**Formation au Logiciel de Matlab**

**TP 1 - Initiation au logiciel Matlab**

1.1 - Généralités

1.2 - Présentation des menus et des principales commandes

.Définition des vecteurs et des matrices

.Construction de vecteurs

.Calcul matriciel

1.3 - Fonctions particulières

1.4 - Fonctions graphiques

1.5 - Utilisation de la "Control" toolbox

.Saisie d'une fonction de transfert ; fonction de transfert en boucle fermée

.Réponse en fréquences (Bode, Black)

.Réponse indicielle

1.6 - Exemples d'application

Réponse indicielle et réponse en fréquences d'un procédé du second ordre à amortissement variable

**1-Généralités**

- Créer un répertoire Matdat dans lequel seront stockés vos fichiers

Le lien pour accéder à la « licence site » gratuite jusqu’au 30 juin 2020.

•            Vous pourrez télécharger toutes les toolboxes listées ci-dessous sur les ordinateurs personnels.

•            Utilisez MATLAB Online : accédez à nos toolboxes via un navigateur internet.

<https://www.mathworks.com/licensecenter/classroom/COVID-19_Access/>

Le lien pour accéder à notre bibliothèque de cours en ligne : des ressources utiles pour améliorer les compétences de vos étudiants sur nos solutions.

<https://www.mathworks.com/learn/teaching-with-matlab.html>

Avec ; en fin de ligne : n’affiche pas le résultat

Sans ; en fin de ligne : affiche pas le résultat

- Calcul matriciel

A=[1 2 3 ; 4 5 6 ; 7 8 9] crée une matrice A

B=inv(A) Inverse de la matrice A

C= det(A) Déterminant de la matrice A

D=A’ Matrice transposée de A

**A(i , j)** : élément ligne i, colonne j de la matrice A

**A(i , :)** : tous les élément de la ligne i, de la matrice A

**A( : ,j)** : tous les éléments de la colonne j de la matrice A

- Vecteurs :

**x(1:5)** : tous les éléments du vecteur x, du 1er au 5ème élément

**x(:)** : tous les éléments du vecteur x

- Construction de vecteurs

> élément espacés linéairement, exemple vecteur temps : t=**linspace** (tmin, tmax, Nt)

Nt valeurs réparties linéairement de tmin à tmax

> élément espacés linéairement, exemple vecteur temps : =**logspace**(Pmin, Pmax, N)

Nw valeurs réparties logarithmiquement de 10^Pmin à 10^Pmax

**2-Fonctions**

Matlab distingue les majuscules et minuscules

**flêche up :** rappelle la commande précédente dans la fenêtre de commande

**clc** efface la fenêtre de commande de Matlab

**close all** ferme les figures ouvertes

**clf** ferme la figure désignée

**clear all** efface le workspace de Matlab

**format long e** format étendu

**whos** affiche l’espace de travail

**size** retourne la dimension d’une matrice

**length** retourne la taille d’un vecteur

**round**

**ceil**

**zeros** crée une matrice de 0

**ones** crée une matrice de 1

**eye** matrice identité

**&** ET logique

**|** OU logique

**help :** dans **le menu** ou dans la fenêtre de commande : **>help nom\_de\_la\_commande**

**3- Fonctions graphiques**

**figure(i)** : pour créer une figure n°i

-Représentation temporelle

**plot(t, y)**

option de tracé : help plot

**title** title(‘Nom de la figure’)

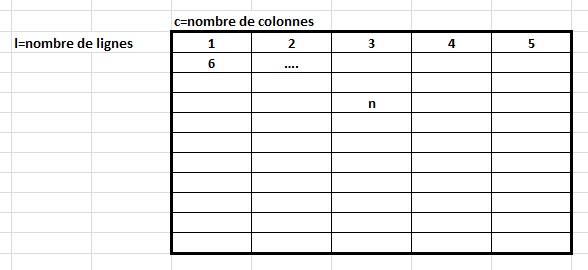
**xlabel** xlabel(‘temps(s)’)

