

EXAMEN DE GÉNIE INFORMATIQUE

Durée 1h00.

Tout document autorisé.

Communication avec l'extérieur (internet, téléphone ou autre) interdit.

Collaboration avec d'autres étudiants interdite.

1 • INTRODUCTION.

1.1 • But de l'examen.

Hormis le fait de tester vos connaissances sur l'utilisation du logiciel MATLAB, cet examen propose la mise en œuvre d'expérimentations de techniques d'identification de type moindres carrés. Tout ce qui concerne les bases théoriques de ces expérimentations vous est donné dans le sujet.

Dans cet examen, on vous donne un programme très simple. Il vous est demandé de le transformer pour rendre plus modulaire. Pour vous noter, nous lancerons simplement votre programme. Si tout fonctionne correctement et que votre programme est bien commenté, vous aurez la note maximale. Dans le cas contraire, les points vous seront attribués en fonction de la qualité de votre travail de programmeur. Pensez à commenter votre programme !!!

Si vous voulez réussir cette épreuve, il vous faut **très bien vous organiser**, ne mettre aucune valeur en dur (par exemple la dimension des vecteurs ou des matrices doivent toujours dépendre de variables) et écrire tout bien proprement, sans quoi vous aurez toutes les peines du monde à enchaîner les différentes parties de cette épreuve.

1.2 • Restitution de votre travail.

Votre travail doit être restitué en fin d'épreuve sous la forme d'un (ou plusieurs) fichier(s). Vous le déposerez dans un dossier portant votre nom sous le répertoire : /commun/master/eea/fmee131/ au maximum 5mn après la fin de l'épreuve! Toute minute de retard est retirée sur la note (1mn = 1pt en moins) - la date de dépôt faisant foi. AUCUN renseignement supplémentaire ne vous sera donné sur cette restitution. **La restitution de votre travail fait partie de l'épreuve.**

Vous devez déclarer clairement vos variables en début de fichier. Evitez absolument d'avoir des valeurs écrites "en dur" dans le programme.

2 • MOINDRES CARRÉS.

2.1 • Code de base.

Le code qui vous est donné permet de générer un tronçon de sinusoire bruité. Après affichage de ce tronçon (sur la figure 2) on procède à l'identification moindres carrés de la parabole d'équation $x = at^2 + bt + c$ passant au plus près des points générés. Cette

parabole est ensuite affichée sur la figure 3. Copiez ce code dans un fichier que vous nommerez `MonProgramme.m`. Faites le tourner (Attention aux répertoires).

```
clear all ;
close all ;

% definition de la base de temps
temps = 0:0.01:0.3 ;

% creation du troncon de sinusoide
x = sin(2*pi*temps) ;

% affichage de ce troncon non bruité
figure(2) ; clf ; hold on ; plot(temps, x, 'b') ;

% bruitage du troncon de sinusoide
x = x + 0.1*randn(size(x)) ; plot(temps, x, 'r*') ;

% estimation moindres-carres des parametres d'une parabole passant au mieux
% par les points generes

Matrice_A = [ temps.^2 ; temps ; ones(size(temps)) ]' ;
Matrice_AA = Matrice_A'*Matrice_A ;
Matrice_AB = Matrice_A'*x' ;

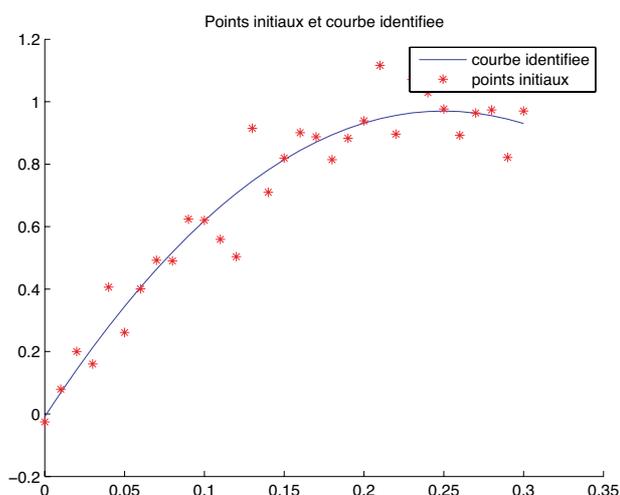
L = Matrice_AA^-1 * Matrice_AB ;

% estimation des points de la paraboloide
y = ( L(1) * temps.^2 ) + ( L(2) * temps ) + L(3) ;

% affichage de la paraboloide superposee sur les points originaux
figure(3) ; clf ; hold on ; plot(temps, y, 'b') ; plot(temps, x, 'r*') ;
```

2.2 • Modification de l'affichage.

Modifiez l'affichage de façon à faire apparaître des titres pour les figures ainsi que des légendes pour chaque courbe.



Faites en sorte que les figures apparaissent l'une à droite et l'autre à gauche de l'écran.

2.3 • Fonctions.

Remplacez la partie identification par une fonction. La fonction doit avoir un prototype de façon à pouvoir être appelée par :

```
>> L = IdentificationParabole(temps, x) ;
```

et ce quel que soient les dimensions de `temps` et de `x`.

2.4 • Estimation de l'erreur d'identification.

L'erreur d'identification est caractérisée par la variance de la différence entre valeurs des points et valeur identifiée. Si $x(t)$ sont les points générés et $y(t)$ les points identifiés, alors l'erreur pour chaque t est simplement donnée par $e(t) = x(t) - y(t)$.

On rappelle que la variance est donnée par $\sigma = \frac{1}{N} \sum_{t=1}^N (e(t) - \mu)^2$, où $\mu = \frac{1}{N} \sum_{t=1}^N e(t)$

est la moyenne de l'erreur.

Faites une fonction estimant la variance de l'erreur d'identification dont le prototype permettra de l'appeler par :

```
>> sigma = ErreurIdentification(x, L) ;
```

où `x` est le vecteur des points bruités et `L` le vecteur des trois paramètres de la parabole.

2.5 • Animation de l'histogramme de l'erreur d'identification.

Votre travail doit s'appuyer sur tout ce que vous avez fait jusqu'à présent.

On souhaite illustrer graphiquement la stabilité de la densité de probabilité de l'erreur d'identification dans la procédure moindres-carrés. Pour ce faire, on vous demande de créer un programme qui va faire 200 fois :

- simuler **150** points de la fonction $\sin(2\pi t)$ pour $t \in [0;0,3]$.
 - bruitez cette simulation avec un bruit Gaussien d'écart type 0,1
 - identifier la parabole passant au plus près au sens des moindres carrés de cet ensemble de points.
 - calculer la variance d'identification
 - afficher l'histogramme de la densité d'erreur d'identification, l'histogramme étant calculé entre -3 et +3 sur 20 cellules (c'est à dire qu'on doit voir apparaître 20 bâtons).
- En fin de programme, vous devez afficher l'évolution de la variance au cours de ces 200 expérimentations.

2.6 • Affichage 3D.

Modifiez le programme précédent de façon à stocker les 200 histogrammes dans un tableau de dimension 20×200 . Affichez ce tableau en 3D en bas à droite de l'écran.