MA 101 - Introduction à la Simulation Numérique

SVD - Décomposition en valeurs singulières

Denis Matignon denis.matignon@isae.fr
Emmanuel Zenou emmanuel.zenou@isae.fr

Résumé

Ce document présente les trois exercices d'application du cours sur la pseudo-inverse : la régression polynomiale, le débruitage de signal par synthèse de Fourier et la reconstitution de signal.

Table des matières

1 Applications

Nous allons voir dans cette partie les trois applications simples suivantes :

1. la régression de points, appliquée à la modélisation polynomiale de données SCAO ;
2. le débruitage de signal par synthèse de Fourier ;
3. et la reconstitution de signal par interpolation polynomiale.

2 Régression sur des données SCAO

Le fichier `data.txt` [1] est un fichier ASCII contenant des données du système de contrôle d'attitude et d'orbite (SCAO) d'un satellite d'observation en phase d'acquisition. Les données se présentent sous forme d'un tableau de 4 colonnes :

- colonne 1 : vecteur d'index ;
- colonne 2 : temps de mesure ;
- colonne 3 : mesure de roulis ;
- colonne 4 : mesure de tangage ;

Question 1 *Charger le fichier* `data.txt` *avec la commande* `load`*.*

Question 2 *Visualiser dans un même graphe les mesures de roulis et de tangage en fonction du temps.*

2.1 Régression linéaire

Soit $t = (t_1 \;\; t_2 \;\; \ldots \;\; t_N)$ le temps et $y = (y_1 \;\; y_2 \;\; \ldots \;\; y_N)$ une fonction à approximer. Le but est de trouver l'équation d'une droite qui minimise l'erreur quadratique. Supposons dans un premier temps une droite $\mathcal{D}_0 : y = p_1 \times t + p_0$. Chaque point doit dont vérifier, dans la mesure du possible,

$$y_i = p_1 \times t_i + p_0 \tag{1}$$

soit, pour N points,

$$\begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_N \end{pmatrix} = \begin{pmatrix} t_1 & 1 \\ t_2 & 1 \\ \ldots & \\ t_N & 1 \end{pmatrix} \times \begin{pmatrix} p_1 \\ p_0 \end{pmatrix} \tag{2}$$

soit, sous forme matricielle,

$$y = T \times p \tag{3}$$

On cherche ainsi à minimiser l'erreur quadratique du terme :

$$||y - T \times p||_2 \tag{4}$$

La matrice optimale p^* s'obtient grâce à la pseudo-inverse de T (instruction `pinv` sous MatLab) :

1. disponible à cette adresse :
`http://personnel.isae.fr/emmanuel-zenou/supaero/1ere-annee-l3/article/initiation-matlab.html`.

$$p^* = T^\dagger \times y \tag{5}$$

Question 3 *Interpréter le cas avec seulement deux points donnés.*

Question 4 *Approximer le roulis et le tangage avec une droite quelconque.*

2.2 Régression polynomiale à l'ordre n

On se propose de construire maintenant le polynôme de degré n passant *au mieux* par les N points de mesure en roulis et en tangage. Chaque point devra donc vérifier, au mieux,

$$y_i = \sum_{j=0}^{n} p_j t_i^j \tag{6}$$

Question 5 *Écrire le nouveau système d'équations sous forme développée puis matricielle.*

Question 6 *Écrire la fonction* `Regression` *qui calcule les coefficients du polynôme de degré n pour des mesures prises aux temps t.*

Question 7 *Faire une régression à l'ordre deux, trois et cinq des données roulis et tangage.*

3 Débruitage d'un signal par synthèse de Fourier

Considérons le signal réel suivant :

$$y(t) = Kt + e^{cos(\pi t)} sin\big(5 sin(2\pi t)\big) + B(t),\ t = [0 \dots 10] \tag{7}$$

que l'on peut interpréter comme un signal périodique avec une tendance linéaire en temps, et bruité. Le but est ici de reconstituer le signal (éventuellement bruité) à partir de ses coefficients de Fourier.

