



# Systèmes asservis

## TP 1 : Initiation à Matlab, polynômes et graphismes

### Version enseignant avec solution

## Objectif du TP

Le but de ce TP est de maîtriser quelques outils de base du logiciel Matlab nécessaires pour la suite du module de systèmes asservis.

Les objectifs principaux sont :

- manipuler des polynômes sous Matlab ;
- calculer des racines, des dérivées, des produits et des divisions de polynômes ;
- évaluer un polynôme en un point ;
- tracer des courbes avec la commande `plot` ;
- utiliser les commandes graphiques : `figure`, `subplot`, `title`, `xlabel`, `ylabel`, `legend`, `grid`.

## Aperçu du chapitre suivant : Simulation sous Matlab

Dans le chapitre suivant, on utilisera Matlab pour simuler des systèmes dynamiques et des systèmes asservis.

On abordera principalement :

- la résolution d'équations différentielles ;
- l'utilisation des commandes `ode45`, `ode23`, `dsolve`, `diff`, `int` ;
- la détermination de la fonction de transfert d'un système ;
- le tracé des réponses temporelles : `step`, `impulse`, `lsim` ;
- le tracé des réponses fréquentielles : `bode`, `nyquist` ;
- l'étude des systèmes en boucle ouverte et en boucle fermée ;
- la stabilité à partir des pôles du système ;
- l'identification graphique de systèmes à partir de leurs réponses temporelles.

## Idée générale

Un système dynamique peut souvent être décrit par une équation différentielle, par exemple :

$$\frac{dy(t)}{dt} + 2y(t) = u(t)$$

où :

- $u(t)$  est l'entrée du système ;
- $y(t)$  est la sortie ;
- $t$  est le temps.

Sous Matlab, cette équation peut être résolue de deux manières :

- analytiquement avec `dsolve` ;
- numériquement avec `ode45` ou `ode23`.

On pourra aussi écrire un système sous forme de fonction de transfert :

$$G(p) = \frac{Y(p)}{U(p)}$$

Par exemple :

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

Sous Matlab, cette fonction de transfert pourra être définie par :

```
s = tf('s');  
G = 5/(s^2 + 3*s + 2);
```

Ensuite, on pourra tracer :

```
step(G)      % reponse indicielle  
impulse(G)   % reponse impulsionnelle  
bode(G)      % diagramme de Bode  
nyquist(G)   % diagramme de Nyquist
```

## Complément enseignant

Exemple de résolution symbolique :

```
syms y(t)  
eqn = diff(y,t) + 2*y == 1;  
cond = y(0) == 0;  
sol = dsolve(eqn,cond)
```

La solution attendue est :

$$y(t) = \frac{1}{2} (1 - e^{-2t})$$

Exemple de résolution numérique :

```
f = @(t,y) -2*y + 1;  
[t,y] = ode45(f,[0 5],0);  
plot(t,y)  
grid on  
xlabel('Temps')  
ylabel('y(t)')  
title('Resolution numerique avec ode45')
```

Exemple de boucle fermée :

```
s = tf('s');  
G = 5/(s^2 + 3*s + 2);  
T = feedback(G,1);  
step(T)  
grid on
```

# 1 Application 1 : Calculs sur les polynômes

On considère les deux polynômes suivants :

$$p(x) = 10x^5 - 2x^2 + 10x$$

et :

$$q(x) = x^2 + 4$$

## Travail demandé

1. Consulter l'aide en ligne Matlab sur les commandes suivantes :

`poly`, `roots`, `conv`, `deconv`, `polyder`, `polyval`

2. Entrer les polynômes  $p$  et  $q$  sous Matlab.
3. Construire un polynôme  $S(x)$  possédant les racines :

$$1, \quad 5, \quad 6$$

4. Calculer les racines de  $p$  et  $q$ .
5. Calculer la dérivée  $p'(x)$ .
6. Calculer le produit  $p(x)q(x)$ .
7. Calculer le quotient et le reste de la division de  $p(x)$  par  $q(x)$ .
8. Calculer les valeurs de  $p(x)$  en :

$$x = 0 \quad \text{et} \quad x = -1$$

9. Tracer la courbe d'équation :

$$y = p(x)$$

sur l'intervalle :

$$[-10, 10]$$

# Correction de l'application 1

## 1. Aide Matlab

Pour consulter l'aide Matlab, on utilise :

```
help poly
help roots
help conv
help deconv
help polyder
help polyval
```

