7. 信号処理

・オーディオデータの使用法
・フーリエ変換
・ノイズ除去

オーディオデータの使用

・audioreadコマンドを用いるとファイルに保存された音声信号を読み込むことができる

・soundコマンドを用いると音声信号を再生できる

```
>> [f,freq]=audioread('test.wav');
>> size(f)
ans =

   211289        1

>> freq
freq =   44100
>> x1=length(f)/freq
x1 =   4.7911
>> x=linspace(0,x1,length(f));
>> plot(x,f);
>> sound(f,freq);
>> sound(f,0.7*freq);
```

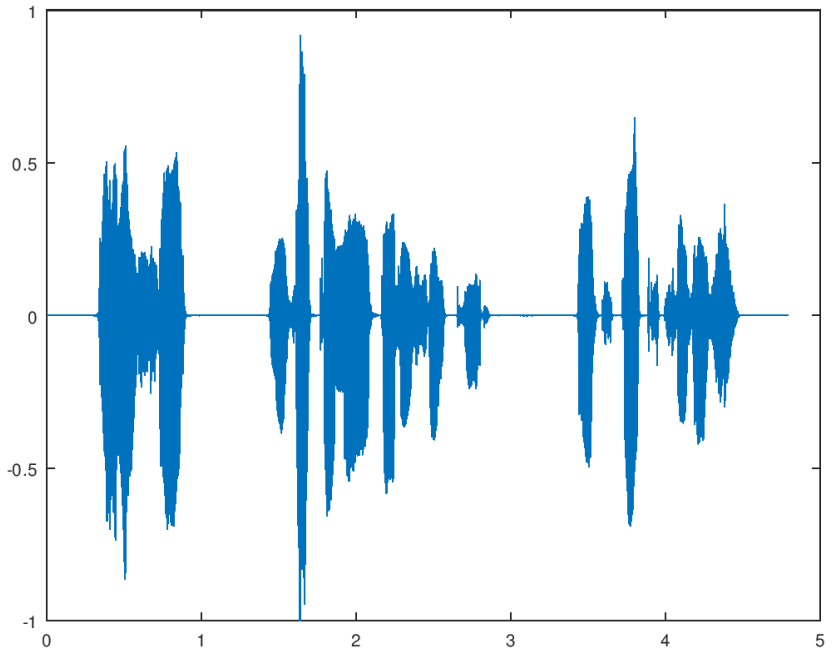
約21万点のデータがある

サンプリングレートは44100 Hz
（1秒当たり4万4千100のデータ点）

音声データの長さは約4.8秒

再生

異なるレートで再生

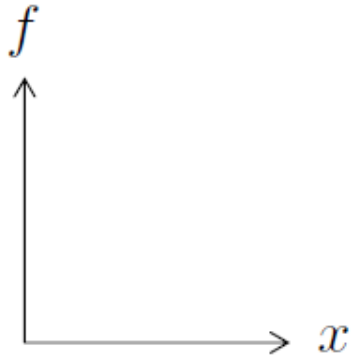


フーリエ変換(1/2)

- フーリエ級数展開：周期的な信号は異なる周波数を持つ三角関数の重み付き線形結合の形で表すことが出来る

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx)$$

- 線形結合のそれぞれの重みは元の信号の周波数成分と見なすことができる



https://commons.wikimedia.org/wiki/File:Fourier_transform_time_and_frequency_domains_(small).gif

