University mohamed khider

Département de génie électrique — 3ème année électromécanique

Enseignant : Dr. Okba KRAA — Matière: TP Systèmes Asservis

TP 1 Fonctions de matlab utilisées en asservissement

1. Objectifs du TP :

- Programmation et résolution des équations différentielles à l'aide du logiciel Matlab ;
- Détermination de la fonction de transfert d'un système ;
- présentation des différentes réponses temporelles et fréquentielles d'un système,

2. Résolution d'équation différentielle ordinaire

2. 1. Fonction ODE

Une équation différentielle ordinaire ODE (ordinary differential equation) est une équation reliant une fonction d'une variable réelle et ses dérivées, c'est à dire de la forme

$$y' = f(t, y)$$

- **Exemple 1 : équation différentielle du premier ordre**

Soit la fonction y1(t) soumise à l'équation différentielle :

$$2\dot{y}_1 + y_1 = 10$$

$$\Leftrightarrow \dot{y}_1 = \frac{10 - y_1}{2}$$

$$\text{Condition initiale : } y_1(t=0) = 5$$

Pour programmer cette équation, nous créons le fichier f10.m :

```
function dy=f10(t,y) ;

% Equation différentielle du 1er ordre

dy(1) = (10-y(1))./2 ;
```

Sur l'espace de travaille du Matlab, taper le code suivant : **[t , y] = ode23 ('f10' , [0 20] , 0) ;**

Plot(t,y)

- Qu'est-ce qui fait la fonction ode23 dans MATLAB ?

- Faire changer l'ordre d'ode23 au ode45 puis tracez y(t) dans le même intervalle **[0 20].**

- **Exemple 2 : équation différentielle du deuxième ordre**

$$\ddot{y}_2 + 0.5\,\dot{y}_2 + 2y_2 = 0$$

On pose : $\dot{y}_2 = y_1$

$$\begin{cases} \dot{y}_1 = -0.5y_1 - 2y_2 \\ \dot{y}_2 = y_1 \end{cases}$$

Conditions initiales :

$$\begin{cases} y_1(t=0) = \dot{y}_2(t=0) = 0 \\ y_2(t=0) = 2 \end{cases}$$

Créer le fichier f11.m :

```
f11.m - Bloc-notes
Fichier  Edition  Format  Affichage  ?
function dy=f11(t,y) ;

% Equation différentielle du 2ème ordre

dy(1) = -0.5.*y(1) - 2.*y(2) ;
dy(2) =  y(1)  ;
```

dy=ydy'

[t , y]=ode23 ('f11' , [0 10] , [0 2])

Plot (t,y)

2.2 Ffonction dsolve

La syntaxe pour résoudre une équation différentielle simple est **dsolve('eqn').** La fonction renvoie une solution formelle de l'équation différentielle définie par l'expression symbolique eqn.

D : représente la dérivée première, D2 la dérivée second..etc, les constantes arbitraires de la solution sont notées C1 C2..etc.

Ezplot(y,[tmin,tmax] : permet de ploter la solution dans l'intervalle de sa variable [tmin,tmax].

Exemple 3

Fonction du 1er ordre

soit l'équation : 2*Dy+y=10

pour déterminer la solution formelle de cette équation on écrire sous Matlab :

y=dsolve('2*Dy+y=10', 'y(0)=5',t)

ezplot(y,[0,10])

Fonction du 2ème ordre : D2y+0.5*Dy+2*y=0

y=dsolve('D2y+0.5*Dy+2*y=0','y(0)=2, Dy(0)=0',t)

ezplot(y,[0,60])

3. Fonction de transfert d'un système

3.1 Définition

Une fonction de transfert permet de représenter un système linéaire invariant via sa relation entrée-sortie. Elle est obtenue en appliquant une transformée de Laplace aux équations différentielles qui décrivant la dynamique du système.

$$T(p) = \frac{S(p)}{E(p)} = \frac{a_0 + a_1.p + \cdots + a_m.p^m}{b_0 + b_1.p + \cdots + b_n.p^n}$$

L'écriture d'une fonction de transfert T(p) sous Matlab se fait donc comme suit :

num= [am ……a1 a0]

den=[bn…. b1 b0]

H=printsys(num,den)

G= tf(num,den)

ou soit directement par :

s=tf('s')

H=(am*s^m+…a0*s)/(bn*s^n+…b0*s)

Si on savait les poles, les zeros et le gain d'une FT, on peut utiliser la fonction zpk pour la définir cet en matlab.

Soit une FT :G=
```
   4  (s+1)
 -----------
 (s+1)  (s+2)
```

Cette fonction peut s'ecrire sous matlabe comme suite : G= zpk([-1],[-1 -2],4)

Exemple 4

Ecrire sur Matlab les fonctions suivantes :

$$H1 = \frac{10}{2\ \ s+1} \qquad H2 = \frac{10\ s+0.5}{s^2+2\ s+10}$$

Calculer leurs pôles et zéros utilisant les fonctions pole et zero, respectivement.

3-1 Réponse impulsionnelle et indicielle

On peut tracer la réponse impulsionnelle et indicielle d'un système s'écrit par sa fonction de transfert H, par les fonctions **impulse** et **step** respectivement**.**

```
ltiview('plottype',sys)
```

step (H1)

impulse(H1)

3.2 Réponses fréquentielles du système :

Bode (H1)
Nyquist(H1)

ltiview('*plottype*',sys): permet tracer la réponse indiquée par 'plottype' (plottype peut-être une des fonctions précédentes) du système 'sys.

4- Réponse d'un système pour une entrée quelconque :

On utilise la fonction lsim

```
t=0 :0.01 :4 ;
e=t ;
y=lsim(H2,e,t) ;
```

-système à retard **sys=tf([2],[3 1],'inputdelay',0.1)**