Travaux dirigés
De Technique de Simulation Numérique
TD1

Classe : GEA2

Enseignant : ben abdallah a

Exercice 1

En utilisant le langage de programmation **Matlab** Créer une fonction **code** qui permet le contrôle du code d'accès d'un compte, en utilisant la structure **while.....end**. Si en se trompe 3 fois en tapant le code en aura un message d'erreur. Le code d'accès doit être une chaine ~~alpha~~numérique.

Exercice 2 :

En utilisant le langage de programmation **Matlab** créer une fonction qui permet de supprimer le blanc de toute position d'une chaine.

Exercice 3 :

En utilisant le langage de programmation **Matlab** créer une fonction qui permet l'extraction des n premiers caractères d'une chaine.

Exercice 4 :

On se propose de crypter une chaine de caractères de longueur n ajoutant au code ASCII de chaque caractère une quantité telle que le résultat ne dépasse pas 127.

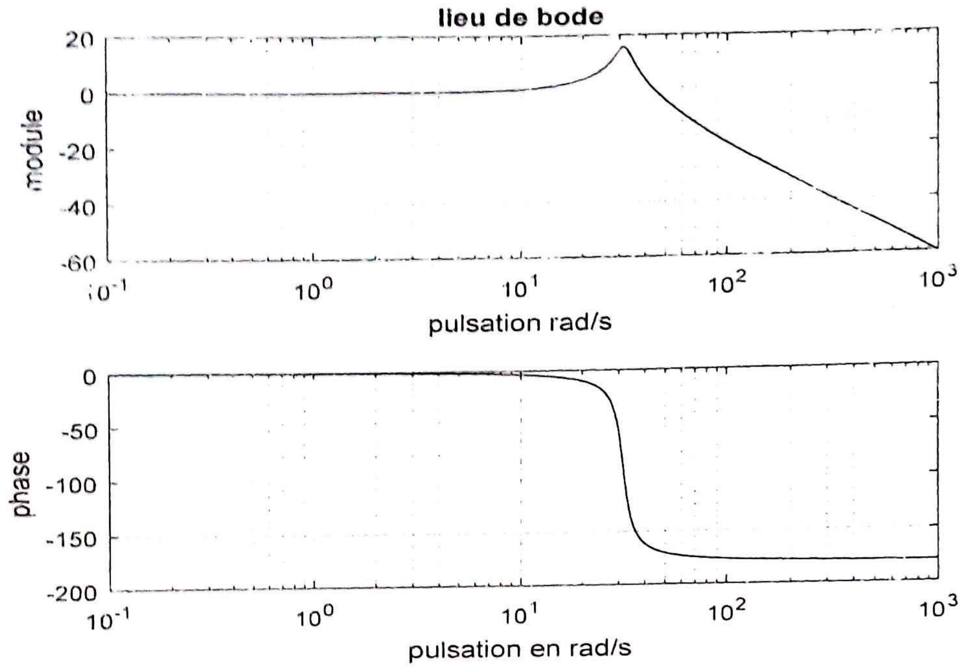
Exercice 5 :

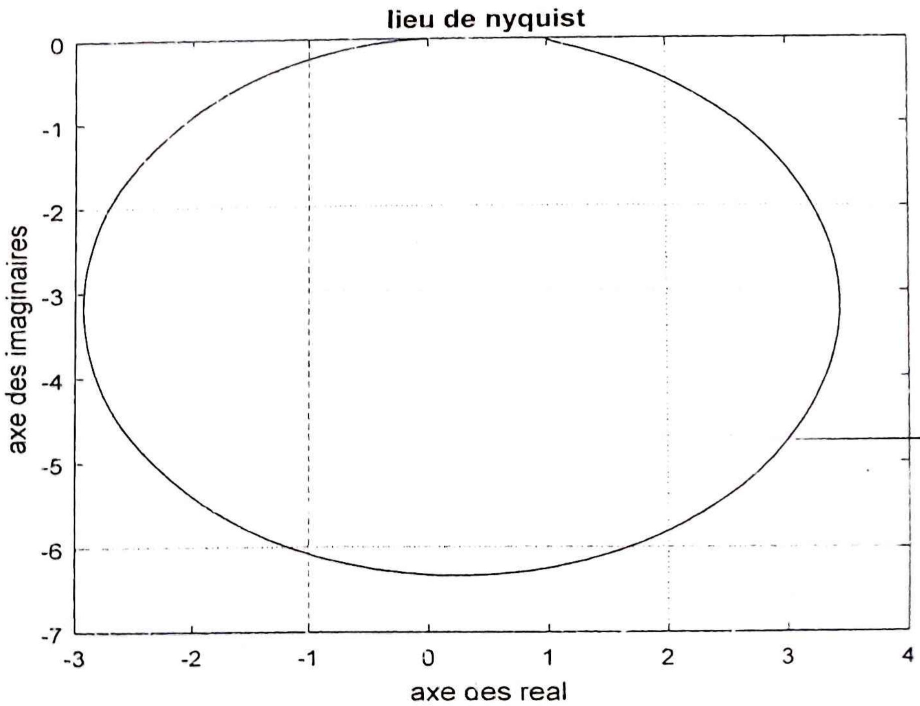
On se propose de créer une fonction en usant du langage de programmation **matlab** qui permet de calculer la surface et le périmètre d'un cercle, l'argument d'entrée de la fonction étant le rayon du cercle qui doit être saisie par l'utilisateur et les arguments de sortie sont la surface et le périmètre.

