SIS Scuola Interdipartimentale
delle Scienze, dell'Ingegneria
e della Salute

DiST
Dipartimento di Scienze
e Tecnologie

L. Magistrale in IA (ML&BD)

Scientific Computing
(part 2 – 6 credits)

prof. Mariarosaria Rizzardi

Centro Direzionale di Napoli – Bldg. C4
room: n. 423 – North Side, 4th floor
phone: 081 547 6545
email: mariarosaria.rizzardi@uniparthenope.it

Contents

- Connection between the Fourier Transform and Fourier coefficients.
- Algorithm for the numerical approximation of the FT.
- Some applications of Fourier Transform.

Connection between the Fourier Transform and Fourier Series coefficients

FT

$$F(\nu)=\int_{-\infty}^{+\infty} f(t)e^{-2\pi i\nu t}\,\mathrm{d}t \quad\Rightarrow\quad F\left(\frac{k}{T}\right)=\int_{-\infty}^{+\infty} f(t)e^{-2\pi i\frac{k}{T}t}\,\mathrm{d}t$$

$$\nu=\frac{k}{T}$$

$$f(t)=0 \quad |t|>\tfrac{T}{2}$$

f: time-limited function

$$\Rightarrow\quad \frac{1}{T}F\left(\frac{k}{T}\right)=\frac{1}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} f(t)e^{-2\pi i\frac{k}{T}t}\,\mathrm{d}t$$

$$\gamma_k=\frac{1}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} f(t)e^{-2\pi i\frac{k}{T}t}\,\mathrm{d}t$$

γ_k: kth Fourier coefficient of f in $[-T/2,+T/2]$

for a time-limited function $f(t)$:

$$F(\nu_k)=F\left(\frac{k}{T}\right)=T\cdot\gamma_k$$

We can apply the same approximation **algorithm** of Fourier coefficients, with an extra step.

SCp2_14d.2

Fourier Transform

(prof. M. Rizzardi)

Algorithm for the numerical approx. of $F(\nu_k)$

Input: $N+1$ equispaced samples $f_j = f(t_j)$ (N: even) in $[-T/2, +T/2]$.

Output: $N+1$ equispaced samples $F_k \approx F(\nu_k)$ in $[-\Omega/2, +\Omega/2]$, where

$$N = \Omega\, T$$

1. Define the vector of samples $\underline{\mathbf{f}}$: $\begin{cases} \mathbf{f}_0 = \frac{1}{2}\left[f(t_0) + f(t_N)\right] \\ \mathbf{f}_j = f(t_j), \qquad j = 1, \dots, N-1 \end{cases}$

2. Compute the **DFT** (MATLAB **`fft()`**).

3. Reorder the vector (MATLAB **`fftshift()`**).

4. Add the last component* and the scale factors ($(-1)^k T/N$, $k = -N/2, \dots, +N/2$).

* equal to the first

new 5. Compute the frequencies: $\nu_k = \dfrac{k}{T}, \quad k = -\frac{N}{2}, \dots, 0, \dots, +\frac{N}{2}$.

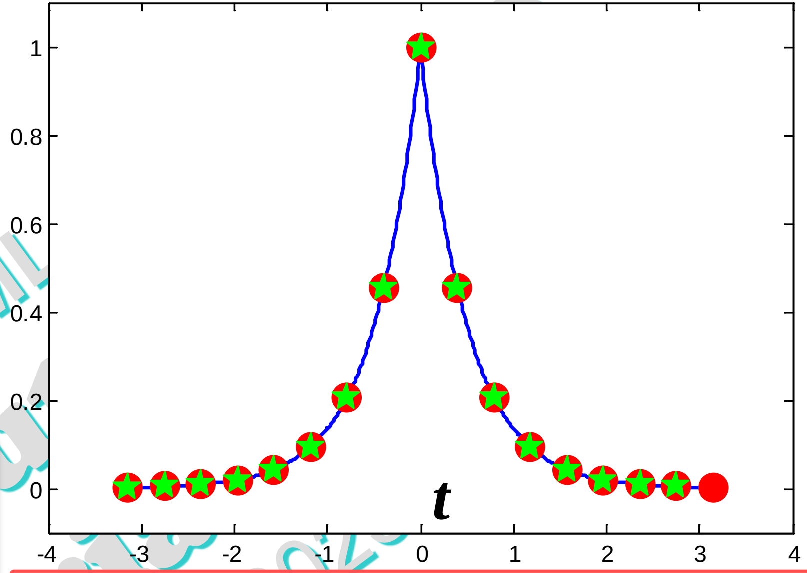
SCp2_14d.3 Fourier Transform (prof. M. Rizzardi)

MATLAB example: FT

```
pf=@(t) exp(-2*abs(t));
T=2*pi; N=16;    tj in [−T/2, +T/2]
tj=T/N*(-N/2:N/2)'; fj=pf(tj);        samples
f=[.5*(fj(1)+fj(end)); fj(2:end-1)];    FT
F=fftshift(fft(f));     F=[F; F(1)]*T/N;
F(2:2:end)=-F(2:2:end); nu=(-N/2:N/2)'/T;
G=F(1:end-1);                          IFT
G(2:2:end)=-G(2:2:end);
g=ifft(fftshift(G)); g=[g; g(1)]/T*N;
x=linspace(-T/2,T/2,499); y=pf(x);
```

the same algorithm as for Fourier Synthesis with series

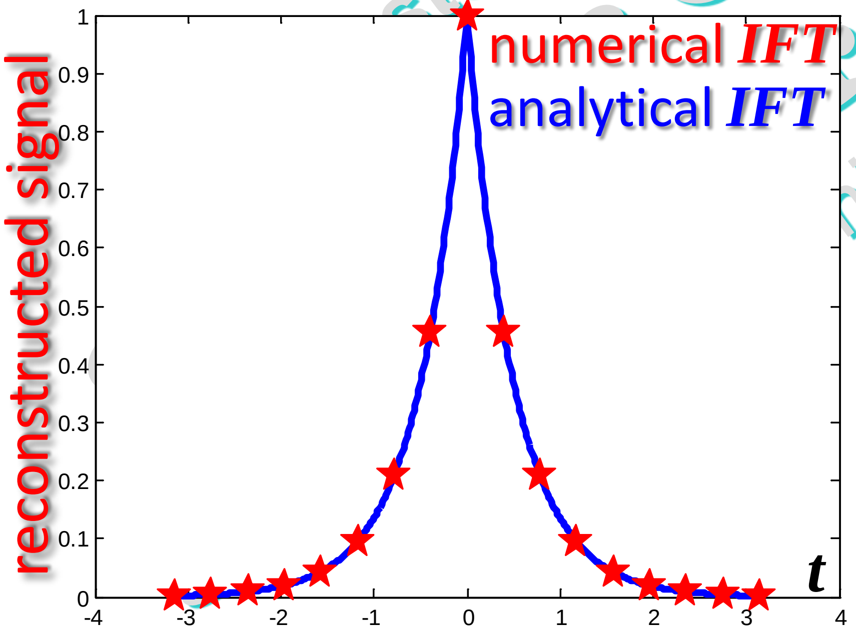
signal: *f*(*t*)=**even decay**

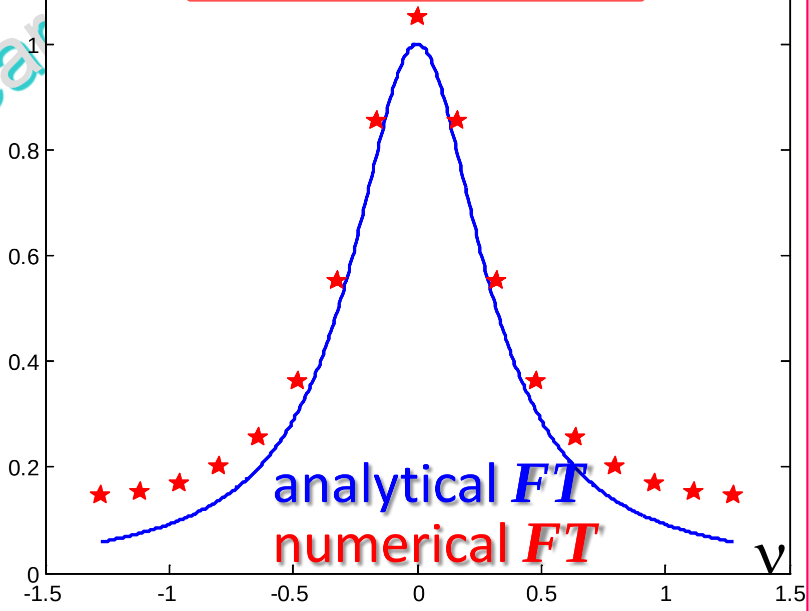


```
plot(x,y,tj,fj,'ro',tj(1:end-1),f,'gp')
```

```
plot(x,y,'b',tj,real(g),'pr')
```

```
plot(nu,abs(F),'rp')
```





How can we add the analytical Fourier Transform for a comparison with its numerical samples?

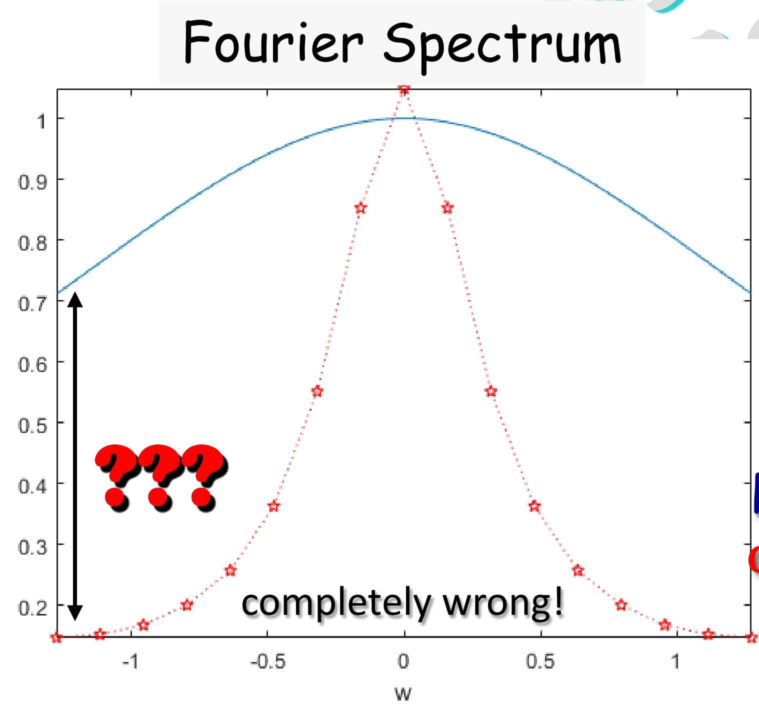
```
pf=@(t)exp(-2*abs(t));  T=2*pi; N=16;  tj=T/N*(-N/2:N/2)'; fj=pf(tj);
f=[.5*(fj(1)+fj(end)); fj(2:end-1)]; F=fftshift(fft(f)); F=[F; F(1)]*T/N;
F(2:2:end)=-F(2:2:end); nu=(-N/2:N/2)'/T;
plot(nu,abs(F),'pr:')
```

numerical FT

```
syms t real
Fw=fourier(sym(pf));
hold on; fplot(abs(Fw),[-1.5,1.5])
axis tight
```

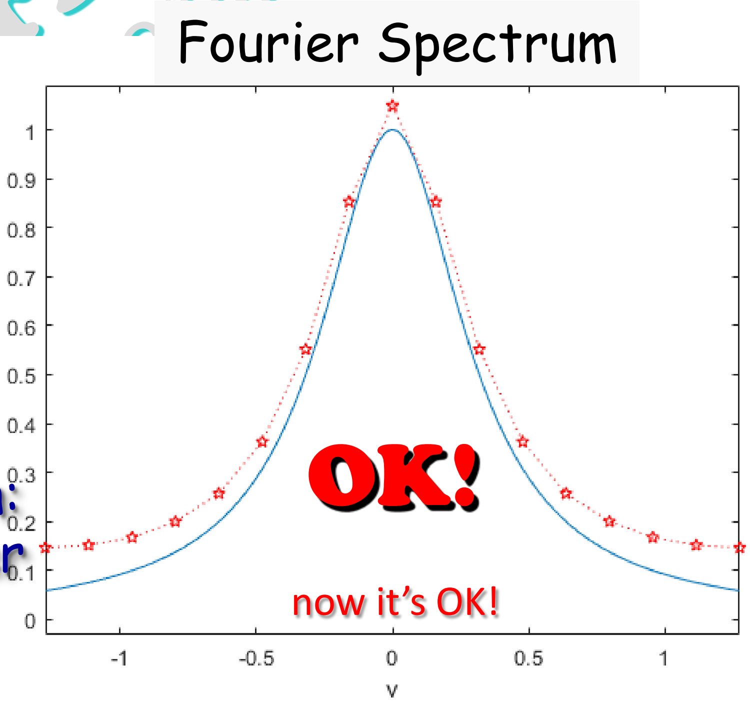
```
syms t w v real; O=N/T;
Fw=fourier(sym(pf));
Fv=subs(Fw,w,2*pi*v);
hold on; fplot(abs(Fv),[-O/2,O/2])
```

$\omega=2\pi\nu$



numerical FT uses circular frequency

Pay attention: ω and ν differ



ω ←→ ν

SCp2_14d.5

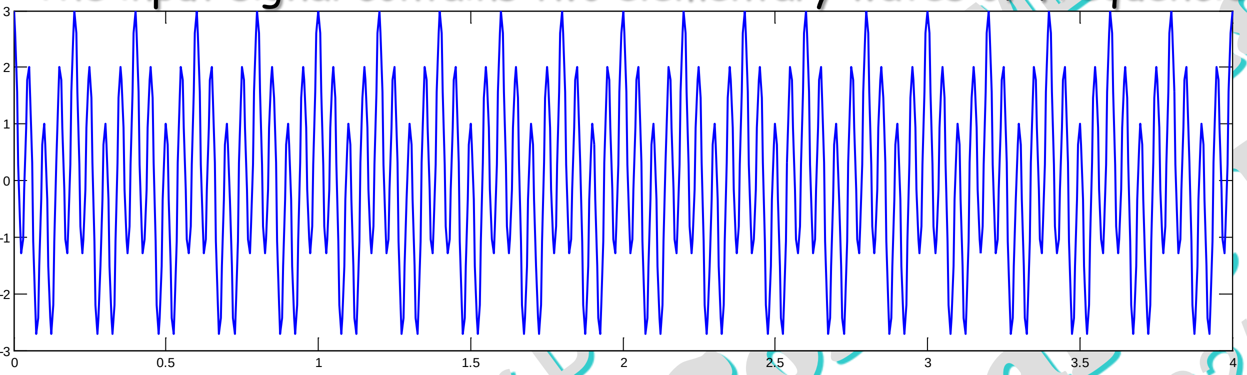
Fourier Transform

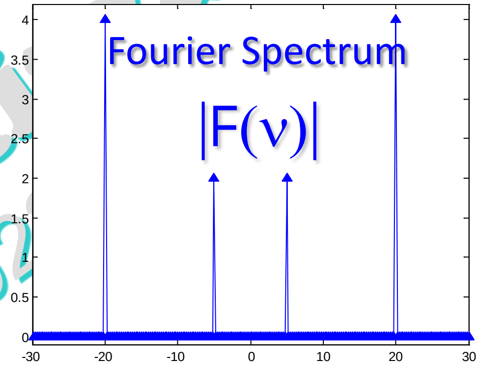
(prof. M. Rizzardi)

Applications example 1: ideal filter

```
dt=.005; t=(0:dt:4)'; N=numel(t)-1; T=4;
fj=cos(2*pi*5*t)+2*cos(2*pi*20*t); plot(t,fj); axis([0 4 -3 3]);
f=[.5*(fj(1)+fj(end));fj(2:end-1)];                          FT
F=fftshift(fft(f));F=[F;F(1)]*T/N; F(2:2:end)=-F(2:2:end);
nu=(-N/2:N/2)'/T;
figure; plot(nu,abs(F),'^-'); axis([-30 30 -.1 4.2])
```

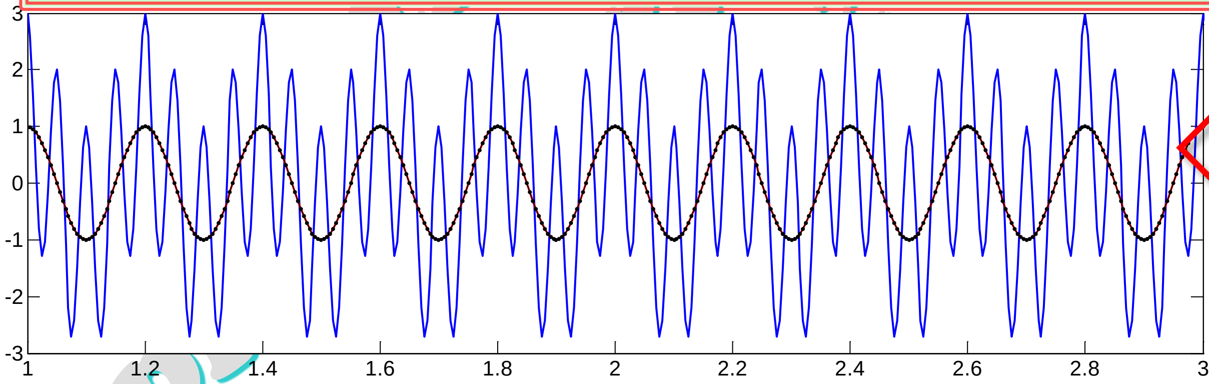
The input signal contains two elementary waves of frequencies 5 and 20 respectively

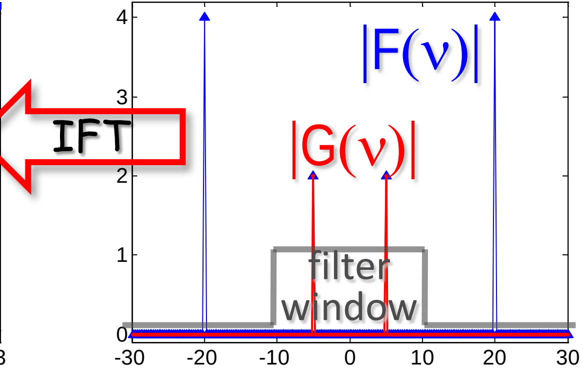




We want to apply a "low-pass filter" to the input signal.

```
G=zeros(size(F)); k=find(abs(nu)<10); G(k)=F(k); hold on; plot(nu,abs(G),'r.-')
H=G(1:end-1); H(2:2:end)=-H(2:2:end);
h=ifft(fftshift(H)); h=[h;h(1)]/T*N;      IFT
figure; plot(t,fj,t,real(h),'r', t,cos(2*pi*5*t),'.k--'); axis([1 3 -3 3])
```





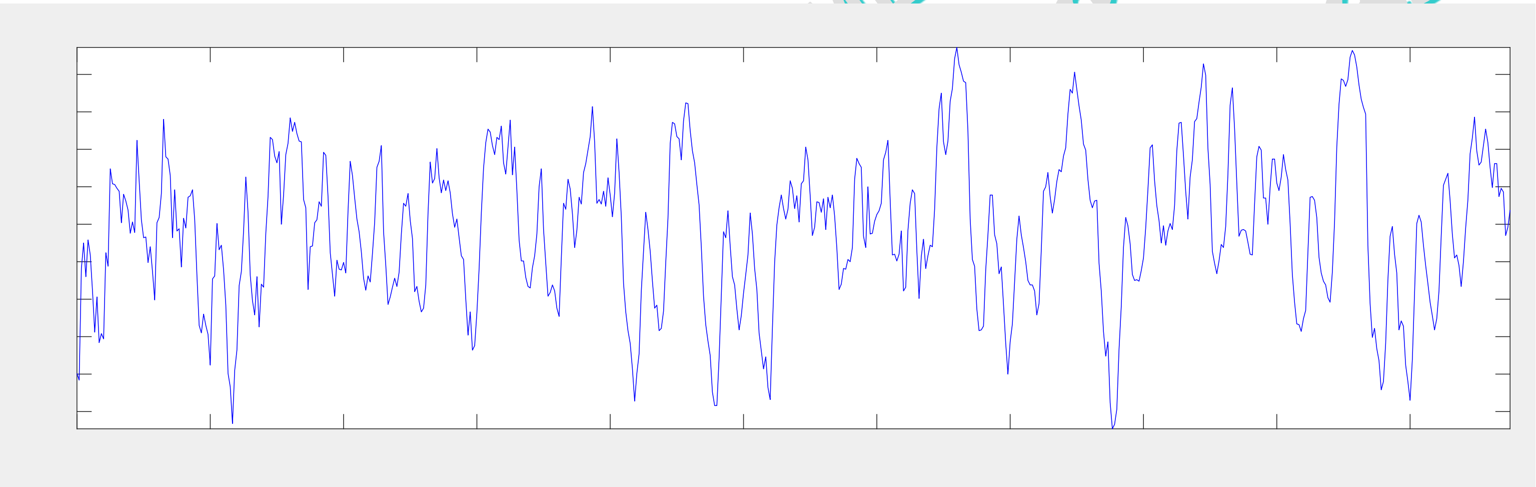
The two signals h(t) and cos(2π5t) overlap perfectly. The “noise” 2cos(2π20t) has been totally removed

Applications
example 2

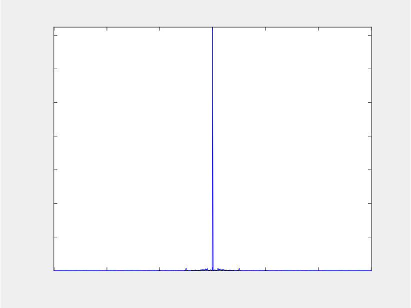
Study of a possible periodicity of equispaced data: how to find the period of "El Ninho“?

```
load ninho; N=numel(fj);
tj=(0:N-1)'/12 + 1950;  % time in years
plot(tj,fj,'b'); ylabel('temperatura boe'); xlabel('anni (un campione per mese)')
```

download ninho.mat

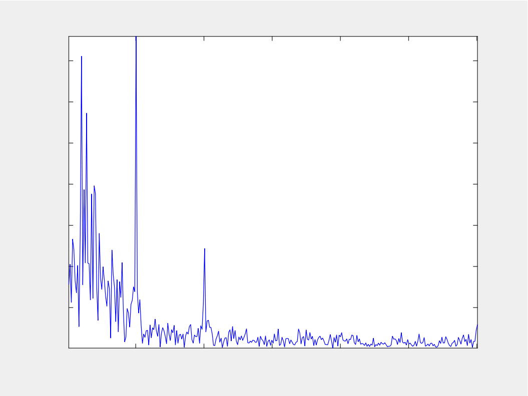


```
T=tj(end)-tj(1); Dt=T/N;  % about 1 sample per month
Dnu=1/T; nu=(-N/2:N/2)'*Dnu;
F=fftshift(fft(fj)); F=[F; F(1)]*T/N;
F(1:2:end)=-F(1:2:end);
plot(nu,abs,'b'); axis tight
title('Spettro di Fourier')
xlabel('frequenze (cicli/anno)')
```



```
mid=N/2+2; plot(nu(mid:end),abs(F(mid:end)),'b')
axis tight; title('Spettro','FontSize',18)
xlabel('0<frequencies(cycles/year)','FontSize',16)
```

only positive frequencies

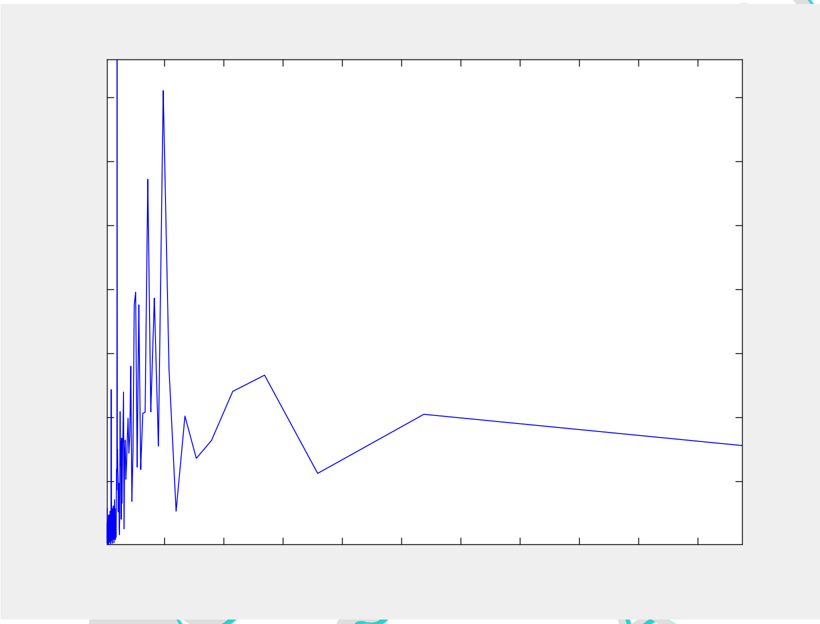


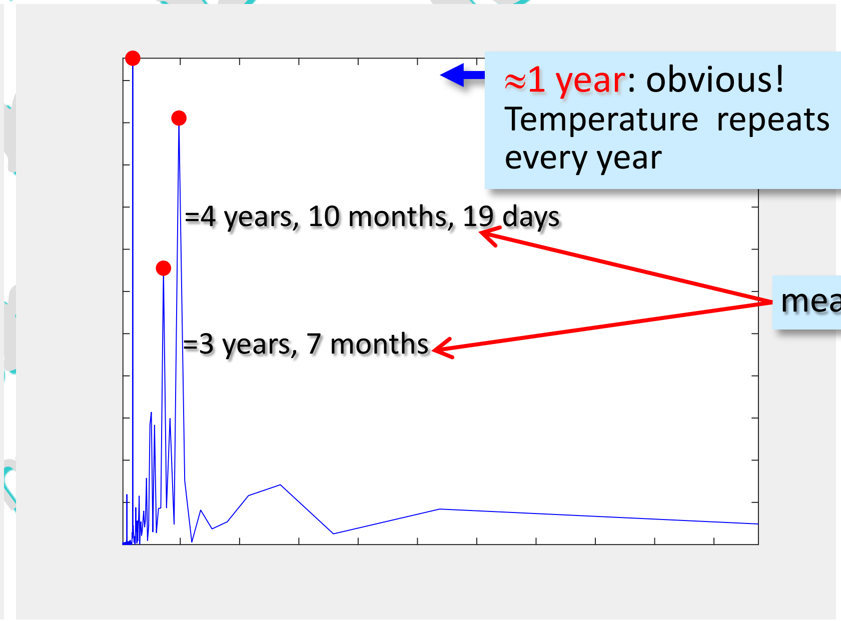
Applications
example 2

period of "El Ninho"

```
mid=N/2+2;  % only freq > 0
period=1./nu(mid:end);  % instead of frequencies, on abscissas we use periods (in years)
plot(period,abs(F(mid:end)),'b'); axis tight
xlabel('period=1/freq (years/cycle)','FontSize',16)
title('Spectrum','FontSize',18)
power=abs(F(mid:end)).^2;
plot(period,power,'b'); axis tight
title(‘Power Spectrum','FontSize',18); xlabel(…); hold on
```

Power Spectrum = $\{|F_k|^2\}_k$





≈1 year: obvious!
Temperature repeats cyclically every year

=4 years, 10 months, 19 days

=3 years, 7 months

mean ≈ 4 years

```
%% find the index of the 1st maximum in the Power Spectrum
index=find(power == max(power));  % 1st
p1=period(index); plot(p1,power(index),'r.', 'MarkerSize',25)
text(period(index),power(index),[' 1^{st} period = ' num2str(period(index)) ' years'])
```

SCp2_14d.8

Fourier Transform

(prof. M. Rizzardi)

Applications
example 3

Dual-tone multi-frequency (DTMF) phone keypad

SCp2_14d.9

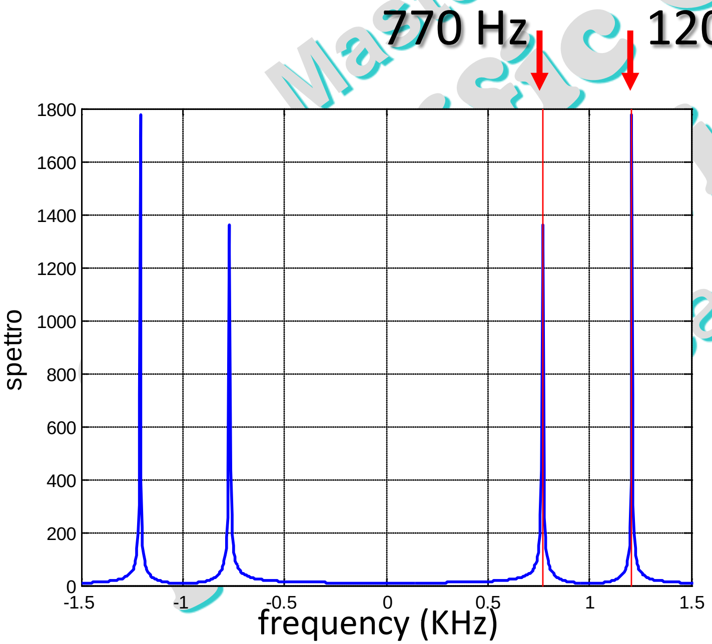
The sound ***y*** of each key is the sum of two “tones”, i.e. two sinusoids of suitable frequencies:

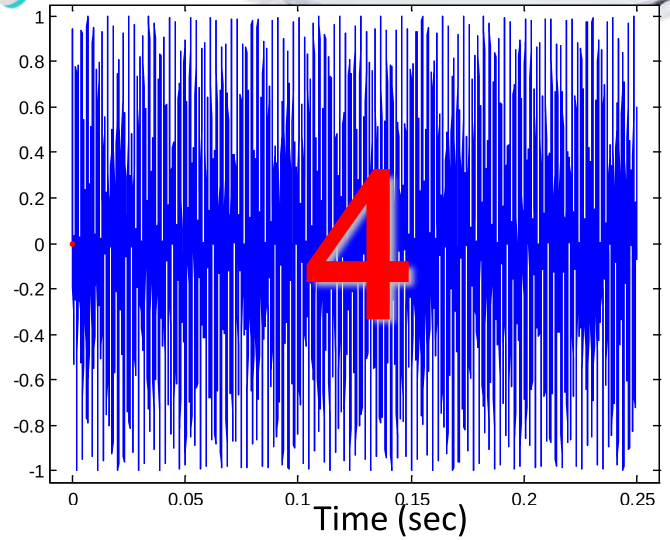
$$y=\frac{\sin\left(2\pi\phi_{\text{row}}t\right)+\sin\left(2\pi\phi_{\text{col}}t\right)}{2}$$

high frequencies ϕ_{col}

low frequencies ϕ_{row}

Hz	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

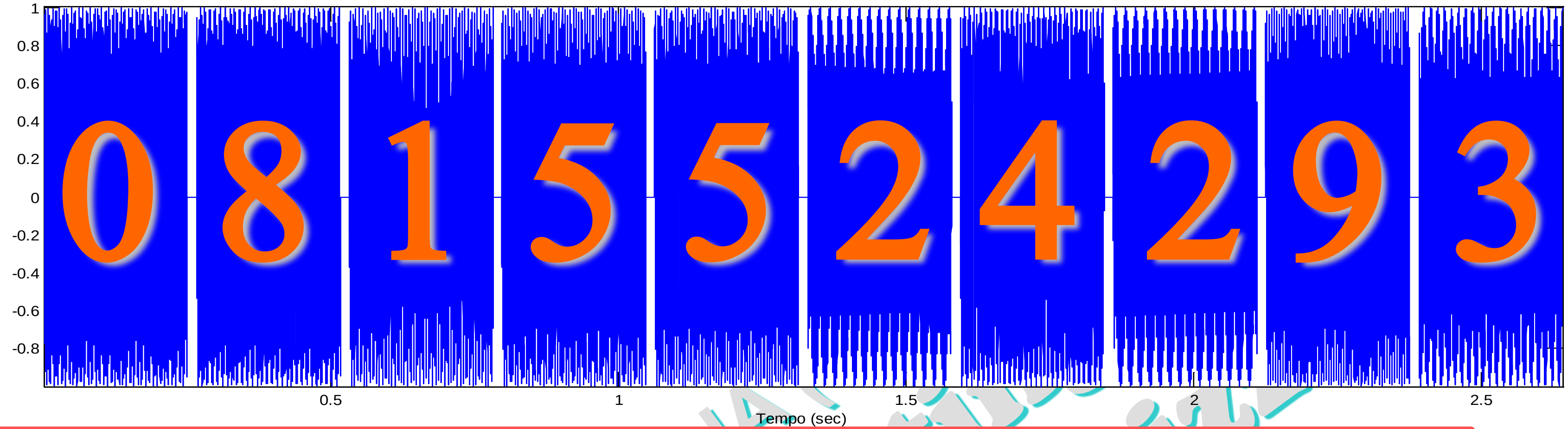




Fourier Transform

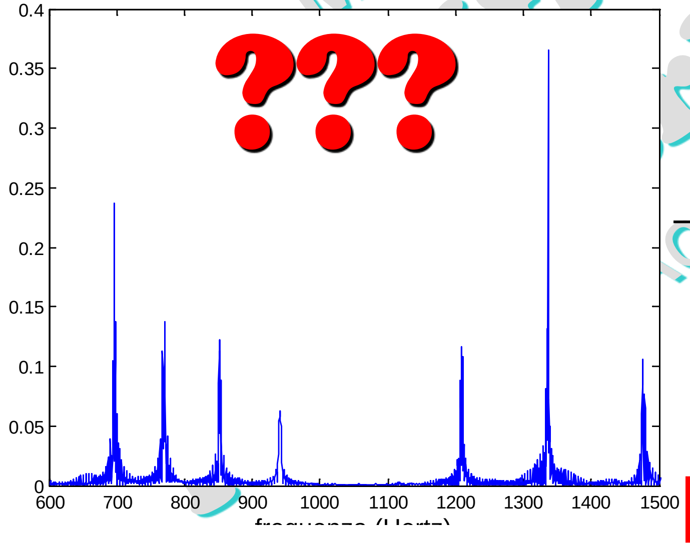
(prof. M. Rizzardi)

example 3

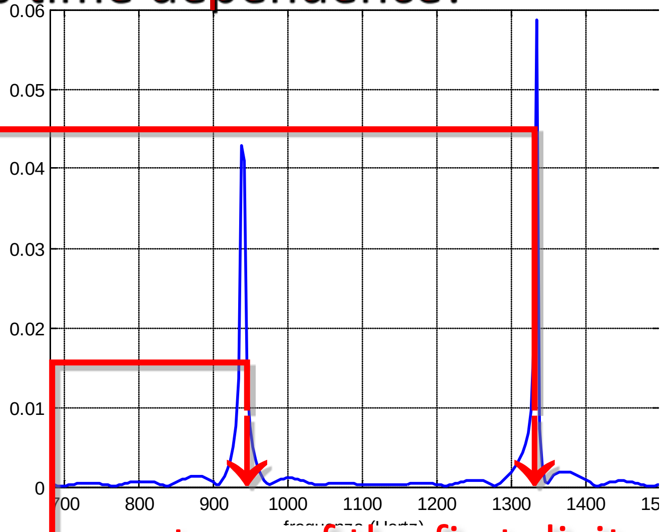


```
info=audioinfo('PhoneNumber.wav')
[y,fs]=audioread('PhoneNumber.wav'); sound(y,fs) % fs: sample rate
N=numel(y); Dt=1/fs; tj=Dt*(1:N)'; T=N*Dt;
figure(1); plot(tj,y,'b'); axis tight; xlabel('Tempo (sec)')
Y=fftshift(fft(y)); Y=[Y;Y(1)]*T/N; Dnu=1/T; nuk=(-N/2:N/2)'/T;
figure(2); plot(nuk,abs(Y),'b');
xlabel('frequency (Hertz)'); ylabel(‘Spectrum of the whole phone number')
```

the sound varies with time: the spectrum of the whole signal contains all the component frequencies, without showing its time dependence!



Hz	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#



spectrum of the first digit

SCp2_14d.10

Fourier Transform

(prof. M. Rizzardi)

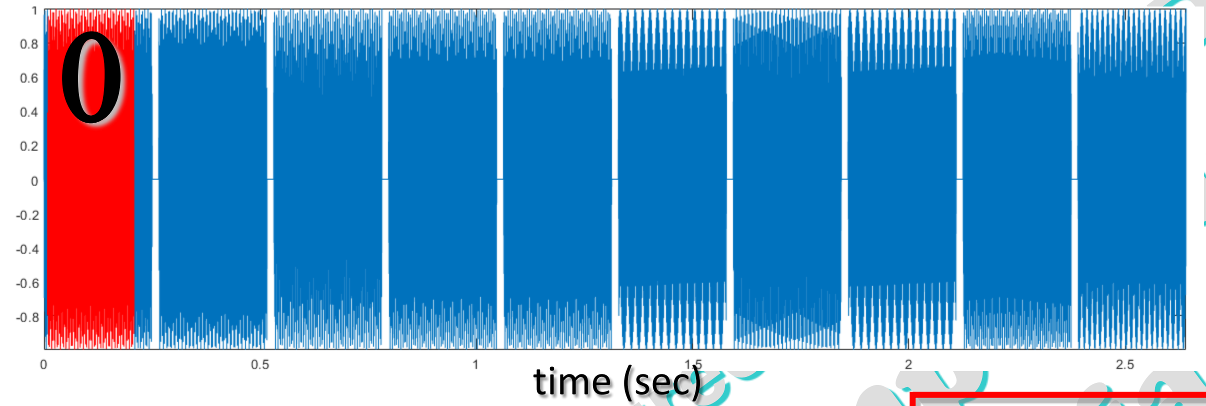
A moving window should be applied to the signal: "Short Time Fourier Transform (STFT)"

Download

STFT.p

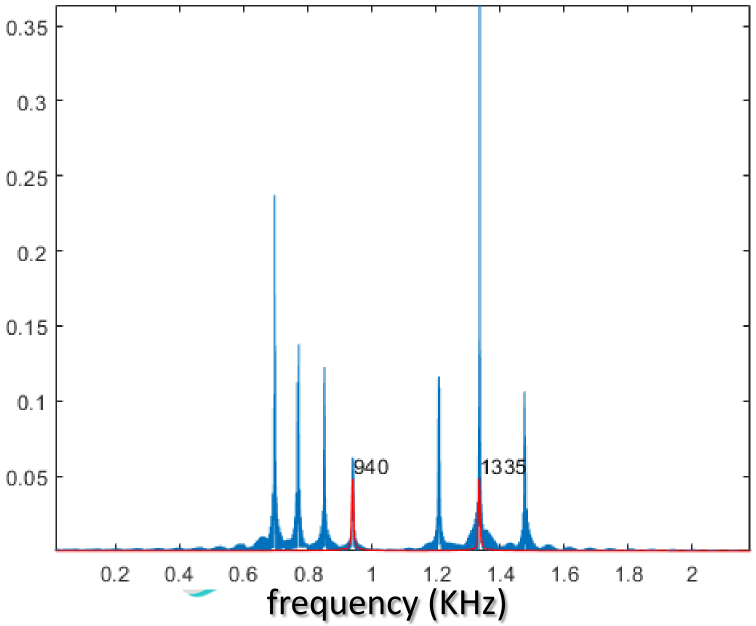
and PhoneNum.zip

PhoneNumber.wav

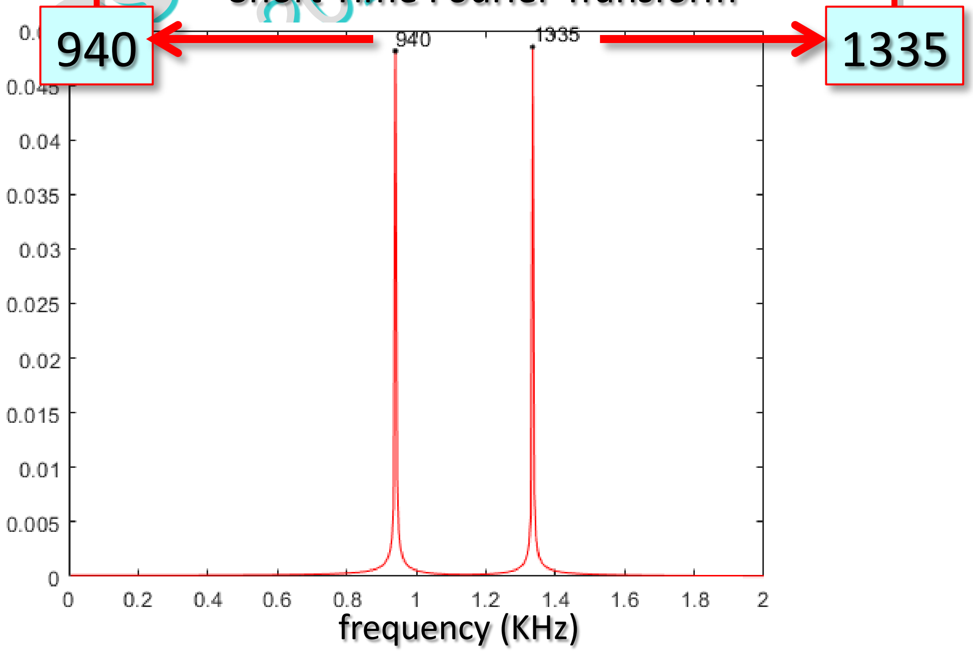


Hz	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

Fourier Spectrum



Short Time Fourier Transform

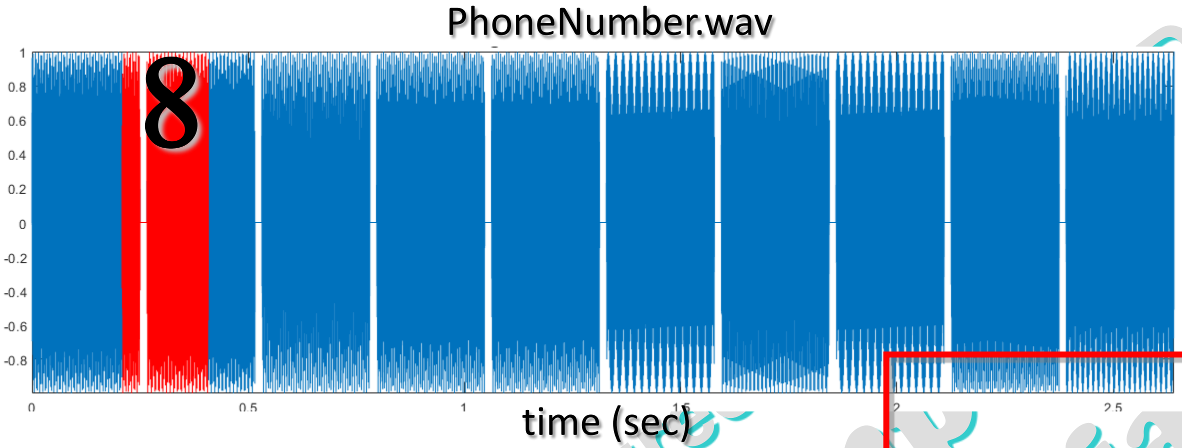


A moving window should be applied to the signal: "Short Time Fourier Transform (STFT)"

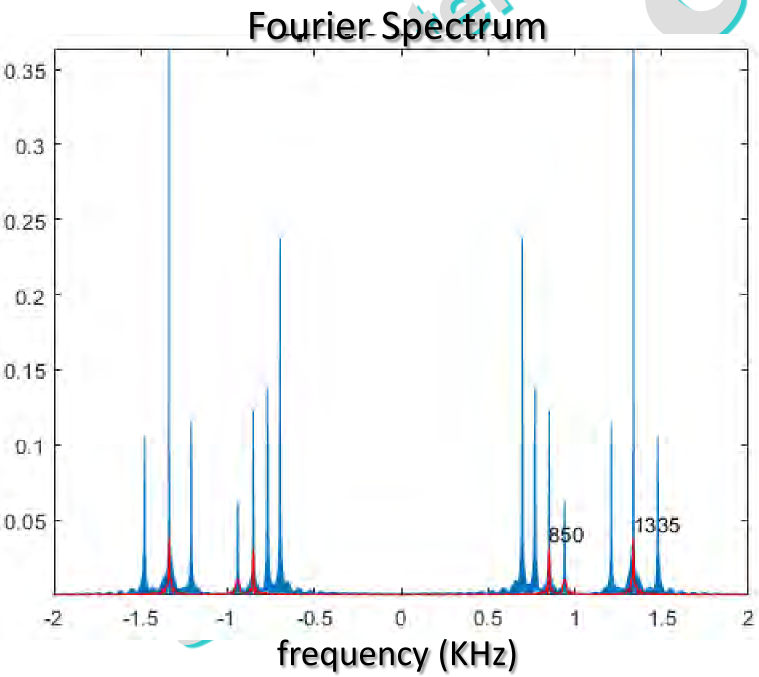
Download

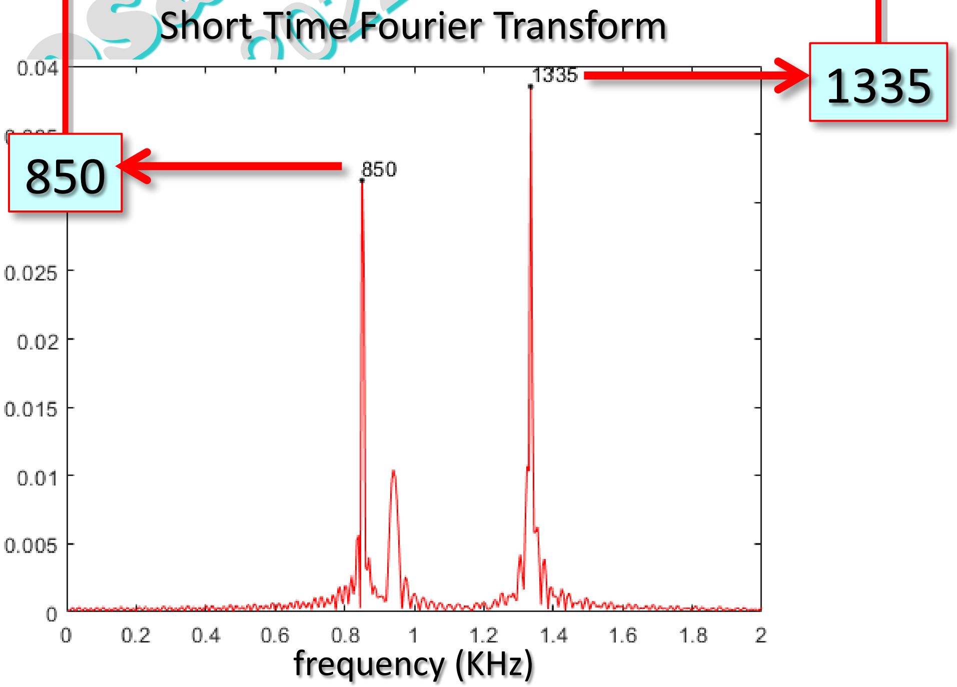
STFT.p

and PhoneNum.zip



Hz	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#





A moving window should be applied to the signal: "Short Time Fourier Transform (STFT)"

Download

STFT.p

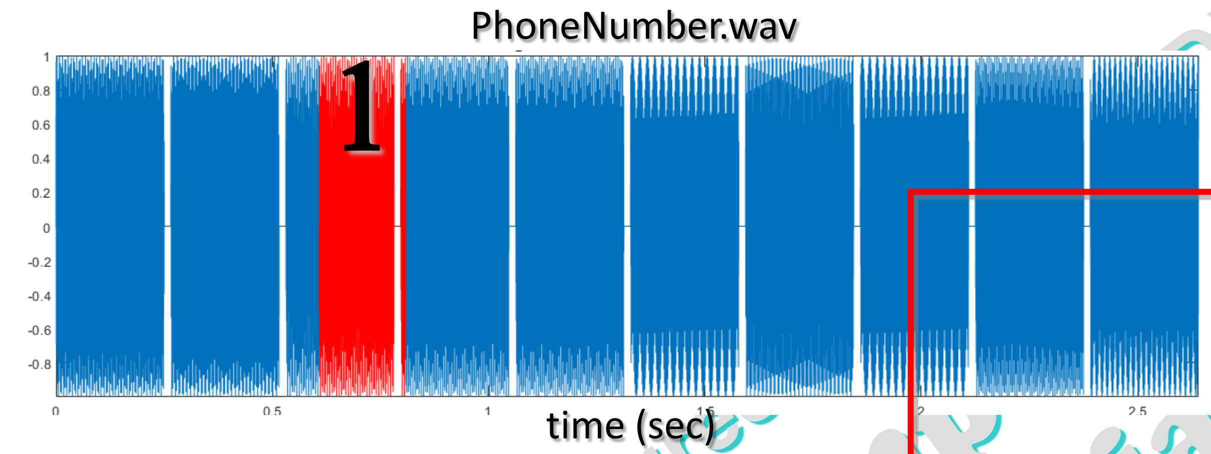
and PhoneNum.zip

SCp2_14d.13

Fourier Transform

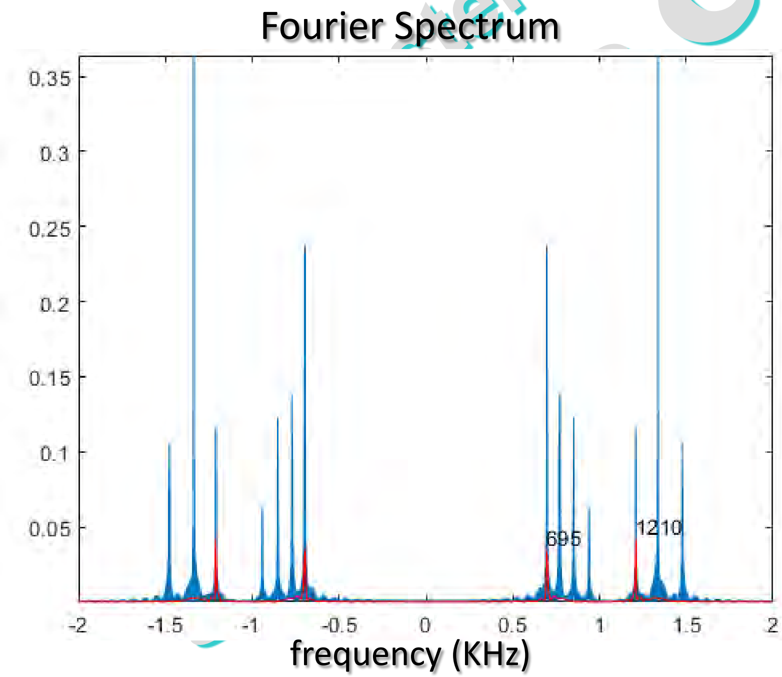
(prof. M. Rizzardi)

PhoneNumber.wav

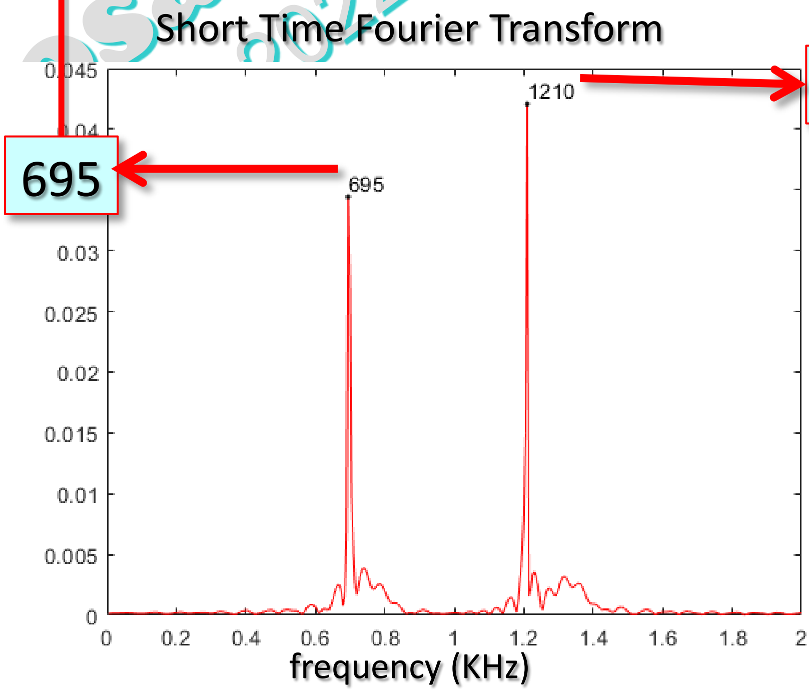


Hz	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

Fourier Spectrum



Short Time Fourier Transform



695

1210

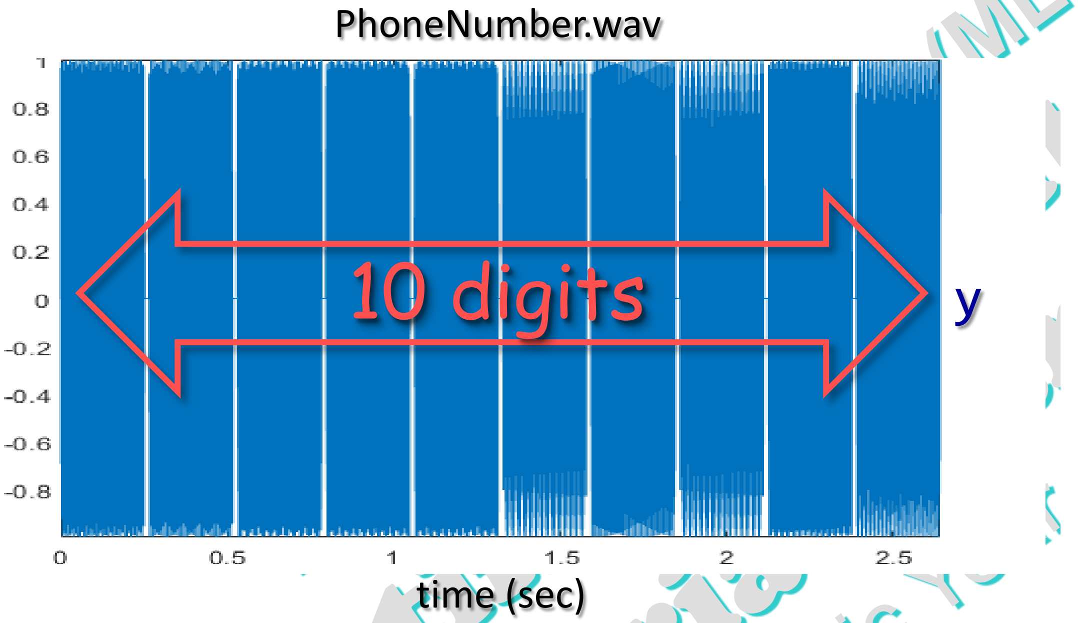
A moving window should be applied to the signal: "Short Time Fourier Transform (STFT)"

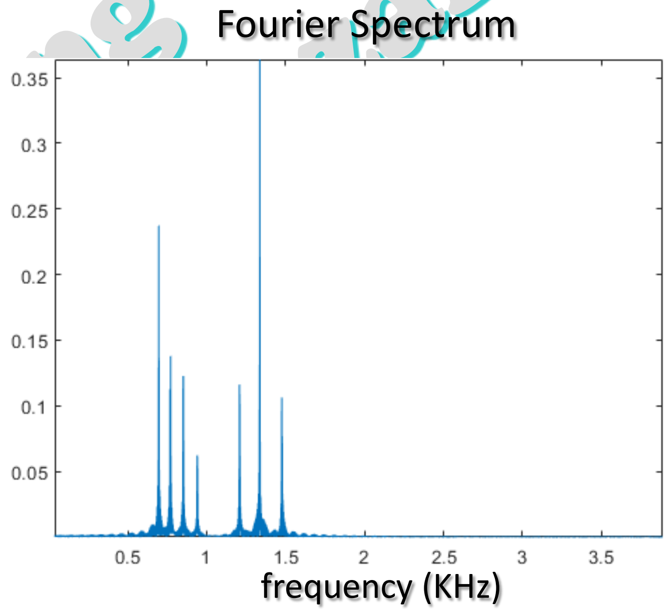
Download

STFT.p

and PhoneNum.zip

Put the code STFT.p in the same folder as files in PhoneNum.zip





Run: STFT.p and answer as follows to the questions

```
Total Duration (sec) = 2.6409
Reduce the window? [y/n]: y
T0: origin of the new window (0 <= t0 < 2.6409) = 0
Width T of the new window = numel(y)/10*Dt
```

the best answer