

Compte rendu de TP: Formation de voies

Thomas Lechat & Dinseny Alice

26 novembre 2015

1 Introduction

La formation de voies est une méthode d'imagerie acoustique permettant d'obtenir la contribution de sources dans un plan à l'aide d'une antenne de microphones.

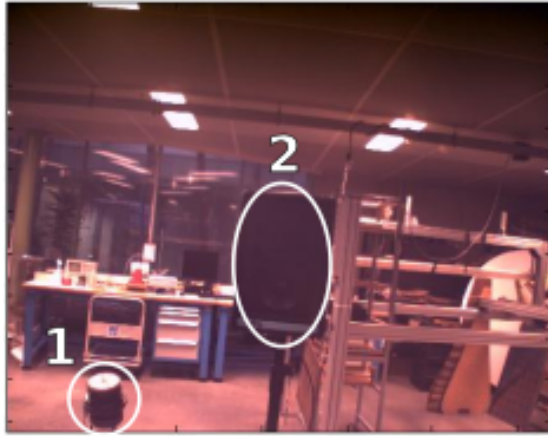
Le principe général est de construire un vecteur de pointage qui pondère les signaux microphoniques en fonction du point d'observation sur le plan source.

Ce rapport vise à comparer trois méthodes de formation de voies testées sur des signaux microphoniques obtenus par des mesures de sources connues.

2 Obtention des données de test

Des mesures de champs acoustiques sont effectuées à l'aide d'une antenne constituée de ?????? microphones disposés en spirale.

Les sources sont placées dans un plan situé à 1,43 m de l'axe de l'antenne. Deux configurations sont testées et présentées en figure 1.



(a) Configuration 1 : le haut-parleur 1 émet un sinus pur à 600 Hz et l'enceinte 2 émet un bruit blanc.



(b) Configuration 2 : le haut-parleur 1 émet un sinus pur à 800 Hz et l'enceinte 2 émet un bruit blanc.

FIGURE 1 – Configurations du plan source.

La matrice interspectrale des microphones appelée G_{pp} peut ainsi être obtenue à l'aide du logiciel Signal Express et du boîtier d'acquisition National Instrument.

3 Méthode de Bartlett

La méthode de Bartlett consiste à déterminer le vecteur de pointage w_i tel que l'amplitude estimée des sources S_i s'écrive :

$$S_i = w_i^H p,$$

avec p les signaux de pression mesurés par les microphones.

Le vecteur \mathbf{w}_i est calculé en minimisant la fonction coût $|\mathbf{w}_i^H \mathbf{p} - A_i|$ où A_i sont les amplitudes réelles des sources.

On trouve alors

$$\mathbf{w}_i = \frac{\mathbf{h}_i}{\mathbf{h}_i^H \mathbf{h}_i}$$

avec \mathbf{h}_i la contribution de la source i , composée des fonctions de transfert entre chaque microphone m et cette source. Ces fonctions de transfert sont ici celles d'un rayonnement en espace libre sur une distance r_{mi} :

$$h_{mi} = \frac{e^{-jkr_{mi}}}{4\pi r_{mi}}.$$

4 Méthode de Capon

Dans la méthode de Capon, seule la définition du vecteur de pointage change : on cherche à minimiser $\mathbf{w}_i^H \mathbf{G}_{pp} \mathbf{w}_i$ avec la contrainte de normalisation $\mathbf{w}_i^H \mathbf{h}_i = 1$.

En résolvant ce problème avec la méthode des multiplicateurs de Lagrange, on trouve le nouveau vecteur de pointage suivant :

$$\mathbf{w}_i = \frac{\mathbf{G}_{pp}^{-1} \mathbf{h}_i}{\mathbf{h}_i^H \mathbf{G}_{pp}^{-1} \mathbf{h}_i}.$$

Cette méthode est supposée donner de meilleurs résultats que la méthode précédente, mais comporte la contrainte de l'inversion de la matrice \mathbf{G}_{pp} .

5 Méthode MUSIC

Cette méthode est basée sur la décomposition en valeurs propres de la matrice interspectrale \mathbf{G}_{pp} . Cette matrice est ensuite décomposée en un sous-espace signal et un sous-espace bruit. Le sous-espace bruit correspond aux plus petites valeurs propres de \mathbf{G}_{pp} et est utilisé pour estimer la présence P_i d'une source au point i :

$$P_i = \frac{1}{\sum_{q=r+1}^M \frac{|\mathbf{h}_i^H \mathbf{u}_q|^2}{\sigma_p^2}} \quad (1)$$

