Signal Processing - Report

Lab 1 – Group 2

OLIVIER JAIS-NIELSEN

17 novembre 2008

Signal Processing - Report

Lab 1 – Group 2

Discrete Fourier Transform

1. Implementation of the DFT

```
function X=DiscreteFourier(x)

  N=length(x);
  W=exp(-2*pi*i/N);
```

N and W are defined as in the text.

```
  k=0:N-1;
  K=k'*k;
  Wk=W.^K;
```

We create the vector $k=\begin{pmatrix}0 & \dots & N-1\end{pmatrix}$ and the matrix $K=k^T \bullet k=(kn)_{0\le k,n\le N-1}$ and finally the matrix $W_k=\left(W^{nk}\right)_{0\le n,k\le N-1}$.

```
  X=Wk*x;

endfunction
```

Indeed, the DFT of x is :

$$X=W_k \bullet x=\left(\sum_{n=0}^{N-1} x(n)\bullet W^{nk}\right)_{0\le k\le N-1}$$

2. Implementation of the FFT

```
function X=FastFourier(x)
```

This algorithm is recursive.

```
  N=length(x);

  if N==1
    X=x;

  else
    W=exp(-2*pi*i/N);
    xOdd=x(1:2:end);
    xEven=x(2:2:end);
```

If the signal's length is a power of two greater than one, after having defined W as before, we split it in two parts : xOdd, the points with odd indexes and xEven, the others.

```
    T0=FastFourier(xOdd);
    T1=FastFourier(xEven);
```

T0 is the FFT of the first and T1 is the FFT of the second.

```
    k=0:(N/2)-1;
    Wk=W.^k';
    WkT1=Wk.*T1;
```

We define $k=\begin{pmatrix}0 & \dots & N-1\end{pmatrix}$,

$W_k=\begin{pmatrix}1 & W & \dots & W^{N-1}\end{pmatrix}$,

$W_kT_1=\left(W^k \bullet T_k^1\right)_{0\le k\le N-1}$.

```
    X1=T0+WkT1;
    X2=T0-WkT1;
    X=[X1;X2];
```

Finally, X1 represents the first half of the DFT and X2 the second half.

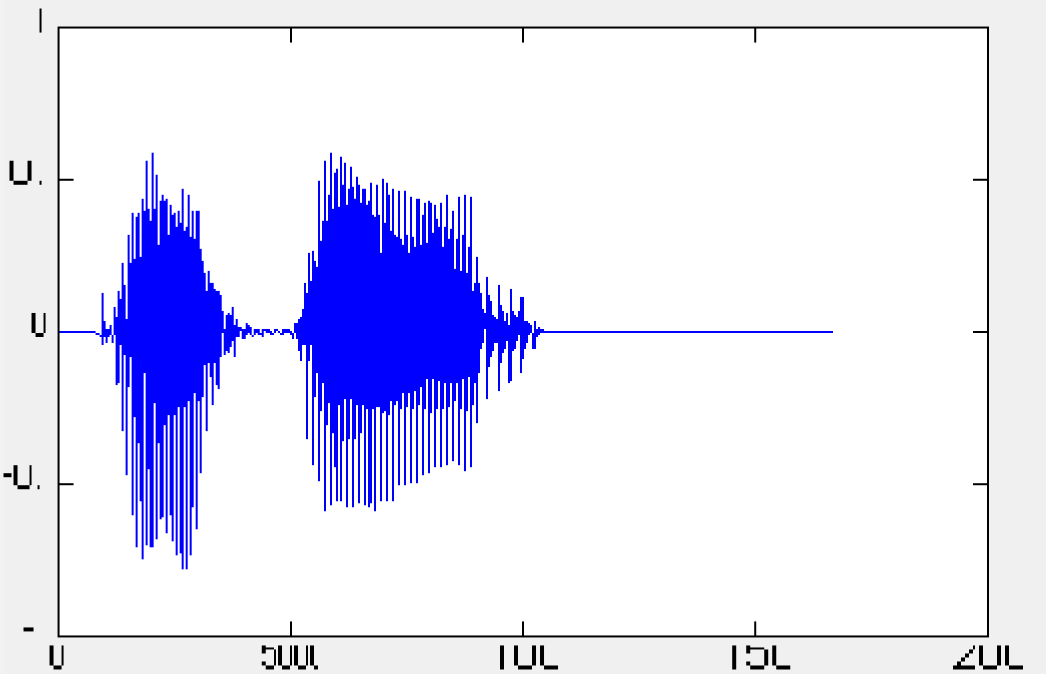
```
    end
endfunction
```

3. Manipulations on WAVE files

a. The “ball paint” sound

The goal is to time-shift the signal, in a circular way, so that the second part of the signal becomes the first part.