**ylabel** ylabel(‘réponse temporelle’)

**grid**

-Ex : Tracer les fonctions sinus(t) et cos(t)

**subplot (lcn) divise la figure en l x c sous figures**



-Représentation fréquentielle

**semilogx(,gain)** trace une figure en semilogarithmique

-Ex : Tracer le module et l’argument d’un intégrateur

1/p > 1/j

**4-Boucles**

**for**

**while**

**if**

-Ex1 : Construire une matrice 5X20 dont tous les éléments d’une colonne sont égaux au N° de la colonne

A =

Columns 1 through 14

1 2 3 4 5 6 7 8 9 10 11 12 13 14

1 2 3 4 5 6 7 8 9 10 11 12 13 14

1 2 3 4 5 6 7 8 9 10 11 12 13 14

1 2 3 4 5 6 7 8 9 10 11 12 13 14

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Columns 15 through 20

15 16 17 18 19 20

15 16 17 18 19 20

15 16 17 18 19 20

15 16 17 18 19 20

15 16 17 18 19 20

-Ex2 : Faire la somme de 2 vecteurs de tailles différentes en respectant les pondérations

A=[1 2 3 4 5 6]

B= [7 8 9]

C=A+B=[1 2 3 11 13 15]

**5-Control Toolbox**

-Définition d’une fonction de transfert

-Avec 2 vecteurs

NumG=[1 2 3 ]

DenG=[4 5 6 1]



DenG=[4 5 6 **0**]



-Avec la fonction **tf**

G=tf(NumG, DenG)

[NumG, DenG]=**tfdata**(G,'v')

-Boucle ouverte : B=C\*G

-Solution 1 : [NumB, DenB]=**series**(NumC, DenC, NumG, DenG)

NumG=[1 2 3 ]

DenG=[4 5 6 1]

NumC=[1 2 ]

DenC=[6 1]

[NumB, DenB]=**series**(NumC, DenC, NumG, DenG)

NumB = 0 1 4 7 6

DenB = 24 34 41 12 1

-Solution 2 : avec les objets

NumG=[1 2 3 ]

DenG=[4 5 6 1]

NumC=[1 2 ]

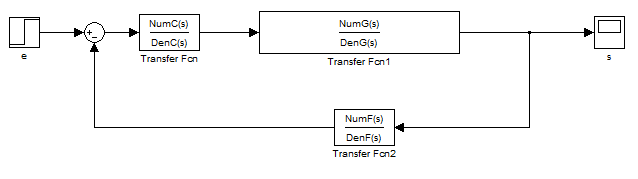
DenC=[6 1]

G=tf(NumG, DenG)

C=tf(NumC, DenC)

B=C\*G

-Solution 3 : avec Simulink



-Boucle fermée : H=(C\*G)/(1+C\*G\*F)=B/(1+B\*F)

-Solution 1: [NumH, DenH]=**feedback**(NumB, DenB, NumF, DenF)

Retour unitaire : : [NumH, DenH]=feedback(NumB, DenB, 1, 1)

NumG=[1 2 ]

