à courant continu"

Un moteur à courant continu (MCC), (Figure I.5), est un dispositif électromécanique qui convertit une énergie électrique d'entrée en énergie mécanique.

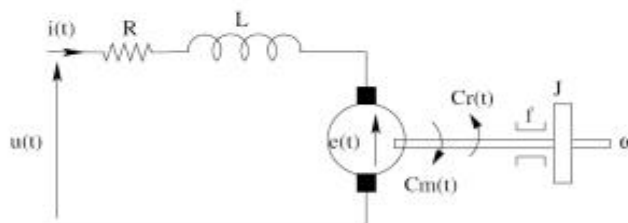


Figure I.5 : Schéma du moteur à courant continu

Pour la partie électrique, on calcule la tension aux bornes de l'induit.

L'équation électrique, liant la tension u aux bornes de l'induit et le courant d'induit i s'écrit :

$$u(t) = R \cdot i(t) + L \frac{di(t)}{dt} + e(t)$$

Où R est la résistance de l'induit du moteur, L son inductance et e la force électromotrice, qui est proportionnelle à la vitesse de rotation du rotor :

$$e(t) = k \cdot w(t)$$

Pour la partie mécanique, on applique le principe fondamental de la dynamique autour de l'axe de rotation. L'équation mécanique rendant compte des couples agissant sur le rotor s'écrit :

$$C_m(t) = J \frac{dw}{dt} + fw(t)$$

Où C_m est le couple moteur, f le coefficient de frottement visqueux et J le moment d'inertie du rotor. Par construction, le couple moteur est proportionnel au courant d'induit i :

$$C_m(t) = k \cdot i(t)$$

On arrive aisément à l'équation différentielle temporelle comportant une dérivée d'ordre deux

$$u(t) = \frac{J \cdot L}{k} \cdot \frac{d^2w(t)}{dt^2} + \frac{RJ + Lf}{k} \cdot \frac{dw(t)}{dt} + \frac{Rf}{k} \cdot w(t)$$

Données numériques du moteur à courant continu voir le tableau I.1

Coefficient de frottement visqueux	$f = 3 \cdot 10^{-4} \text{N/m/rad/s}$
Moment d'inertie	$J = 1.1 \cdot 10^{-3} \text{kg.m}^2$
Résistance de l'induit	$R = 4.8\Omega$
Inductance de l'induit	$L = 18.4 \text{mH}$
Coefficient de fcém (c'est aussi constant du moteur en N.m/A)	$k = 0.45 \text{ V/rad/s}$

Tableau I.1: Données numériques du moteur à Courant Continu

On considérera comme état :

- Courant d'induit $x_1(t) = i(t)$,
- Vitesse angulaire $x_2(t) = w(t)$,
- Entrée : tension d'alimentation $u(t)$,
- Sortie : position angulaire $y(t) = \theta(t)$.

I.5. Etude et Analyse de la représentation d'état des systèmes

❖ Préparation : (à remettre au début du TP)

- ✓ Donner la représentation d'état de chacun des trois systèmes présentés ci-dessus,
- ✓ Parmi les éléments $A, B, C, D, \underline{x}, \underline{u}, \underline{y}$, quels sont ceux qui varient en fonction de la représentation d'état choisie? Pourquoi les autres éléments restent-ils identiques quel que soit la représentation d'état?
- ✓ La représentation d'état d'un système est-elle unique ?
- ✓ Ecrire les équations générales permettant, de passer de la représentation d'état d'un système à la représentation externe (Fonction de transfert), en utilisant la transformation de Laplace,
- ✓ En prenant la transformée de Laplace de l'équation différentielle du système, donner sa fonction de transfert qui relie l'entrée avec la sortie. Quel est l'ordre du système ?

❖ Répondre à toutes les questions suivantes pour chaque système

- ✓ Créer un fichier de type M-file et en utilisant la fonction Matlab **ss**, construire la représentation d'état continue du système,
- ✓ Indiquer la dimension de chacune des matrices A, B, C , et D , en utilisant la fonction Matlab **size**,
- ✓ Calculer la fonction de transfert $H(p) = C \cdot [pI - A]^{-1} \cdot B$ en utilisant les fonctions Matlab **syms, inv, eye**,
- ✓ Trouver la représentation d'état sous la forme canonique diagonale en utilisant la fonction Matlab **canon** avec l'option 'modal',
- ✓ Trouver la représentation d'état sous la forme canonique observable en utilisant la fonction Matlab **canon** avec l'option 'companion',
- ✓ Comment vérifier si deux modèles d'état sont équivalents du point de vue du comportement entrée-sortie ?
- ✓ En utilisant la fonction Matlab **ss2tf**, passer de la représentation d'état (représentation interne) du système à sa fonction de transfert (représentation externe),
- ✓ En utilisant la fonction Matlab **tf2ss**, passer de la fonction de transfert (représentation externe) du système à sa représentation d'état (représentation interne),
- ✓ Analyser la position des pôles et des zéros du système, et déterminer s'il est stable en boucle ouverte ou pas, tout en justifiant. Pour cela utiliser la fonction Matlab **pzmap**,
- ✓ Calculer les valeurs propres de la matrice d'état en utilisant la fonction Matlab **eig**,
Faire le point en répondant à cette question : Que peut-on dire sur les valeurs propres de A et les pôles du système ?
- ✓ Calculer le gain statique du système en utilisant la fonction Matlab **dcgain**,

TP n° 03

Commande par retour d'état (Par placement des poles) Simulation dans l'environnement logiciel Matlab/Simulink ...
