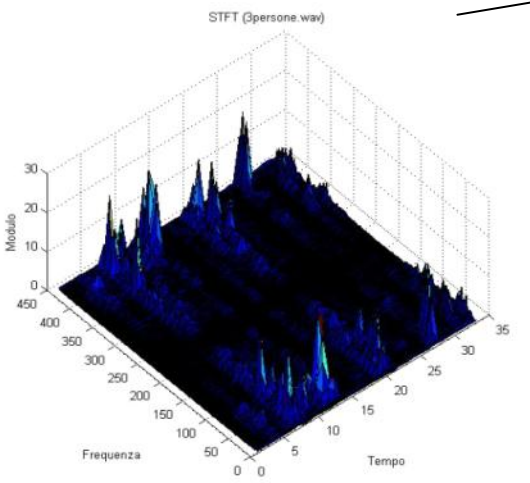
Algoritmi

lunedì 23 marzo 2009
13.46

Codice Antonacci ULA.m

Chiaramente per la simulzione devono essere creati anche due file diversamente ritardati per ogni sorgente per simulare le ricezioni diverse dei due distinti microfoni

STFT



Come descritto nell'articolo 1

-->STFT , size della finestra T=512 con fs=8kHz ,overlap di T/4 (nell'articolo ...capire se va bene T/2) -->T/2 !!!
-->sorgenti sonore di durata circa 3 sec
-->valutazioni fatte con SIR e SDR (vedere l'appendice del primo articolo!) ???

Ritardi al secondo microfono:

1° sorg --> 160 (20 ms)
2° sorg --> 560 (70 ms) ***campioni***
3° sorg--> 800 (100 ms)

Sostituire il numero dei cicli parametrizzandolo, Usare la formula calcolata:

numero_splice=1+((length(signal)-length(win))/(length(win)/2))

Seganli generati per i microfoni:

CODICE:

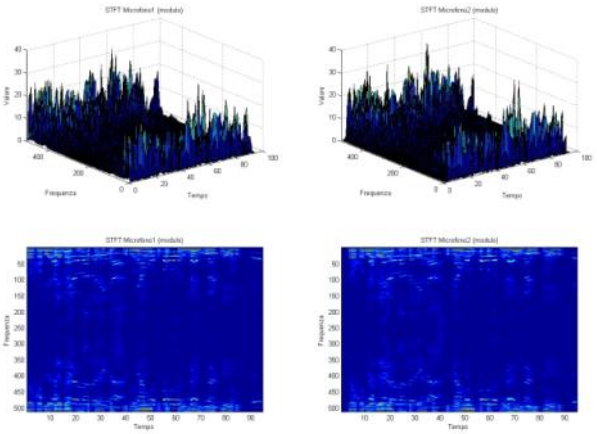
```
%creo il segnale totale al primo e al secondo microfono
mic1=signal1+signal2+signal3;
mic2=signal1r+signal2r+signal3r;

%ora devo creare i grafici SDTF ottenuti dai segnali dei due
microfoni

%grafico microfono 1
mic1part(1:512,1)=mic1(1:length(win),1);
mic1splice(1:512,1)=mic1part(1:512,1)'.*win';

%andrà poi inserito il controllo per usare tutti i campioni dei
%segnali....con uteriore padding

cicli=fix((length(mic1)-length(win))/(length(win)/2)); %uso fix per
prendere l'intero inferiore
```

```
%FILE DEL PROGETTO DI TATA
%prova stft e risintesi di un segnale audio
%
% CALCOLO DEL NUMERO DI SPLICE:
% numero_splice=1+((length(signal)-length(win))/(length(win)/2))
%


clear all;
close all;

%creazione finestra
win=hamming(512);

%definita anche la frequenza di campionamnto per come viene
importato il file audio

[signal1,Fs1]=wavexread('3personePCM.wav');
[signal2,Fs2]=wavexread('Toms_diner.wav');
[signal3,Fs3]=wavexread('voce_maschile.wav');

lunghezza=25000;

if length(signal1)>lunghezza | | length(signal2)>lunghezza | |
length(signal3)>lunghezza;
   lunghezza=max(length(signal1),length(signal2));
   lunghezza=max(lunghezza,length(signal3));
end;

signal1=cat(1,signal1,zeros(lunghezza-length(signal1),1));
signal2=cat(1,signal2,zeros(lunghezza-length(signal2),1));
signal3=cat(1,signal3,zeros(lunghezza-length(signal3),1));

%eseguo la finestratura dei seganli, lo splicing
part1(1:512,1)=signal1(1:length(win),1);
part2(1:512,1)=signal2(1:length(win),1);
part3(1:512,1)=signal3(1:length(win),1);
splice1(1:512,1)=part1(1:512,1)'.*win';
splice2(1:512,1)=part2(1:512,1)'.*win';
splice3(1:512,1)=part3(1:512,1)'.*win';

for i=2:95;

   start=1+(i-1)*(256);
   fin=start+length(win)-1;
   part1(1:512,i)=signal1(start:fin,1);
   part2(1:512,i)=signal2(start:fin,1);
   part3(1:512,i)=signal3(start:fin,1);
   splice1(1:512,i)=part1(1:512,i)'.*win';
   splice2(1:512,i)=part2(1:512,i)'.*win';
   splice3(1:512,i)=part3(1:512,i)'.*win';

end;

%memorizzo le trasformate

for i=1:95;

   fourier1(1:512,i)=fft(splice1(1:512,i));
   fourier2(1:512,i)=fft(splice2(1:512,i));
   fourier3(1:512,i)=fft(splice3(1:512,i));

end;

%moduli per poterli rappresentare
mod_stft1=abs(fourier1);
mod_stft2=abs(fourier2);
mod_stft3=abs(fourier3);
figure(1); surf(mod_stft1);
figure(2); surf(mod_stft2);
figure(3); surf(mod_stft3);


%segnale risintettizzato (utilizzando solamente le stft calcolate)
```

```
for i=2:cicli;                                                    %       [...]

   start=1+(i-1)*(256);
   fin=start+length(win)-1;
   mic1part(1:512,i)=mic1(start:fin,1);
   mic1splice(1:512,i)=mic1part(1:512,i)'.*win';

end;
```