On peut aussi utiliser :

```
doc poly
doc roots
doc conv
doc deconv
doc polyder
doc polyval
```

## 2. Entrée des polynômes

Un polynôme est représenté dans Matlab par le vecteur de ses coefficients dans l'ordre décroissant des puissances.

Pour :

$$p(x) = 10x^5 - 2x^2 + 10x$$

on écrit :

$$p = [10, 0, 0, -2, 10, 0]$$

car :

$$p(x) = 10x^5 + 0x^4 + 0x^3 - 2x^2 + 10x + 0$$

Pour :

$$q(x) = x^2 + 4$$

on écrit :

$$q = [1, 0, 4]$$

Code Matlab :

```
p = [10 0 0 -2 10 0];
q = [1 0 4];
```

## 3. Polynôme possédant les racines 1, 5 et 6

On utilise la commande :

```
s = poly([1 5 6])
```

Résultat :

$$S(x) = (x - 1)(x - 5)(x - 6)$$

Développement :

$$S(x) = x^3 - 12x^2 + 41x - 30$$

Donc :

$$S = [1, -12, 41, -30]$$

#### 4. Racines de $p$ et $q$

Code Matlab :

```
rp = roots(p)
rq = roots(q)
```

Pour  $q(x) = x^2 + 4$ , on a :

$$x^2 + 4 = 0$$

donc :

$$x^2 = -4$$

et :

$$x = \pm 2j$$

Ainsi :

$$rq = 2j, -2j$$

Pour  $p(x)$ , Matlab donne une racine nulle et des racines complexes.

En effet :

$$p(x) = 10x^5 - 2x^2 + 10x = x(10x^4 - 2x + 10)$$

donc :

$$x = 0$$

est une racine de  $p$ .

#### 5. Dérivée de $p(x)$

La dérivée est :

$$p'(x) = 50x^4 - 4x + 10$$

Code Matlab :

```
dp = polyder(p)
```

Résultat :

$$dp = [50, 0, 0, -4, 10]$$

## 6. Produit $p(x)q(x)$

Code Matlab :

```
prod_pq = conv(p,q)
```

Calcul :

$$p(x)q(x) = (10x^5 - 2x^2 + 10x)(x^2 + 4)$$

$$p(x)q(x) = 10x^7 + 40x^5 - 2x^4 + 10x^3 - 8x^2 + 40x$$

Donc :

$$prod\_pq = [10, 0, 40, -2, 10, -8, 40, 0]$$

## 7. Quotient et reste de la division de $p(x)$ par $q(x)$

Code Matlab :

```
[quotient,reste] = deconv(p,q)
```

La division donne :

$$p(x) = (x^2 + 4)(10x^3 - 40x - 2) + (170x + 8)$$

Donc :

$$quotient = 10x^3 - 40x - 2$$

et :

$$reste = 170x + 8$$

Sous forme vectorielle :

$$quotient = [10, 0, -40, -2]$$

et le reste utile est :

$$reste = [170, 8]$$

## 8. Valeurs de $p(x)$ en 0 et -1

Code Matlab :

```
p0 = polyval(p,0)  
pm1 = polyval(p,-1)
```

Calcul :

$$p(0) = 10(0)^5 - 2(0)^2 + 10(0) = 0$$

Donc :

$$p(0) = 0$$

Pour  $x = -1$  :

$$p(-1) = 10(-1)^5 - 2(-1)^2 + 10(-1)$$

$$p(-1) = -10 - 2 - 10$$

$$p(-1) = -22$$

### 9. Tracé de $y = p(x)$ sur $[-10, 10]$

Code Matlab :

```
x = -10:0.01:10;  
y = polyval(p,x);  
  
figure(1)  
plot(x,y, 'b', 'LineWidth',1.5)  
grid on  
xlabel('x')  
ylabel('p(x)')  
title('Courbe du polynome p(x)')
```

## 2 Application 2 : Représentation graphique

On considère la fonction :

$$f_1(t) = \sin(t)e^{-2t}$$

### Travail demandé

1. Créer un vecteur ligne  $t$  avec un pas de 0,01.
2. En utilisant la commande `plot`, tracer  $f_1(t)$  dans une fenêtre `figure(1)`.
3. Annoter les axes :
  - axe des abscisses : temps ;
  - axe des ordonnées : Volts.
4. Donner un titre à cette courbe.
5. La courbe doit être de couleur rouge et marquée par des étoiles.
6. Dans une même fenêtre `figure(2)`, tracer sur l'intervalle :

$$t \in [-1, 1]$$

les courbes suivantes :

$$f_1(t) = \sin(t)e^{-2t}$$

en bleu,

$$f_2(t) = -\sin(t)e^{-2t}$$

en vert,

$$f_3(t) = e^{-2t}$$

en rouge,

$$f_4(t) = e^{2t}$$

en cyan.

7. En utilisant la commande `subplot`, tracer  $f_1$ ,  $f_2$  et  $f_3$  dans une même fenêtre `figure(3)`.

## Correction de l'application 2

### 1. Tracé de $f_1(t) = \sin(t)e^{-2t}$

On choisit par exemple :

$$t \in [0, 5]$$

avec un pas de :

$$0,01$$

Code Matlab :

```
t = 0:0.01:5;
f1 = sin(t).*exp(-2*t);

figure(1)
plot(t,f1,'r*')
grid on
xlabel('Temps')
ylabel('Volts')
title('Courbe de f1(t) = sin(t) exp(-2t)')
```

Remarque importante :

On utilise :

.\*

au lieu de :

\*

car les opérations sont effectuées élément par élément sur les vecteurs.

Ainsi :

```
sin(t).*exp(-2*t)
```

signifie que Matlab multiplie chaque valeur de  $\sin(t)$  par la valeur correspondante de  $e^{-2t}$ .

### 2. Tracé de quatre fonctions sur $[-1, 1]$

On définit :

$$t = -1 : 0,01 : 1$$

Les quatre fonctions sont :

$$f_1(t) = \sin(t)e^{-2t}$$

$$f_2(t) = -\sin(t)e^{-2t}$$

$$f_3(t) = e^{-2t}$$

$$f_4(t) = e^{2t}$$

Code Matlab :

```

t = -1:0.01:1;

f1 = sin(t).*exp(-2*t);
f2 = -sin(t).*exp(-2*t);
f3 = exp(-2*t);
f4 = exp(2*t);

figure(2)
plot(t,f1,'b','LineWidth',1.5)
hold on
plot(t,f2,'g','LineWidth',1.5)
plot(t,f3,'r','LineWidth',1.5)
plot(t,f4,'c','LineWidth',1.5)
hold off

grid on
xlabel('Temps')
ylabel('Amplitude')
title('Représentation de f1, f2, f3 et f4')
legend('f1 = sin(t)e^{-2t}', ...
       'f2 = -sin(t)e^{-2t}', ...
       'f3 = e^{-2t}', ...
       'f4 = e^{2t}')

```

### 3. Utilisation de subplot

La commande `subplot` permet de diviser une fenêtre graphique en plusieurs parties.  
Par exemple :

*subplot(3,1,1)*

signifie :

- 3 lignes;
- 1 colonne;
- première position.

Code Matlab :

```

figure(3)

subplot(3,1,1)
plot(t,f1,'b','LineWidth',1.5)
grid on
xlabel('Temps')
ylabel('f1(t)')
title('f1(t) = sin(t)e^{-2t}')

subplot(3,1,2)
plot(t,f2,'g','LineWidth',1.5)
grid on
xlabel('Temps')
ylabel('f2(t)')
title('f2(t) = -sin(t)e^{-2t}')

subplot(3,1,3)
plot(t,f3,'r','LineWidth',1.5)
grid on
xlabel('Temps')

```

```
ylabel('f3(t)')  
title('f3(t) = e^{-2t}')
```

## Résumé des commandes Matlab utilisées

| Commande             | Rôle                                                    |
|----------------------|---------------------------------------------------------|
| <code>poly</code>    | Construire un polynôme à partir de ses racines          |
| <code>roots</code>   | Calculer les racines d'un polynôme                      |
| <code>conv</code>    | Multiplier deux polynômes                               |
| <code>deconv</code>  | Diviser deux polynômes                                  |
| <code>polyder</code> | Calculer la dérivée d'un polynôme                       |
| <code>polyval</code> | Évaluer un polynôme en un point ou sur un vecteur       |
| <code>plot</code>    | Tracer une courbe                                       |
| <code>figure</code>  | Ouvrir une nouvelle fenêtre graphique                   |
| <code>grid on</code> | Afficher la grille                                      |
| <code>xlabel</code>  | Nommer l'axe des abscisses                              |
| <code>ylabel</code>  | Nommer l'axe des ordonnées                              |
| <code>title</code>   | Donner un titre au graphique                            |
| <code>legend</code>  | Ajouter une légende                                     |
| <code>hold on</code> | Garder le graphique actif pour tracer plusieurs courbes |
| <code>subplot</code> | Diviser une fenêtre graphique en plusieurs sous-figures |

## Script Matlab complet du TP

```
clc
clear
close all

%% Application 1 : Polynomes

p = [10 0 0 -2 10 0];
q = [1 0 4];

% Polynome ayant pour racines 1, 5 et 6
S = poly([1 5 6])

% Racines
rp = roots(p)
rq = roots(q)

% Derivee
dp = polyder(p)

% Produit
prod_pq = conv(p,q)

% Division
[quotient,reste] = deconv(p,q)

% Valeurs de p
p0 = polyval(p,0)
pm1 = polyval(p,-1)

% Trace de p(x)
x = -10:0.01:10;
y = polyval(p,x);

figure(1)
plot(x,y,'b','LineWidth',1.5)
grid on
xlabel('x')
ylabel('p(x)')
title('Courbe du polynome p(x)')

%% Application 2 : Graphismes

% Fonction f1 sur [0,5]
t = 0:0.01:5;
f1 = sin(t).*exp(-2*t);

figure(2)
plot(t,f1,'r*')
grid on
xlabel('Temps')
ylabel('Volts')
title('Courbe de f1(t) = sin(t) exp(-2t)')

% Quatre fonctions sur [-1,1]
t = -1:0.01:1;
f1 = sin(t).*exp(-2*t);
```

```

f2 = -sin(t).*exp(-2*t);
f3 = exp(-2*t);
f4 = exp(2*t);

figure(3)
plot(t,f1,'b','LineWidth',1.5)
hold on
plot(t,f2,'g','LineWidth',1.5)
plot(t,f3,'r','LineWidth',1.5)
plot(t,f4,'c','LineWidth',1.5)
hold off

grid on
xlabel('Temps')
ylabel('Amplitude')
title('Représentation de f1, f2, f3 et f4')
legend('f1','f2','f3','f4')

% Subplot
figure(4)

subplot(3,1,1)
plot(t,f1,'b','LineWidth',1.5)
grid on
xlabel('Temps')
ylabel('f1(t)')
title('f1(t) = sin(t)e^{-2t}')

subplot(3,1,2)
plot(t,f2,'g','LineWidth',1.5)
grid on
xlabel('Temps')
ylabel('f2(t)')
title('f2(t) = -sin(t)e^{-2t}')

subplot(3,1,3)
plot(t,f3,'r','LineWidth',1.5)
grid on
xlabel('Temps')
ylabel('f3(t)')
title('f3(t) = e^{-2t}')

```