Considérons dans un premier temps uniquement le signal périodique. La synthèse donne par définition

$$\begin{aligned} y(t) &= \sum_{n=-\infty}^{+\infty} c_n e^{int\frac{2\pi}{T}} && (8)\\ &= \sum_{n=0}^{+\infty} a_n \cos(nt\frac{2\pi}{T}) + b_n \sin(nt\frac{2\pi}{T}) && (9) \end{aligned}$$

Dans la pratique on s'arrête à l'ordre M (variable) :

$$y(t) = \sum_{n=0}^{M} a_n \cos(nt\frac{2\pi}{T}) + b_n \sin(nt\frac{2\pi}{T}) \tag{10}$$

Question 8 *Écrire l'ensemble des équations sous forme matricielle.*

Pour voir la puissance de cette méthode, on cherchera à identifier le signal initial (sans bruit) à partir du même signal bruité (on supposera le bruit additif gaussien). Le programme sera le suivant :

```
clear all, close all, clc ;

%% Paramètres
sig=1;                          % Niveau de bruit
K=0;                            % Tendance linéaire
M=10 ;                         % Nb de coeff Fourier

t=0:0.01:10;

%% Construction du signal :
y= exp(cos(pi*t)).*sin(5*sin(2*pi*t)) + K*t;
yb = y + sig*randn(size(y)) ;

figure, plot(t,y,t,yb) ;
legend('Signal initial','Signal bruité') ;
title('Signal initial') ;
```

Question 9 *Identifier les coefficients de Fourier du signal bruité. Construire la synthèse et la comparer au signal initial.*

Question 10 *Que se passe-t-il si l'on augmente l'ordre de la synthèse (par exemple* $M = 100$*) ? Est-ce intéressant pour cette application ? Quel serait a priori la valeur optimale de* M *?*

Question 11 *Quels sont selon vous les limites de cette approche ?*

Supposons maintenant qu'une tendance linéaire apparaisse dans le signal.

Question 12 *Adapter la méthode pour reconstruire le signal et mesurer le paramètre* K*.*

4 Reconstitution de signal (facultatif)

Dans cette partie, nous allons aborder les notions importantes de discrétisation et d'interpolation, au travers d'une étude du langage animal.

4.1 Le signal acoustique

L'étude acoustique du signal a permis une modélisation au travers d'un algorithme stochastique, décrit dans `Chant.m`, qui vous permet de simuler le signal acoustique.

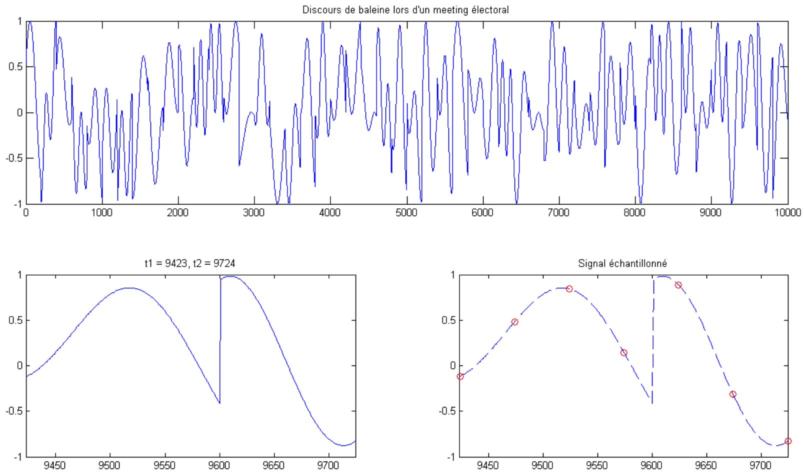

Figure 1 – Signal initial, fenêtre temporelle et échantillonnage.

4.2 Enregistrement du signal acoustique

Sur le terrain, le signal est enregistré avec une période d'échantillonnage T_e. Le signal échantillonné est donc une mesure partielle du son réel. L'échelle de temps est la ms : tous les vecteurs temps sont indicés par un temps en ms. Ceci est illustré figure 1.

Ainsi est enregistrée, sur support électronique, la valeur de $s(t)$ toutes les T_e ms. Nous allons simuler ceci, en ne prenant qu'une valeur de $s(t)$ sur 50 de manière régulière. Nous allons faire cette étude sur une fenêtre temporelle de taille $T = 300ms$.

Question 13 *En étudiant la fonction* `Chant.m`*, créer un petit programme (macro)*

1. *qui lance une simulation, avec un pas minimal de* $1ms$*, 50 phases minimales et un index des fréquences maxi de 30 ;*
2. *qui extrait une fenêtre temporelle au hasard de taille* $T = 300$*,*
3. *qui extrait une valeur sur* 50 *du signal initial, simulant ainsi un échantillonnage du signal,*
4. *et qui affiche l'ensemble des données avec plein de titres et de légendes.*

Deux niveaux de numérisation du signal sont mises en œuvre :

1. En enregistrant, pour chaque échantillon, la valeur du signal,
2. et en enregistrant, toujours pour chaque échantillon, la valeur du signal et de sa dérivée.

Nous allons voir ici deux méthodes de reconstitution du signal temporel à partir des échantillons enregistrés : la première méthode consiste à faire une interpolation entre les échantillons sur une fenêtre entière ; la seconde méthode consiste à utiliser une fenêtre glissante. Chaque méthode se décline selon deux approches : la première approche consiste à n'utiliser que la valeur des échantillons (premier ordre) ; la seconde approche consiste à utiliser la valeur des échantillons et leur dérivée (deuxième ordre).

4.3 Interpolation par fenêtre

∗ Premier ordre

Supposons, dans la fenêtre que vous venez de sélectionner, l'ensemble des N échantillons enregistrés. Pour reconstituer le signal avec la première méthode de numérisation, il suffit de trouver un polynôme de degré $n = N - 1$ passant par ces N points.

Question 14 *Poser le système d'équation correspondant sous forme matricielle, puis proposer une méthode de résolution simple par pseudo-inverse.*

Question 15 *Implémenter une fonction permettant de répondre au problème. La mettre en œuvre sur la fenêtre T.*

Question 16 *Proposer un outil permettant de mesurer la différence entre le signal réel et le signal approximé (en %).*

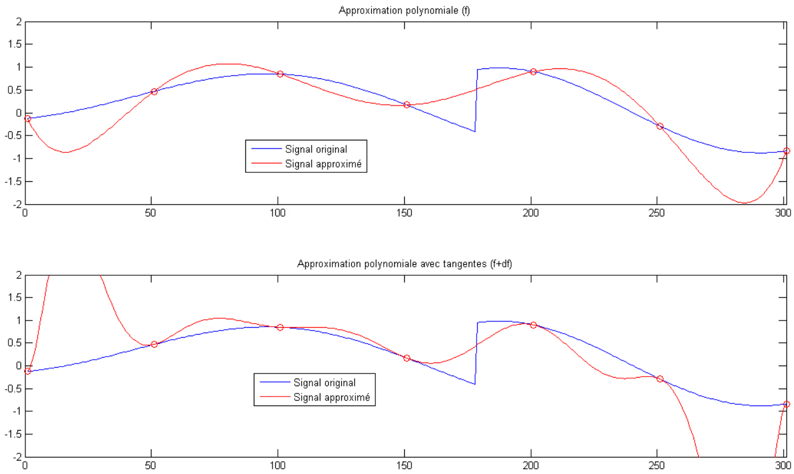

Figure 2 – Interpolation.

∗ Deuxième ordre

Ensuite, avec la seconde méthode de numérisation, il est possible de tenir compte de la dérivée de chaque échantillon. Le polynôme minimum pour obtenir un tel signal est de degré $n = 2 \times N - 1$.

Question 17 *Poser le système d'équation correspondant sous forme matricielle, puis proposer une méthode de résolution simple par pseudo-inverse.*

Question 18 *Implémenter une fonction permettant de répondre au problème. La mettre en œuvre sur la fenêtre T.*

Question 19 *Comparer le résultat avec précédemment. Commenter.*

4.4 Fenêtre glissante

Les fonctions B-splines sont très utilisées dans tous les problèmes d'interpolation et d'approximation aussi bien dans le plan que dans l'espace. Elles sont présentes à l'heure actuelle dans tous les systèmes de Conception ou de Dessin Assisté par Ordinateur.

Supposons deux échantillons aux instants t et $t + T_e$ du signal : $s(t)$ et $s(t + T_e)$. Il est possible de relier ces deux points, en fonction de la stratégie de numérisation choisie, avec une fonction

- linéaire pour respecter les valeurs du signal en chaque extrémité,
- de degré 3 pour respecter les valeurs et les dérivées.

Question 20 *Poser les équations permettant de relier linéairement deux échantillons consécutifs et proposer une méthode de résolution simple par pseudo-inverse.*

Question 21 *Poser les équations permettant de relier, en tenant compte de la dérivée de chacun d'entre eux, deux échantillons consécutifs et proposer une méthode de résolution simple par pseudo-inverse.*

Question 22 *Implémenter et mettre en œuvre ces méthodes sur la fenêtre T.*

Question 23 *Comparer le résultat avec précédemment.*

Question 24 *Comparer les différentes méthodes sur le signal complet. Conclure.*