Exercice 6:

En vous servant du langage de programmation **matlab** créer une fonction **dichotomie** qui permet de chercher le zéro d'une fonction par la méthode numérique de recherche d'un zéro par dichotomie sur un domaine [a , b]. Les arguments d'entrées étant : le tolérance **tol** les bornes de l'intervalle **a** et **b** et la fonction **f** , l'argument de sortie étant le zéro de la fonction.

```
% circuit rlc : comportement fréquentiel
r=5;l=1;c=100*10e-6;
w=[0:0.1:1000];
h=1./(r*c*(j*w)-l*c*w.^2+1);
module=abs(h);
phase=angle(h);
figure(1);
%tracage du lieu de bode
subplot(2,1,1);semilogx(w,20*log10(module),'k');
xlabel('pulsation rad/s');
ylabel('module');grid;
title('lieu de bode')
subplot(2,1,2);semilogx(w,(phase/pi)*180,'k');
xlabel('pulsation en rad/s');
ylabel('phase');grid;
%tracage du lieu de nyquist
partie_r=real(h);
partie_im=imag(h);
figure(2);
plot(partie_r,partie_im,'k');
xlabel('axe des real');
ylabel('axe des imaginaires');
title('lieu de nyquist');grid;
```


lieu de bode
module
pulsation rad/s
phase
pulsation en rad/s


lieu de nyquist
axe des imaginaires
axe des real

Travaux Dirigés De Techniques de Simulation Numérique

Classe : GEA2

Série 2

Exercice 1 (DR2005)

Un système linéaire invariant dans le temps (LTI), d'entrée u et de sortie y est caractérisé par l'équation différentielle :

$m\ddot{y} + f\dot{y} + ky = u$

1/ on le suppose au repos $y(0) = \dot{y}(0) = 0$

En utilisant le langage de programmation **Matlab,** créer un programme qui permet de :

a/ déterminer et afficher la pulsation naturelle wn, le coefficient d'amortissement ksi et le gain statique kc du système. Le script doit recueillir l'affectation des 3 paramètres : k=3, m=2 et f=1.

b/ Tracer la réponse à un échelon unitaire, mesurer et afficher la valeur du dépassement.

c/ tracer le diagramme de bode du système, déterminer et afficher la pulsation de résonance .

2/ pour $y(0) = \dot{y}(0) = 1$, tracer la réponse du système à ces conditions initiales .

Exercice 2

Soit un processus régi par l'équation différentielle suivante :

$\dot{y}(t) + y(t) = u(t)$

En utilisant le langage de programmation **Matlab,** créer un programme qui permet de :

1/ Tracer la réponse du système à une rampe u(t) = at, avec a = 2.

2/ Tracer la réponse indicielle du système en boucle fermée à retour unitaire $y_1(t)$. Calculer et afficher le temps de réponse et l' erreur statique du système.

3/ On désire corriger la précision du système, pour k=[1 5 10], tracer les réponses indicielles on boucle fermée pour les différentes valeurs de k sur le même graphique, afficher pour chaque cas la valeur du gain statique.

4/ Tracer le lieu de Nyquist du système en boucle ouverte pour k = 1, afficher la valeur de la partie imaginaire lorsque la phase du système est – 45° .

Devoir de contrôle
Technique de Simulation Numérique

%classe : GEA2 % date : 05/06/21 % Durée : 1H30mn% Documents : autorisés % enseignant : Ben Abdallah. A

Exercice 1 (6points)

En vous servant du langage de programmation **Matlab**, créer une fonction qui permet de tracer la réponse d'un système quelconque a un échelon d'accélération, calculez et afficher l'écart d'accélération.

Exercice 2 (6points)

On cherche une approximation exponentielle de type : $y_a = ae^{bx}$

La méthode de **newton** donne les expressions suivantes de ces coefficients :

$$b = \frac{\sum_{k=1}^{n}[\,x(k)\ln y(k)\,] - \frac{1}{n}\sum_{k=1}^{n}x(k) * \sum_{k=1}^{n}[\ln y(k)]}{\sum_{k=1}^{n}x(k)^2 - \frac{1}{n}\left[\sum_{k=1}^{n}x(k)\right]^2}, \qquad a = \frac{\sum_{k=1}^{n}\frac{e^{bx(k)}}{y(k)}}{\sum_{k=1}^{n}\frac{e^{2bx(k)}}{y(k)^2}}$$

En vous servant du langage **Matlab**

1.1/ Créer une fonction **Newton** qui permet de calculer les coefficients a et b ainsi que l'approximation y_a .

