

TP n° 1 :Initiation à Matlab

1. Introduction à Matlab

Matlab est un logiciel interactif permettant d'effectuer des calculs numériques complexes particulièrement utiles dans le domaine de l'ingénierie.

1.1 Vecteurs et matrices

1.1.1 Les vecteurs

On peut définir un vecteur x en donnant la liste de ses éléments:

```
x=[0.5 1 -3 5 7] % vecteur ligne
```

```
y=[1 ; 2 ;3] % vecteur colonne
```

ou en donnant la suite qui forme le vecteur: $x=1:0.5:5$

ou en utilisant une fonction qui génère un vecteur: $x=\text{linspace}(1,5,8)$

ou: $y=\text{logspace}(1,3,5)$

$\text{length}(x)$ %dimension du vecteur

1.1.2 Les matrices

Une matrice peut être construite de différentes manières ;

Soit : $A=[1\ 2\ 3 ; 6\ 7\ 8 ; 4\ 5\ 9]$

Ou : $A=\begin{bmatrix} 1 & 2 & 3 \\ 6 & 7 & 8 \\ 4 & 5 & 9 \end{bmatrix}$

ou bien : $v1=[1\ 2\ 3]$; $v2=[6\ 7\ 8]$; $v3=[4\ 5\ 9]$ et $A=[v1 ; v2 ; v3]$

Une matrice transposée se calcule avec l'apostrophe (') : $A^t=A'$

L'inverse d'une matrice A est calculé comme suit : $A\text{inv}=\text{inv}(A)$

La dimension de la matrice A est donnée par : $\text{Size}(A)$

Les valeurs propres d'une matrice sont calculées par : $\text{eig}(A)$

Le déterminant de la matrice A est calculé par : $\text{det}(A)$

Le rang d'une matrice A est calculé par : $\text{rank}(A)$

La trace de la matrice A est calculée par : $\text{trace}(A)$

1.1.3 Matrices et vecteurs particuliers

Les fonctions **ones** et **zeros** permettent de construire des matrices dont tous les éléments sont égaux à **1** respectivement à **0**. Voir également **eye** (matrice identité), **diag** (matrice diagonale), **linspace** (vecteur dont les composantes sont espacées linéairement entre deux

limites) et **logspace** (vecteur dont les composantes sont espacées logarithmiquement entre deux limites).

1.2 Graphiques

Pour ouvrir une fenêtre dans laquelle seront tracés des graphiques, il faut utiliser l'instruction **figure**.

1.2.1 Graphiques 2D

1.2.1.1 Tracée de courbes

On utilise l'instruction **plot** pour tracer un graphique 2D :

plot(x,y) Trace le vecteur y en fonction du vecteur x.

plot(t,x,t,y,t,z) Trace x(t), y(t) et z(t) sur le même graphique.

plot(t,z,'r--') Trace z(t) en trait mixte rouge.

stem(k,z) Trace z(k.T) en échantillons (pour les fonctions discrètes).

Exemple 1 : Tracer la courbe de la fonction $y = \sin(t)$; pour $t = -2*\pi:\pi/100:2*\pi$;

```
plot(t,y)           %Trace y(t) avec des échelles linéaires
grid                %Quadrille le graphe
title('Fonction Sinusoidale') % titre du graphe
xlabel('temps (s)') %commentaire sur l'axe des abscisses
ylabel('amplitude') % commentaire sur l'axe des ordonnées
```

1.2.1.2 Graphique multiple

On peut tracer plusieurs graphiques dans la même fenêtre en utilisant l'instruction **subplot**. **Subplot(m,n,p)** divise la fenêtre graphique en **m**x**n** sous graphiques, m et n étant le nombre de lignes et de colonnes respectivement, la variable **p** désigne le numéro du graphique.

Exemple 2: Tracer les courbes des fonctions $y1 = \sin(t)$ et $y2 = \cos(t)$; pour

```
t = -2*pi:pi/50:2*pi;
subplot(2,1,1)
plot(t,y1,'r')
subplot(2,1,2)
plot(t,y2,'b')
```

1.3 Création de fonction MATLAB

Une fonction MATLAB est un fichier M particulier dont la première ligne commence par «**function**». Une fonction peut être utilisée dans les expressions mathématiques ou dans les instructions MATLAB.

On définit la fonction **func** de la manière suivante :

function [vars1, . . . , varsn] = func(vare1, . . . , varem)

séquence d'instructions

où

– vars1, . . . , varsn sont les variables de sortie de la fonction ;

– vare1, . . . , varem sont les variables d'entrée de la fonction ;

– séquence d'instructions est le corps de la fonction.

Exemple de fonction :

```
function y = sinuscard(x)
```

```
z = sin(x); % Variable de stockage
```

```
y = z./x; % Résultat de la fonction
```

Cette fonction pourra être appelée par : `sincpi = sinuscard(pi)`;

1.5 Description des systèmes linéaires

Matlab permet de décrire des systèmes linéaires sous forme de représentation d'état ou sous forme de fonction de transfert et de passer d'une forme à l'autre.

Pour générer une fonction de transfert on utilise la fonction `(tf)% g=tf(num,den)`

Pour générer une représentation d'état on utilise la fonction `(ss)% sys=tf(A,B,C,D)`

Pour calculer la représentation d'état à partir de la fonction de transfert on utilise la fonction `tf2ss : %sys_ss=tf2ss(g)`

Pour calculer la fonction de transfert à partir de la représentation d'état on utilise la fonction `ss2tf : %g_tf=ss2tf(sys).`

1.6 Analyse des systèmes et simulation

a. Analyse d'un système

Un système linéaire étant représenté sous Matlab il peut être intéressant d'en étudier les propriétés.

La fonction *bode* trace le diagramme de Bode (gain et phase) du système.

La fonction *nichols* trace le diagramme de Black-Nichols. La fonction *ngrid* superpose l'abaque de Black (gain et fréquence en boucle fermée) sur le diagramme.

La fonction *pole* calcule les pôles du système.

La fonction *zero* calcule les zéros du système.

La fonction *damp* calcule les pôles et, pour chacun, son amortissement et sa fréquence.

La fonction *pzmap* trace les pôles et zéros dans le plan complexe.

b. Réponse d'un système

La fonction *impulse* trace la réponse impulsionnelle d'un système.

La fonction *step* trace la réponse à une entrée en échelon unitaire.

La fonction *lsim* trace la réponse à une entrée quelconque d'un système.

1.7 Travail à faire

1- Exécuter sous matlab les exemples cités ci-dessus.

2- Soit par exemple le système représenté par la fonction de transfert suivante :

$$G(p) = \frac{s^2 + 3s + 4}{s^3 + 2s + 1}$$

1- Déclarer sous Matlab cette fonction de transfert

2- Mettre ce système sous forme de représentation d'état.

3- Calculer les pôles du système et ces valeurs propres. Conclure.

4- Tracer la réponse impulsionnelle, indicielle et pour une entrée $e(t)=\sin(t)$, pour $t=0 :0.01 :5$. Faire ceci sur la même figure (Donner un titre à chaque courbe).

5- Calculer le gain statique et le temps de réponse du système.

6- Le système est t-il stable ? expliquer.

7- Mettre le système en boucle fermé. Est-ce que le bouclage change l'ordre du système ? expliquer. Est-ce que le bouclage améliore les performances du système bouclé par rapport au système en boucle ouverte ?