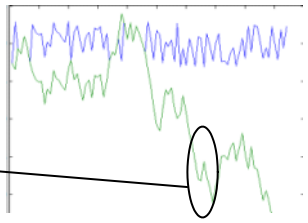
(***stessi cilci per il microfono 2***)

```
%diagrammi STFT
for i=1:cicli;

   mic1FFT(1:512,i)=fft(mic1splice(1:512,i));
   mic2FFT(1:512,i)=fft(mic2splice(1:512,i));

end;

%calcolo i moduli per rappresentarli (solo per chiarezza e
controllo)
mod_mic1=abs(mic1FFT);
mod_mic2=abs(mic2FFT);
figure(1);
subplot(2,2,1); surf(mod_mic1);
subplot(2,2,2); surf(mod_mic2);
subplot(2,2,3); imagesc(mod_mic1);
subplot(2,2,4); imagesc(mod_mic2);
```



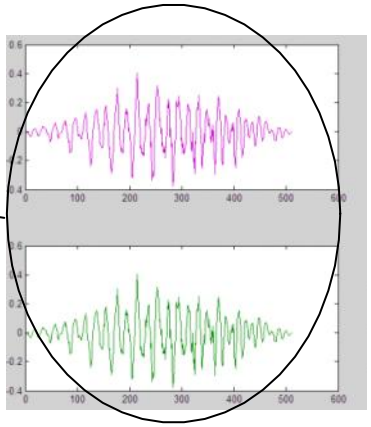
IFFT + PhaseUnwrapping --> vedere codici di Antonacci

Controllare bene il funzionamento della funzione MATLAB "unwrap", sembra funzionare al contrario!!!???



Curva della fase dopo l'applicazione della funzione, l'altra è l'originale

1° PROVA --> provo a ricostruire senza eseguire il phase unwrapping!!!



Sono due tratti del segnale ricavati, il primo dallo splicing (semplice finestratura) ed il secondo dalla trasformata di fourier inversa di una colonna della matrica STFT.

SONO UGUALI --> in effetti non c'è ragione per eseguire il phaseunwrapping se non vengono effettuate particolari elaborazioni sugli spettri...

OK!!! --> la ricostruzione OVERLA-&-ADD funziona anche senza phase unwrapping nel caso in cui non si eseguano elaborazioni particolari sugli spettri!!!!

--> Bisognerà capire se l'applicazione delle maschere binarie è una di queste elaborazioni per cui poi si richiede l'applicazione di algoritmi di phase unwrapping!!

File sonori originali e rigenerati dopo trasformazione inversa di fourier

Vedi

Mic1.wav -->originale
Mic1ric.wav -->ricostruito

Mic2.wav
Mic2ric.wav

CLUSTERING

APPLICAZIONE MASCHERE BINARIE