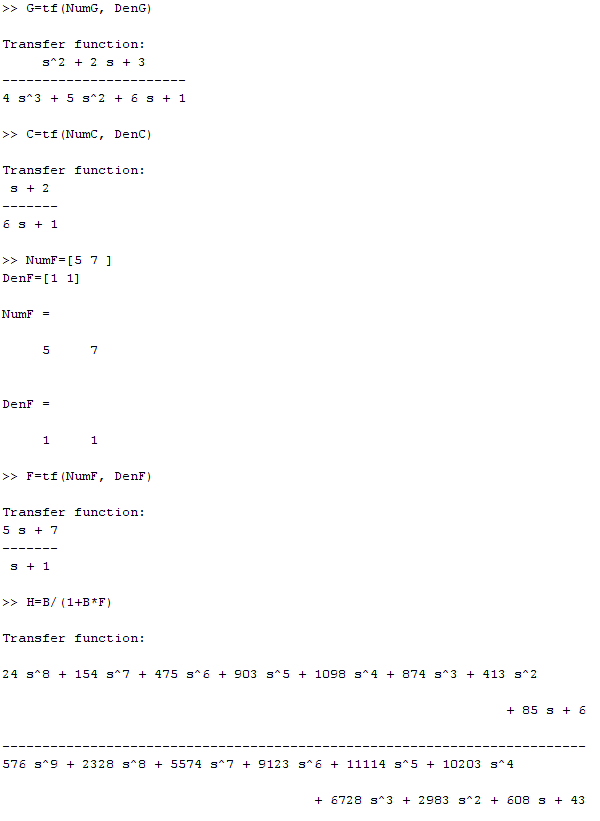
DenG=[4 5 6 1]

NumC=[1 2 ]

DenC=[6 1]

NumF=[5 7 ]

DenF=[1 1]



-Solution 2 : avec les objets

G=tf(NumG, DenG)

C=tf(NumC, DenC)

F=tf(NumF, DenF)

H=B/(1+B\*F)

-Solution 3 : avec Simulink

-Diagrammes de Bode

>> bode(NumG, DenG, )

>> bode(G, ) --> Trace les diagrammes avec le module ***en dB***

>>[g,p]= bode(NumG, DenG, ) --> Crée 2 vecteurs : module g (***pas en dB***) et argument p

calcul du module en dB : gdB=20\*log10(g)

>>Pour récupérer le module et l’argument à une pulsation donnée, par exemple u :

[gu ,pu ]= bode(NumG, DenG, u )

-Lieu de Nichols

>> nichols(NumG, DenG, )

>> nichols(G, )

ngrid

NumG=[1 ]

DenG=[4 6 1]

G=tf(NumG, DenG)

-Réponse indicielle

**step**(NumH, DenH, t)

[y,x]=**step**(NumH, DenH, t)

Ex :

>> t=linspace(0,30,100);

>> figure(2)

>> step(NumH, DenH, t)

>>grid

**TP 2 - Matlab avancé**

C:\Nom\Matdat

**6-Exercice : Etude d’un système du 2ème ordre**

Etude temporelle et fréquentielle

%G(s)=G0/(1+(2z/wn)s + s^2/wn^2)

Saisir les paramètres G0 et wn au clavier > help input

Echelles automatiques en temps et en > cf après

Etude pour 10 valeurs du facteur d’amortissement z =[0.2 2] > z vecteur avec linspace

Superposer les réponses > help hold on / hold off

Etude temporelle : Réponse indicielle > step

Etude fréquentielle Diagrammes de Bode > bode

Lieu de Nichols > nichols

Cartouches de légende > help legend

**- Input**

G0 = input('Gain statique =');

wn = input('Pulsation propre non amortize wn =');

**-Echelles automatiques**

\*Echelle en temps

tr=3/wu

tr=3/wn

tmin = 0; tmax = 20/wn; Nt=100;

t=linspace(tmin, tmax, Nt);

\*Echelle en pulsations

wn

nw = 100;

ndec = 2;

pwmin = round(log10(wn/(10^ndec)));

pwmax = round(log10(wn\*(10^ndec)));

w = logspace(pwmin,pwmax,nw);

**-Vecteur z**

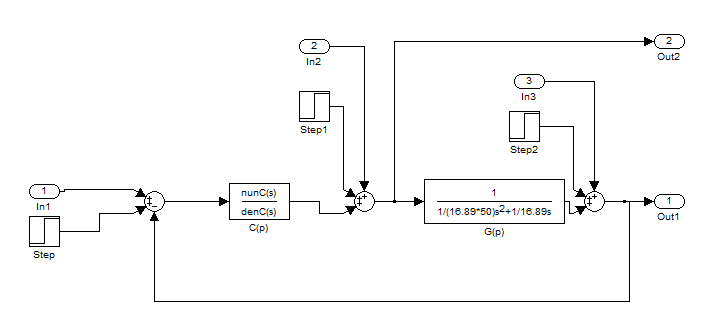
zmin = 0.2 ; zmax = 2; Nz=10;

z=linspace(zmin, zmax, Nz);

NumG=[G0] ;

DenG=[1/wn^2 2\*z(i)/wn 1]

**TP3- Synthèse d’un régulateur PIDF**



avec

et G0=1 ; p0=16.89 rad/s ; 0=50 rad/s

**Cahier des charges : faire un programme qui permette :**

**Programme principal : ptfcomplet**

***Init***

**Menu principal**

**Saisie procédé G(s)**

***Saisie à l’aide d’une boîte de dialogue***

***NumG=[ … ]; DenG=[ … ];***

***Echelles automatiques en t et w***

**Etude du procédé G(jw)**

***bode***

***nichols***

***Retour***

**Synthèse régulateur PIDF**

***Dans une fonction***

***A l’aide d’une boîte de dialogue***

***Saisie wu, Mphi, alpha***

***Appel de la fonction f\_pidf***

***Retour de la fonction : NumC=[ … ]; DenC=[ … ];***

**bode(NumC, DenC) ;**

**Etude de la boucle ouverte Beta(jw)=C(jw)\*G(jw)**

***[NumB, DenB]=series(NumC, DenC, NumG, DenG);***

***ou:***

***C=tf(NunC, DenC) ;***

***G=tf(NumG, DenG) ;***

***B=C\*G***

***bode B***

***nichols B***

***Retour***

**Etude de l’asservissement en t et en w**

***Réponse indicielle (s et u)***

***Fonction asservissement S(jw)/E(jw)***

***Fonction régulation S/Pu***

***Fonction régulation S/Ps***

***Fonction de sensibilité U/E***

***Fonction de sensibilité U/Pu***

***Fonction de sensibilité U/Ps***

***Retour***

**Exit**

**> help memu**

> boîte de dialogues

>echelles atomatiques

------

**1-Init**

Reprendre l’initialisation

clear all;

clc;

close all;

format long e;

%G(s)= G0/(s/wp0\*(1+s/w0))

**2-Boîte de dialogue**

prompt = {'Gain statique','wp0','w0'};

dlgtitle = 'Procede';

line = 1;

def = {'1','16.89','50'};

procede = inputdlg(prompt, dlgtitle, line, def);

G0 = str2num(strvcat(procede(1)));

wp0 = str2num(strvcat(procede(2)));

w0 = str2num(strvcat(procede(3)));

avec

NumG = [G0];

DenG = [1/(wp0\*w0) 1/wp0 0];

G=tf(NumG, DenG) ;

**4-Echelles automatiques**

W00 = min(w0,wp0);

Startt = 0; Stopt = 200/w00; Nt=100;

t=linspace(Startt, Stopt, Nt);

nw = 100;

ndec = 2;

pwmin = round(log10(w00/(10^ndec)));

pwmax = round(log10(w00\*(10^ndec)));

w = logspace(pwmin,pwmax,nw);

**5-Menu**

K1=0;

while(K1~=6)

K1=menu('Titre Menu principal', 'Saisie procédé G(s)', ' Etude du procédé G(jw)', 'Synthèse régulateur PIDF', 'Etude Beta(jw)', Etude de l’asservissement en t et en w', 'Exit');

if(K1==1)

….

end;

if(K1==2)

K2=0;

while(K2~=3)

K2 = menu('Etude du procédé G(jw)', 'Bode',

'Nichols', 'Retour');

if(K2==1)

testk2 ok

end ;

if(K2==2)

…

end ;

end;

if(K1==3)

K3=0;

while(K3~=3)

K3 = menu('Etude du procede G', 'Bode', 'Nichols', 'Retour');

if(K3==1)

…

end ;

….

…

end ;

end;

……

end;