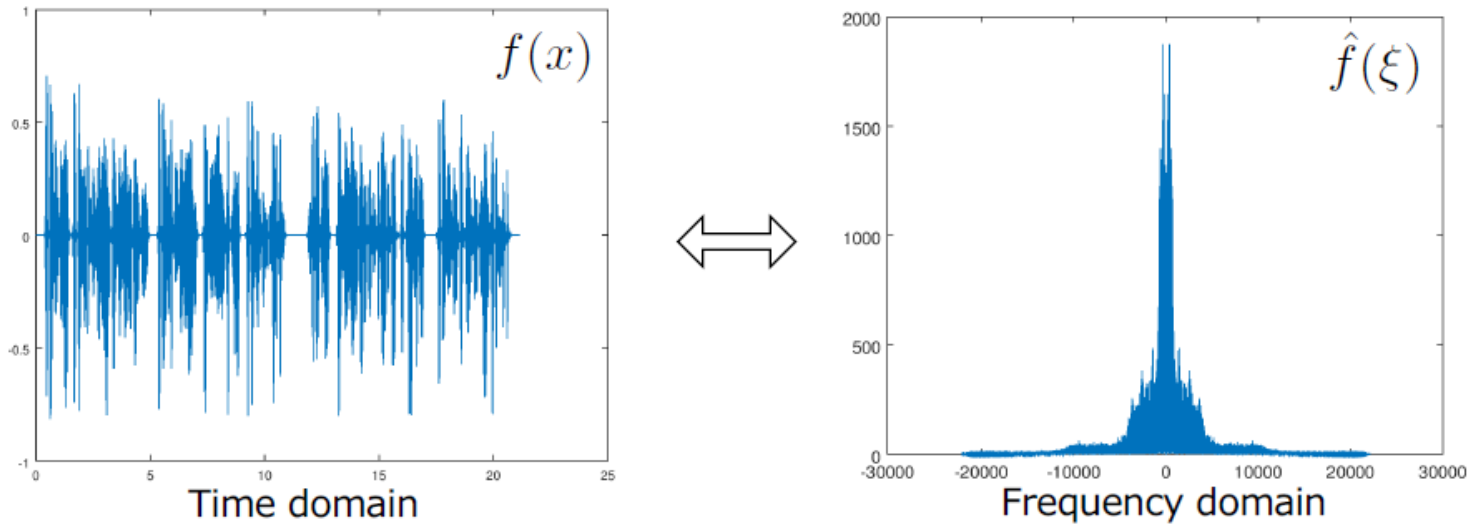
フーリエ変換(1/2)

- (周期的な)信号は周波数成分の形で表すことができる
- フーリエ変換は周波数成分の分布を示している

$$\hat{f}(\xi) := \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx$$

- 周波数領域で記述した$\hat{f}(\xi)$は時間領域に再び変換することができる

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i x \xi} d\xi$$



周波数成分の数値計算

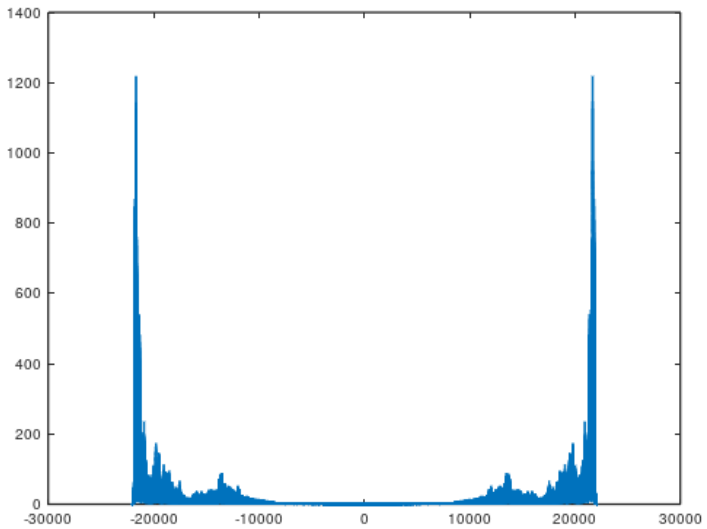
- fftコマンドを用いると(fast Fourier transformアルゴリズムによって)信号のフーリエ変換を実行する

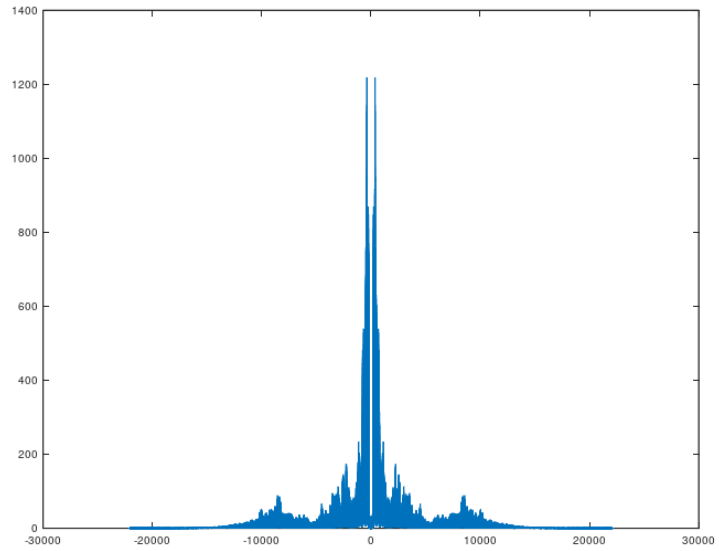
```
>> F=fft(f);
>> fs=size(F)
fs =
   211289        1
>> F(10000)
ans = -26.654 - 40.397i
>> df=freq/length(F);
>> xi=-freq/2:df:freq/2-df;
>> plot(xi,abs(F))
```

```
>> Fshift=fftshift(F);
>> plot(xi,abs(Fshift))
```

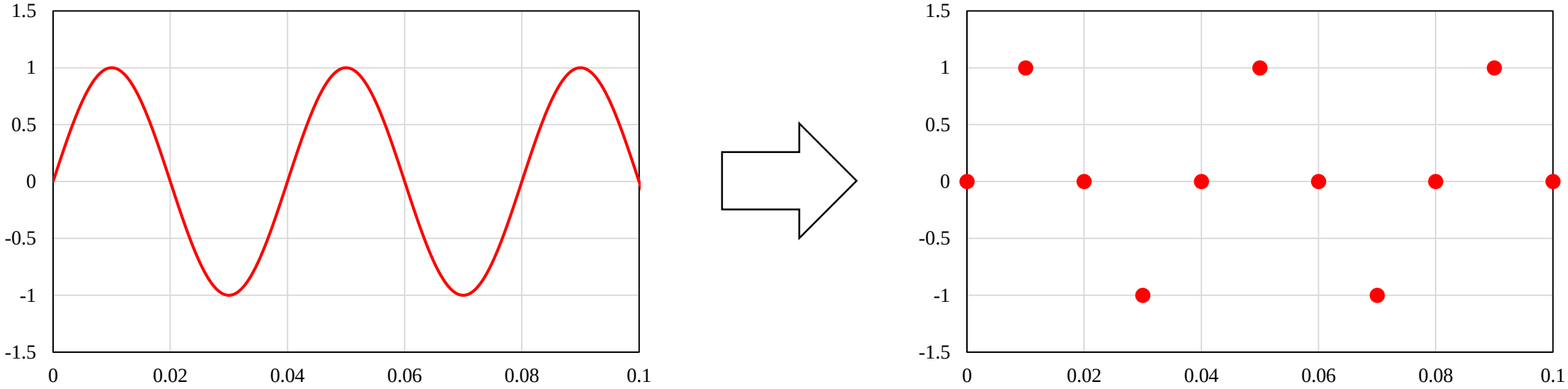
Fは0成分が両端になっているため，fftshiftによって０の周波数成分が中央に来るようにしている

サンプリング周波数の1/2以上の周波数の
信号は正しくサンプルされない
（ナイキスト周波数）

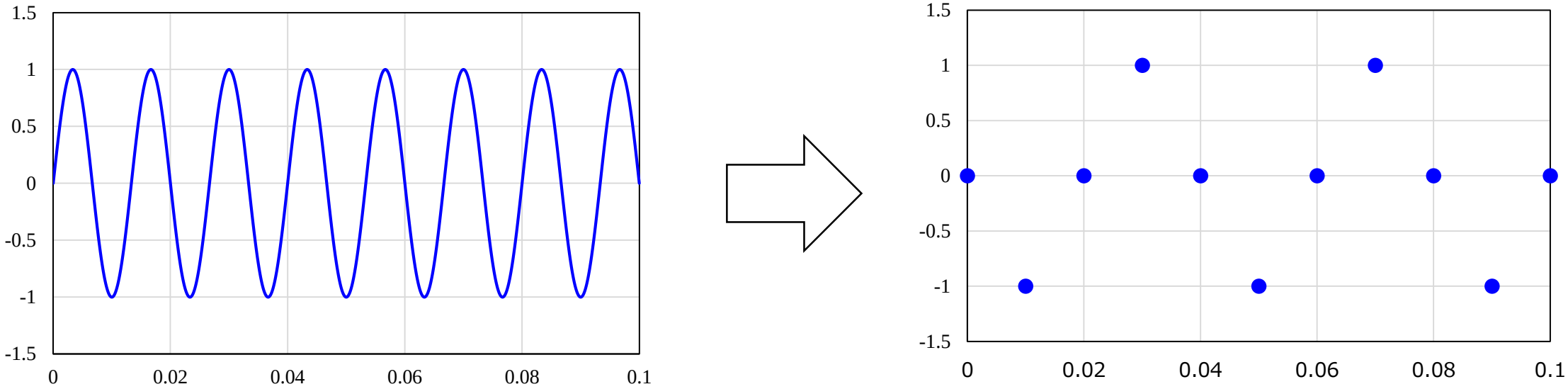




25Hzの波を100Hzでサンプリング



75Hzの波を100Hzでサンプリング



25Hzと75Hzの波が100Hzでサンプリング後に区別がつかない

サンプリング周波数を、計測する信号の持つ最大周波数の2倍以上にする必要がある

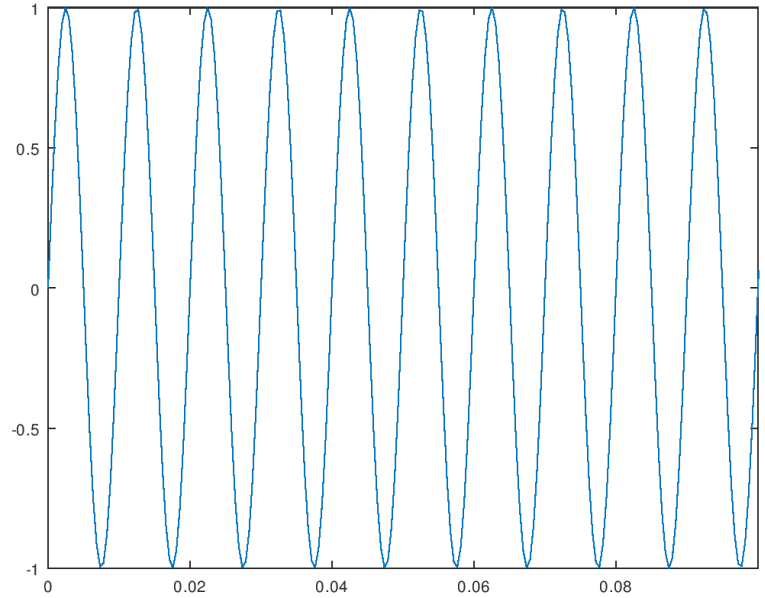
単一周波数の波で確認

```
>> t=linspace(0,1.0,2048);
>> f=sin(2*pi*100*t)';
>> plot(t,f)
>> xlim([0 0.1])
>> freq=length(f)/1.0
freq =   2048
>> F=fft(f);
>> fs=size(F)
fs =

   2048      1

>> df=freq/length(F);
>> xi=-freq/2:df:freq/2-df;
>> Fshift=fftshift(F);
>> plot(xi,abs(Fshift)/(2048/2))
>> xlim([0 200])
```

時間 t： 0から１まで2048分割

波形 f： 周波数100の正弦波

波形 f をプロット

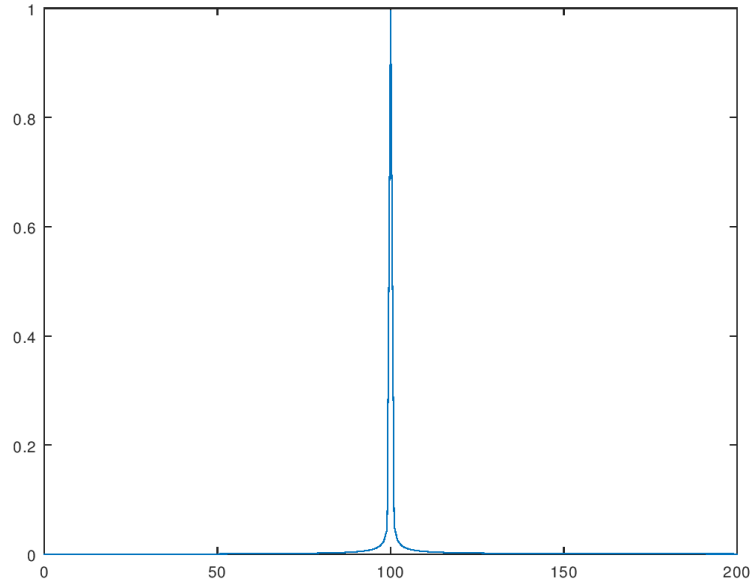


サンプリング周波数 freq=2048/1

fftの実行

周波数空間の分解能

中心を0に



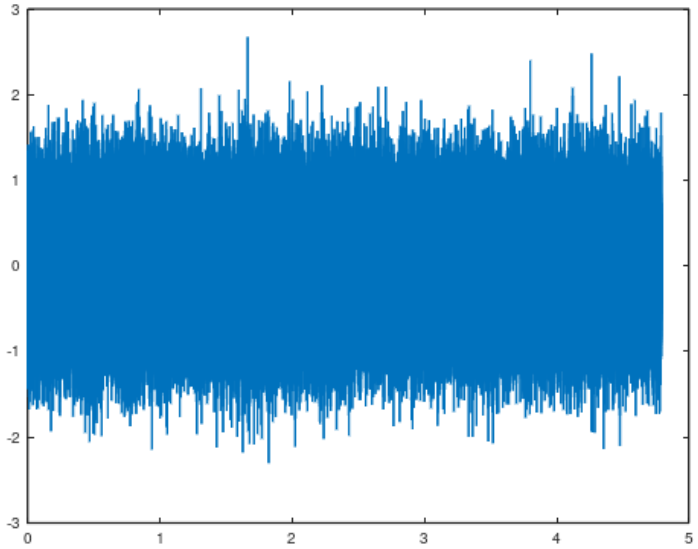
0から200まで拡大

縦軸の値をデータ数の半分で割ると，元の波形の振幅と一致する.

ノイズ除去(1/2)

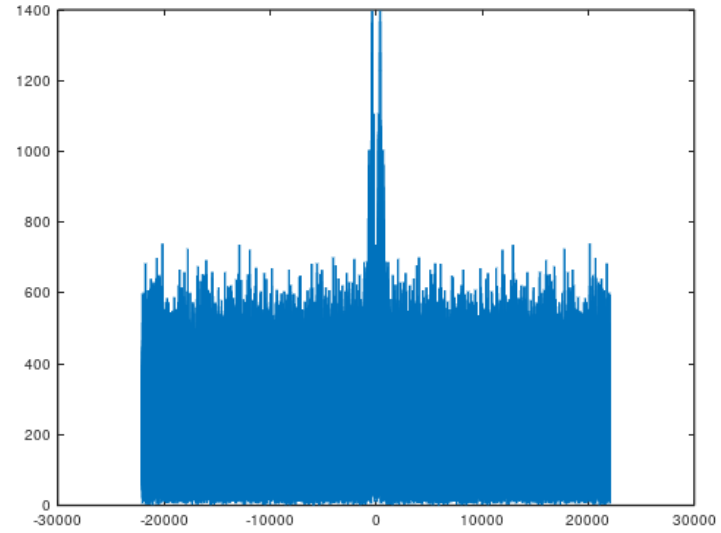
- 先ほどの音声にわざとノイズをのせてみる

```
>> f2=f+0.5*randn(size(f));
>> sound(f2,freq);
>> plot(x,f2)
```



- 信号を周波数領域で見ると

```
>> F2shift=fftshift(fft(f2));
>> plot(xi,abs(F2shift(:,1)))
```



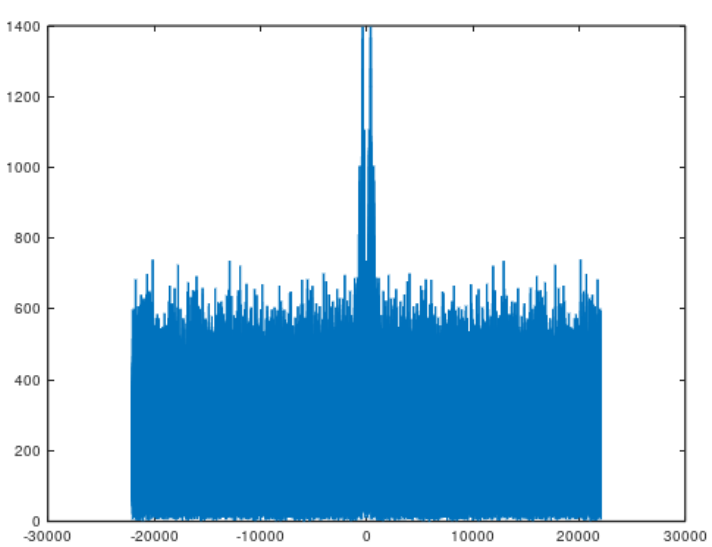
ノイズ除去(2/2)

- 高周波数成分を取り除いてから逆フーリエ変換する
- 3kHz以下の周波数成分だけ取り出すフィルターを作る

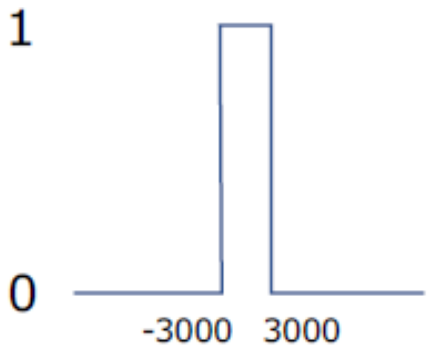
```
>> filter=abs(xi)<3000;
```

- 周波数成分にフィルターをかけて，逆変換(ifftshiftとifft)を行う

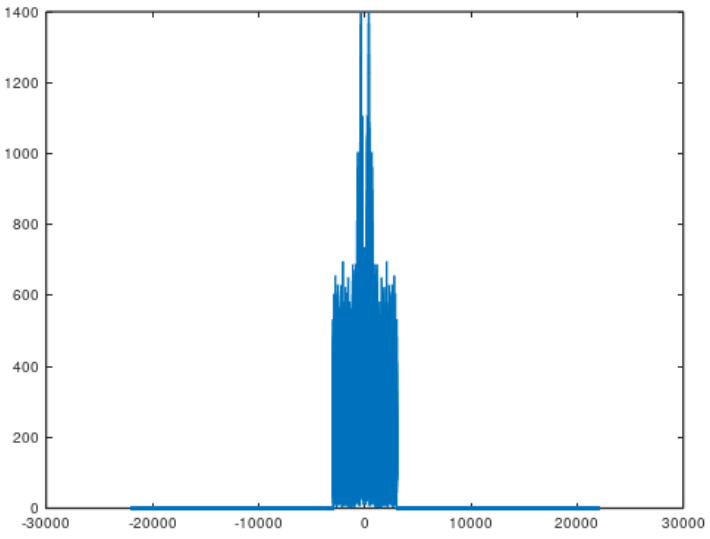
```
>> f2filtered=ifft(ifftshift(F2shift.*filter'));
>> plot(xi,abs(F2shift.*filter')(:,1))
>> sound(f2filtered,freq)
```



×



=



Exercise 7

1. ある周波数の正弦波のデジタルデータを作成し，オーディオファイルとして再生せよ．そのデジタルデータを時間領域と周波数領域にプロットした図をそれぞれ作成せよ.
2. 作成した正弦波のデータをtest.waveに加える．（2つ前のスライドでノイズを加えたように，正弦波を足す）正弦波の加わった信号を時間領域と周波数領域にプロットした図をそれぞれ作成せよ.
3. ２で作成した信号から，正弦波をできるだけ取り除く方法を考えよ．取り除いた結果を時間領域と周波数領域にプロットした図をそれぞれ作成せよ.