When we draw the signal we can easily see the value of the shift we will need :



The “ball” sound starts at the 5100th sample. Therefore, the time shift is $T = 5100$. This can be done by multiplying the k-ieth point of the DFT of the signal by $e^{2i\pi \bullet 5100 \bullet k/N}$ and applying the inverse DFT. Working in the Fourier space has the advantage to naturally achieve the circular shifting :

```
[y, FS, BITS] = wavread
("paint_ball.wav");

N=length(y);
```

We open the file.

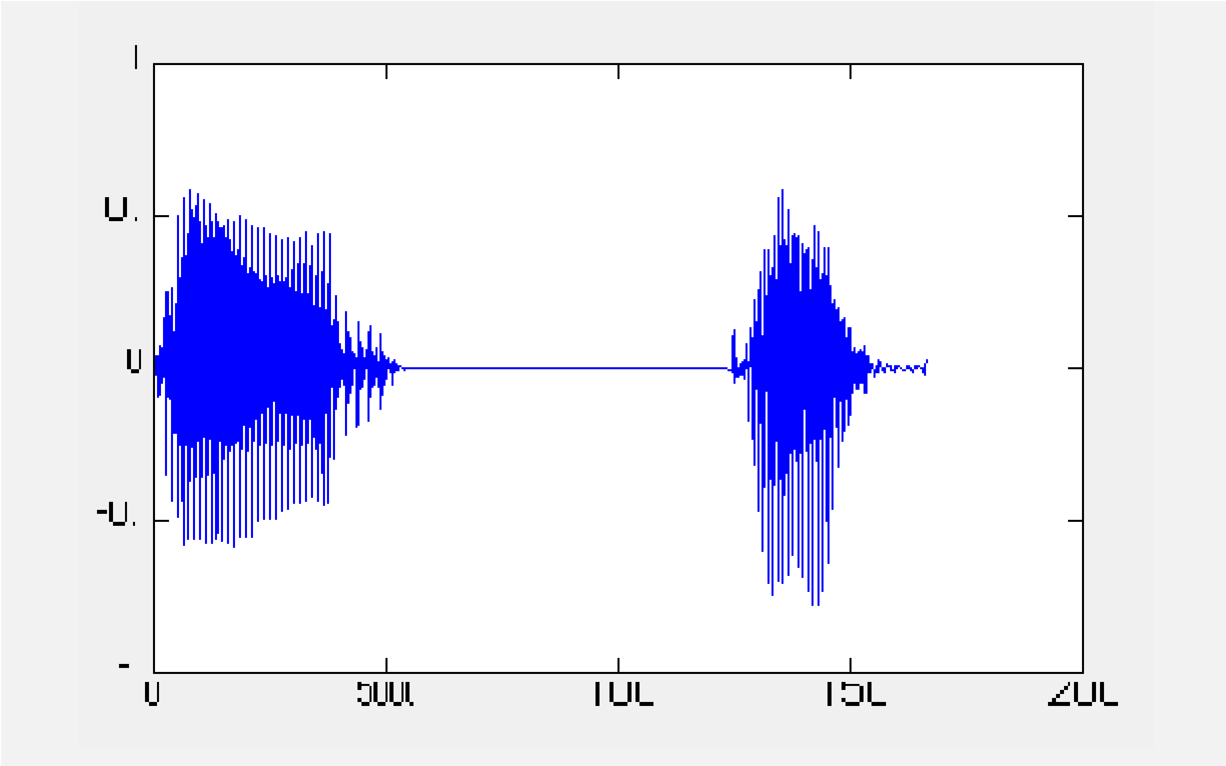
```
Y=fft(y);
k=0:N-1;
Coef=(exp(i*2*pi*5100/N)).^k';
YShift=Coef.*Y;
```

We work on the DFT of the signal.

We apply the coefficient $e^{2i\pi \bullet 5100 \bullet k/N}$ to each point of the DFT.

```
yShift=ifft(YShift);
wavwrite(yShift, FS, BITS,
"ball_paint.wav");
```

We save the final signal.

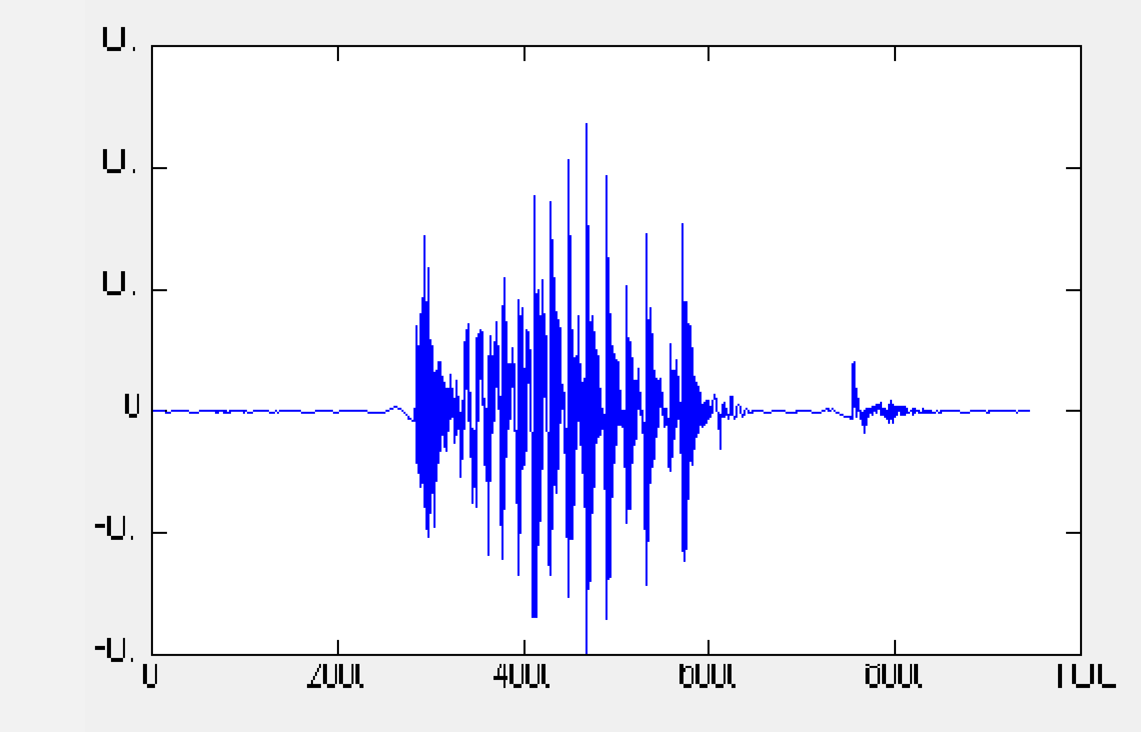


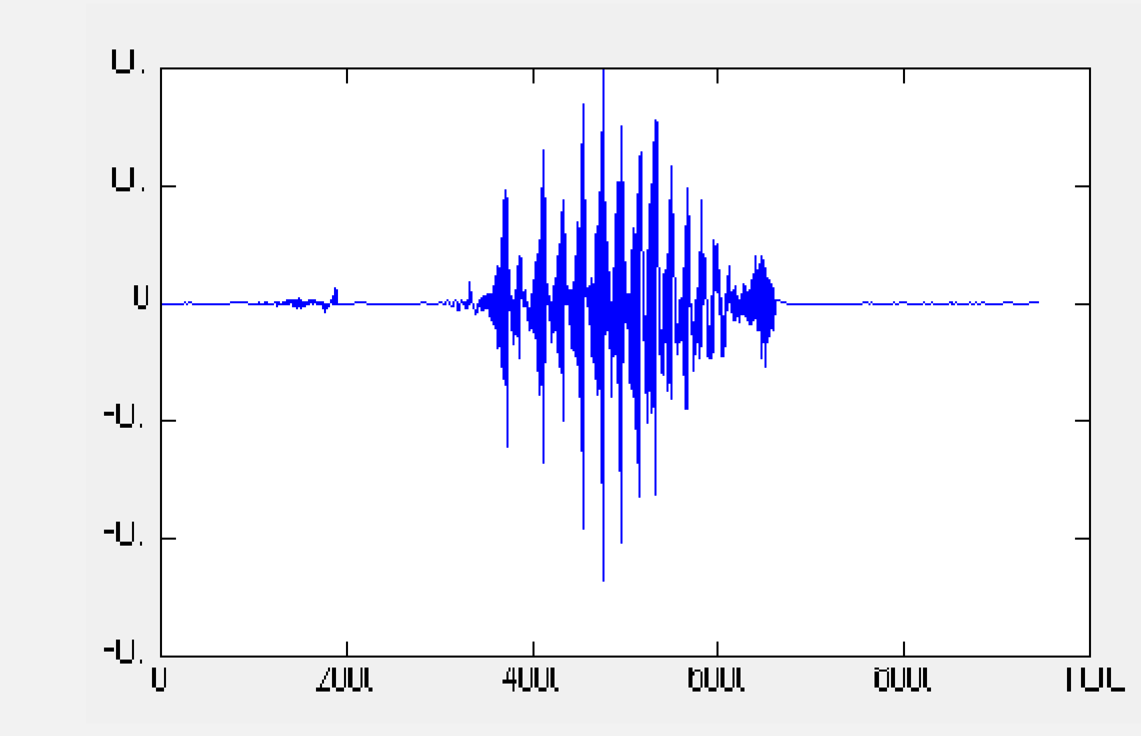
b. The "pot" sound

Reversing the time is equivalent to reversing the frequencies. Therefore, we just need to reverse the DFT of the signal and to apply the inverse DFT.

`[y, FS, BITS] = wavread ("top.wav");`	We open the file.
`N=length(y);`	
`Y=fft(y);` `k=1:N;`	We work on the DFT of the signal.
`YInv=Y(end:-1:1);` `yInv=ifft(YInv);`	We reverse the DFT.
`wavwrite(yInv, FS, BITS, "pot.wav");`	We save the final signal.

Here are the first and the resulting signals.





c. Vanishing with reverberation sound

To obtain the reverberation effect, we add a small time-shift to the signal. We add this new signal to the first one. The time-shift is obtain as before and the addition is immediate, as the DFT is linear.

To have a vanishing effect, we want to multiply the signal with a decreasing coefficient, for instance :

$$h = \left(\left(1 - k/N\right)^2 \right)_{0 \le k \le N-1}$$

Once we have got the DFT of this pseudo signal, we need to convolve it with the DFT of the signal. If Y is the DFT of the initial signal, the DFT of the vanishing signal is :

$$Y_v = \left(\sum_{n=0}^{N-1} Y(n) h((k-n) \bmod N) \right)_{0 \le k \le N-1}$$

```
[y, FS, BITS] = wavread
("paint_ball.wav");
```

We open the file.

```
N=length(y);
Y=fft(y);
k=0:N-1;
```

We work on the DFT of the signal.

```
Coef=(exp(i*2*pi*1000/N)).^k';
Reverb=Y+Coef.*Y;
```

We add the reverberation effect.

```
vanish=((1).-(k'/N)).^2;
Vanish=fft(vanish);
```

We define h (here as vanish) and its DFT.

```
for k=1:N-1
 ReverbVanish(k)=dot(Vanish,[Reverb(k:-
1:1);Reverb(end:-1:k+1)]);
End
ReverbVanish(N)=dot(Vanish,Reverb(end:-
1:1));
```

We convolve the DFT of h and the DFT of the signal.

```
reverbvanish=ifft(ReverbVanish);
wavwrite(reverbvanish, FS, BITS,
"paint_ball_reverb_vanish.wav");
```

We save the final signal.