où σ_q sont les valeurs propres de l'espace bruit et M est le nombre de champs cohérents orthogonaux qui composent \mathbf{G}_{pp} .

```

1  % bf_traitement
2  %
3  % TP Formation de voies
4  % Jean-Claude Pascal et Jean-Hugh Thomas
5  %
6  close all; clear all;
7  %-- determination de la methode : 'bartlet','capon','music'
8  bfmethod = 'music';
9
10 disp(' '), disp(['-- Traitement avec la methode ',bfmethod]), disp(' ')
11
12 %-- construction du path
13 rootpath = cd;
14 addpath([rootpath '/Bf_bib']),
15
16 %=====
17 % Lecture de la matrice interspectrale (code complet)
18 %-----
19
20 %-- Selection du fichier hdf
21 %
22 FilterSpec = {'*.hdf','hdf file'};
23 [HDFFileName,HDFPathName] = uigetfile(FilterSpec,'load an hdf spectral file');
24
25 if ischar(HDFFileName)
26     filename = [HDFPathName HDFFileName];
27 else
28     return,
29 end
30
31 %-- Visualisation du spectre moyen sur les microphones de l'antenne
32 %
33 [avspect,freqvect,axename] = HDFInterface('averageddata',filename);
34 HDFInterface('averageddata',filename);
35
36 %-- Selection de la frequence de traitement
37 %
38 disp(' '), disp('Selection de la frequence'),
39 freq = input('        entrer la frequence a traiter (en Hz) : ');
40 [freq,ifreq] = nearest(freqvect,freq);
41
42 %-- Chargement de la matrice interspectrale
43 %
44 [Refarray,axevector,axeorder] = HDFfileAScontrol('getdata',filename,'Refarray',ifreq,':',1);
45 Gpp = TransRefarray('vec2mat',Refarray,'single');
46 Gpp = conj(Gpp);
47
48 % INFO : Gpp = Gpp'; dans Matlab Gpp' represente la transposee hermitienne de la matrice complexe Gpp
49 %        la matrice Gpp est une matrice carree [M M] (M nombre de microphones)
50
51 %=====
52 % Lecture des coordonnees des points de l'antenne (code complet)
53 %-----
54 %
55 % INFO
56 % micropnts est une matrice [M 3] ou M est le nombre de microphones de l'antenne
57 % micropnts(:,1) est le vecteur colonne des coordonnees x

```

```

58 % micropnts(:,2) est le vecteur colonne des coordonnees y
59 % micropnts(:,3) est le vecteur colonne des coordonnees z (normalement nul car le
60 % plan de l'antenne est en z = 0)
61
62 [micropnts,coordsys,arraysys] = HDFInterface('micropnts',filename);
63
64 %=====
65 % Determination du plan de representation (code a completer : donner des valeurs)
66 %-----
67
68 % INFO
69 % Le plan de representation est parallele a celui de l'antenne Pour definir les points
70 % ou seront estimees les sources il faut fournir les informations suivantes :
71 % Nx, Ny -> le nombre de points en x et y
72 % dist -> la distance du plan de representation a celui de l'antenne
73 % Xmin Xmax Ymin Ymax -> les limites du plan de representation
74 % L'origine du repere est situee sur l'axe de l'antenne. Par exemple :
75 Nx = 40;
76 Ny = 40;
77 dist = 1.5;
78 Xmin = -1;
79 Xmax = 1;
80 Ymin = -1;
81 Ymax = 1;
82 M=size(micropnts);
83 M=M(1);
84
85 %-- construction du maillage sur le plan de representation
86 %
87 x = linspace(Xmin,Xmax,Nx);
88 y = linspace(Ymin,Ymax,Ny);
89 [Xmat,Ymat] = meshgrid(x,y);
90
91 %-- vecteur [Np 3] des positions des sources
92 Np = Nx*Ny;
93 srcpnts = [Xmat(:) Ymat(:) -dist*ones(Np,1)];
94
95 %=====
96 % Pre-traitement selon la methode choisie (code a completer)
97 %-----
98 if strcmpi(bfmethod,'capon')
99     disp(' :: pre-traitement pour la methode de Capon'),
100     % INFO
101     % le traitement consiste ici a inverser la matrice Gpp
102
103     Gpp_inv=Gpp^-1;
104
105 elseif strcmpi(bfmethod,'music')
106     disp(' :: pre-traitement pour la methode de MUSIC'),
107     % INFO
108     % le traitement consiste ici a decomposer la matrice Gpp en utilisant la fonction svd
109     % de Matlab [U,S,V] = svd(Gpp) (dans ce cas particulier V = U')
110
111     [u,s,v] = svd(Gpp);
112
113
114 end
115
116 %=====
117 % Boucle de traitement pour chacun des points sources (code a completer)
118 %-----
119 hw = waitbar(0,['traitement methode ',bfmethod,' ...']);
120 S = zeros(Np,1);