1.2/ On dispose de deux séries x et y de valeurs suivantes :

x	0	2	4	6	10
y	2.1	4.7	6.7	12	39.4

Pour le cas ci-dessus calculer l'approximation y_inter (en vous servant de la fonction **Newton**).Ainsi que l'erreur d'approximation. Tracer l'évolution y , yinter et l'erreur sur le même Graphique.

DC . TSN

Exercice 1 :

f^{ct} permet de tracer la réponse d'un système quelconque à un échelon d'accélération

```
function [sys] = reponse
  a = 6
  t = [0 : 1.2 : 10]
  e = (a/2)* t.^2.
  Num = [2  1]
  Den = [5  10]
  Sys = tf (Num, Den)
  lsim (sys, e, t)
```

Exercice 2 :

$y_a = a e^{bx}$

```
function [ya] = Newton(x, y, n)
  for  k = 1 : n
    c = c + x(k)* log (y(k)) ;
    d = d + x(k) ;
    e = log (y(k)) + e ;
    f = f + x(k)^2 ;
    g = g + x(k) ;
  end
  b = (c - (1/n)* d*e)/(f - 1/n g^2)
  for  k = 1 : n
    h = h + (exp(b*x(k))) /y(k)
    N = i + (exp (2*b *x(k) )) /y(k)^2
  end
  a = h / n ;
  ya = [ ] ;
  for k = 1 : n ;
  Ya = [ ] ;
  for  k = 1 : n
    ya = a* exp (b * x(k)) ;
    ya = [ Ya  ya ] ;
  end
end
x = [0  2 4  6  10] ;
y = [1  5  7  8  9] ;
n = rank(x) ;
Ya = Newton (x, y, n) ;
figure (1)
plot(x, y) ;
plot (x, Ya) ;
for  k = 1 : n
  e = y(2) - Ya(k)
  E = [e  E] ;
end
plot [x, E] ;
```

Devoir de Synthèse
De Technique de Simulation Numérique

% Classe : GEA2% Date : 22/06/ 2021% Durée: 2H00mn% Documents : Autorisés% Enseignant : Ben Abdallah A

Exercice 1 (12 points) :

Soit le système décrit par les équations différentielles suivantes :

$$\begin{cases} J\dfrac{d\Omega(t)}{dt} + f\,\Omega(t) = C_m + C_r \\ L\dfrac{di(t)}{dt} + Ri(t) = U - K_e\Omega(t) \end{cases} \qquad \text{On posant } x = [\theta \ \Omega \ i]^T, \ \text{sachant que } \Omega = \frac{d\theta}{dt}$$

1.1/avec J est l'inertie du moteur, f est le coefficient du frottement, C_r est le couple de charge(<0), L est l'inductance de l'armature, R est la résistance de l'armature, U est la tension d'alimentation de l'armature et $K_e\Omega(t)$ est la force électromotrice (fem). En vous servant du langage de programmation **matlab**, calculer et afficher le modèle d'état du système sous la forme : $\dot{x} = Ax + BU + e$.

On donne :

$K_e=6\times10^{-3}v/rad/s$, $K_m=5\times10^{-3}Nm/a$, $R=5\ ohm$, $L=7.2mh$, $J=10^{-4}kg\ m^2$, $f=0.02Nm/s$ et $C_m(t)=K_m i(t)$.

1.2/ Tracer la variation des états (courant et vitesse) du système a un échelon de consigne en tension de 220v et un couple de charge(e=0).

1.3/ On désire connaitre la valeur maximale du courant. Déterminer et afficher la valeur du courant nominale sur le graphique.

1.4/On désire corriger le système avec un PI : $K_r(s) = k_p + \dfrac{1}{k_i s}$

1..4.1/Donner la fonction de transfert du système en boucle ouverte corrigée.

1.4.2/Tracer le lieu de **bode** du système corrigé, déterminer la marge de phase.

1.4.3/Tracer la réponse du système en boucle fermée corrigée pour un échelon de consigne de 220v et un échelon de perturbation (couple de charge) de -1.2 rad/s retardé de 5s. Calculer et afficher le temps de réponse à ±5%.

1.4.4/ Donner le schéma de câblage du système corrigé sur **simulink**.

On donne : $k_p=16$, $k_i=1$.

Exercice 2 (8 points) :

On cherche à calculer l'intégrale d'une fonction continument dérivable sur un intervalle [a,b] par la méthode de **simpson** . En vous servant du langage de programmation **matlab**, créer une fonction **simpson** qui permet de :

2.1/ Calculer $\int_a^b f(x)dx$ par la méthode de **simpson.**

-L'approximation de **simpson** est donnée par :

$$\int_a^b f(x)dx = \sum_{k=1}^{n} \frac{h}{6}\left(f(x_k) + 4f\left(\frac{x_k + x_{k+1}}{2}\right) + f(x_{k+1}) \right)$$

2.2/ Calculer l'intégrale de la fonction : $\int_0^{\frac{\pi}{4}} sin(x)dx$: