APPENDIX D

Cmatrix.m

```
%
% FUNCTION Cmatrix
%
% INPUTS:
% ======
%
% s is the signal to analyze
% fs is the sampling rate of the signal
%
% OUTPUTS:
% ========
%
% r is the transformed signal
%

function r = Cmatrix(s, fs)
  l = length(s);
  m = 100;
  n = 256;

  nbFrame = floor((l - n) / m ) + 1;

%
% Create a matrix M containg all the frames
%
  disp('CREATE MATRIX CONTAINING ALL THE FRAMES...');
  for i = 1:n
    for j = 1:nbFrame
       M(i, j) = s(((j - 1) * m) + i);
    end
 end

%
% Matrix M created. Now apply HAMMING window and store in matrix N. Column
vectors of N are
% the original frame vectors transformed by the Hamming window filter
%
  disp('APPLY THE HAMMING WINDOW...');
  h = hamming(n);
  N = diag(h) * M;

%
% Now apply FFT and create a new matrix M2 where the column vectors are the
FFTs of the
% column vectors of N. The elements of column matrix M2 contain the frames
of the original
```

```
% signal, filtered by the Hamming window and transformed with the FFT. The
elements of M2
% are complex numbers and symmetrical because FFT was used to transform the
data.
%
% Each column in M2 is a power spectrum of the original signal
%
  disp('APPLY FFT...');
  for i = 1:nbFrame
      M2(:,i) = fft(N(:, i));
  end

 t = n / 2;
 tmax = 1 / fs;

%
% Determine mel-spaced filterbank
%
 disp('DETERMINE MEL-SPACED FILTERBANK COEFFICIENTS...');
 m = mel(20, n, fs);
 n2 = 1 + floor(n / 2);
 z = m * abs(M2(1:n2, :)).^2;

 r = dct(log(z));
%
% END OF FUNCTION Cmatrix
%
```