```

```

121
122 for ii=1:Np
123
124     waitbar(ii/Np,hw);
125
126     %-- vecteur [1 3] des coordonnees du point source
127     coorsrc = srcpnts(ii,:);
128
129     %-----
130     % Calcul des distances
131     % calcul du vecteur R [M,1] des distance entre chaque microphone et le point
132     % source
133     %-----
134
135     for m=1:M
136         R(m)=norm(coorsrc-micropnts(m,:));
137     end
138
139
140     %-----
141     % Calcul du vecteur h [M,1] representant les fonctions de transfert entre
142     % les microphones et le point source (Eqs. 2.2 et 2.3)
143     %-----
144     c=343; %celerite du son dans l'air en m/s
145     k=2*pi*freq/c;
146
147     h=exp(-j*k*R)./(4*pi*R);
148     h=h';
149
150
151     %-----
152     % Calcul du vecteur de pilotage selon la methode
153     % La methode MUSIC n'est pas concernee par cette phase
154     %-----
155
156     if strcmpi(bfmethod,'bartlet')
157         % INFO
158         % voir Eq. 3.7
159
160         w=(h'*h)^(-1)*h;
161
162     elseif strcmpi(bfmethod,'capon')
163         % voir Eq. 5.5
164
165         w=Gpp_inv*h/(h'*Gpp_inv*h);
166
167     end
168
169
170     %-----
171     % Calcul de la distribution des sources selon les methodes
172     % les resultats du calcul sont ranges dans un vecteur S [Np 1]
173     %-----
174
175     if strcmpi(bfmethod,'bartlet')
176         % INFO
177         % voir Eq. 2.5
178
179         S(ii)=w'*Gpp*w; %dsp
180
181
182     elseif strcmpi(bfmethod,'capon')
183         % voir Eq. 2.5

```

```

184
185         S(ii)=w'*Gpp*w; %dsp
186
187
188     elseif strcmpi(bfmethod,'music')
189         % voir Eq. 6.3
190         somme=0;
191         q0=15; %taille de l'espace signal
192         for q=q0:M
193             somme= somme+ (abs(h'*u(:,q)))^2;
194         end
195         S(ii)=1/somme; %Pi
196
197     end
198
199 %-----
200 end % fin de la boucle ii=1:Np
201
202 close(hw),
203 %-----
204
205
206 %=====
207 % Reconstitution de la matrice rectangulaire et visualisation (code a completer)
208 %-----
209 %
210 % INFO
211 % selon meshgrid S doit avoir comme dimensions [Ny Nx]
212 S = reshape(S,Ny,Nx);
213 S = real(S);
214
215 %-- utiliser ici eventuellement une interpolation pour avoir un maillage de
216 % representation plus fin (fonction interp2 de Matlab)
217 x1 = x;
218 y1 = y;
219 S1 = S;
220
221 % A COMPLETER EVENTUELLEMENT
222
223 %-- visualiser la carte des sources en utilisant la fonction imagesc
224 %
225
226 titre = ['Methode ',bfmethod,' - f = ',num2str(freq),' Hz'];
227
228
229
230
231
232 reptype = 'lin'; % 'lin' ou 'dB'
233
234 if strcmpi(reptype,'lin')
235
236     %-- pour une representation lineaire
237     repstruct.mode = 'mod*';
238     repstruct.dynscal = []; % -> range of representation of scalar map (used for dB)
239     repstruct.maxscal = []; % -> max value of scalar map ( [] -> automatic scaling)
240     repstruct.stepscale = 10; % -> step for rounded max value in automatic scaling
241     repstruct.dBref = 1; % -> energy reference for dB (A = 10 log [real(Z)/dBref])
242     repstruct.title = titre; % -> map title string
243     repstruct.underrange_color = [0.9 0.9 0.9]; % -> underrange color
244     repstruct.unit = ''; % -> string of quantity unit to put under the colorbar
245
246 else

```

```
247
248 %-- pour une representation en dB
249 Dyn = 15;
250 RefdB = 1e-12;
251 repstruct.mode = 'dB*';
252 repstruct.dynscal = Dyn;    % -> range of representation of scalar map (used for dB)
253 %repstruct.maxscal = dBmax; % -> max value of scalar map ( [] -> automatic scaling)
254 repstruct.maxscal = [];    % -> max value of scalar map ( [] -> automatic scaling)
255 repstruct.stepscal = 1;    % -> step for rounded max value in automatic scaling
256 repstruct.dBref = RefdB;   % -> energy reference for dB ( $A = 10 \log [\text{real}(Z)/\text{dBref}]$ )
257 repstruct.title = titre;   % -> map title string
258 repstruct.underrangecolor = [0.95 0.95 0.95]; % -> underrange color
259 repstruct.unit = 'dB';     % -> string of quantity unit to put under the colorbar
260
261 end
262
263 figure
264 %imagesc(x1,y1,S1)
265 ccmmap(repstruct,x1,y1,S1);
```
