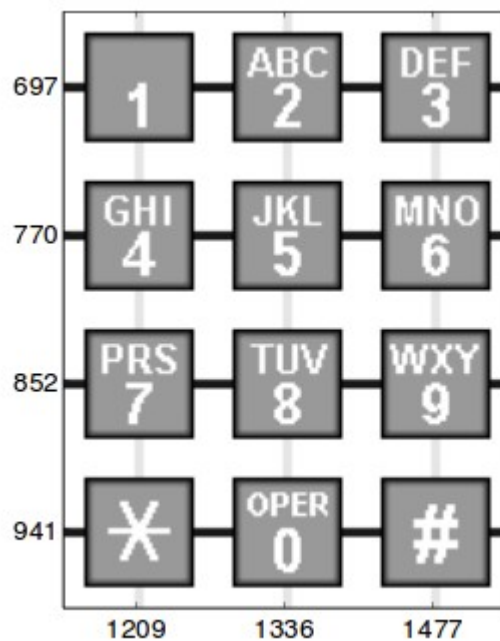


Alcune tastiere producono un suono diverso a seconda di qualche tasto viene schiacciato. In particolare si mostra in figura un esempio di tale tastiera:



I numeri in corrispondenza delle linee (697, 770, 852, 941) corrispondono alla frequenza armonica elementare, allo stesso modo i numeri in corrispondenza delle colonne (1209, 1336, 1477) sono le frequenze che caratterizzano le stesse colonne. Quando si schiaccia un tasto in riga k e colonna j si produce un suono che corrisponde alla media tra due funzioni sinusoidali con frequenze corrispondenti alla riga/colonna selezionata.

Memorizziamo in due vettori le frequenze:

```
fr = [697 770 852 941];
fc = [1209 1336 1477];
```

Per questa tastiera il suono viene prodotto campionando con tasso di campionamento

$F_s = 32768$

per cui $dt=1/F_s=1/32768$. Generiamo quindi i valori (in secondi) del tempo che consideriamo, tempo finale T ,

```
t = 0:1/Fs:T;
```

per esempio se consideriamo 0.4 secondi abbiamo,

```
t = 0:1/Fs:0.4;
```

Un tasto in posizione (k,j) produrrà il suono y seguente:

```
y1 = sin(2*pi*fr(k)*t);
y2 = sin(2*pi*fc(j)*t);
y = (y1 + y2)/2;
```

Supponiamo di aver registrato i suoni prodotti da una persona che utilizzava la tastiera, i dati sono stati memorizzati nel file `touchtone.mat` (presente nel sito Ariel), possiamo caricarli in MATLAB:

```
load touchtone
```

Il file contiene una struttura `y`, scrivendo quindi semplicemente `y` (e poi <invio>), si ottiene,

```
y =  
sig: [1x74800 int8]  
fs: 8192
```

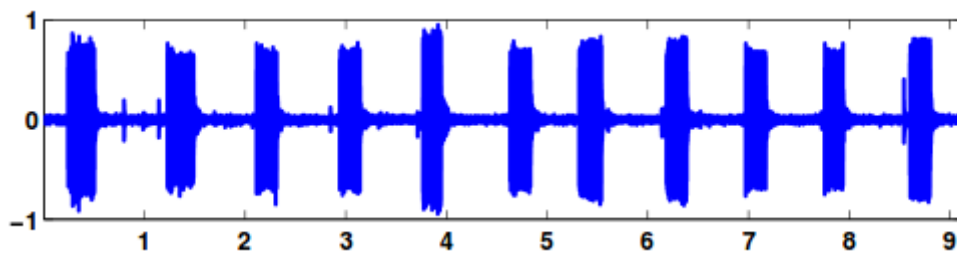
che indica che `y` è una struttura composta da due campi: un vettore `sig` di tipo `int8` (interi 8 bit con segno), ed uno scalare `fs` (con il tasso di campionamento che abbiamo utilizzato). Con il comando `max(abs(y.sig))` è facile verificare che i valori sono compresi tra -127 e 127, normalizziamo tali valori e cambiamo il tipo dato in `double` (per poter operare con l'aritmetica floating point),

```
Fs = y.fs;  
y = double(y.sig)/128;
```

Generiamo l'intervallo di tempo dell'intera registrazione con la suddivisione determinata dal tasso di campionamento,

```
n = length(y);  
t = (0:n-1)/Fs
```

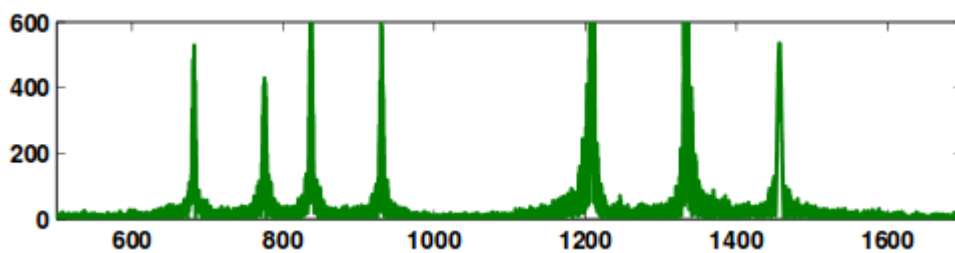
Possiamo rappresentare con `plot(t,y)` il segnale registrato (la registrazione è lunga poco più di 9 secondi).



Problema: Attraverso la FFT trovare quali tasti sono stati schiacciati (sono stati composti 11 tasti).

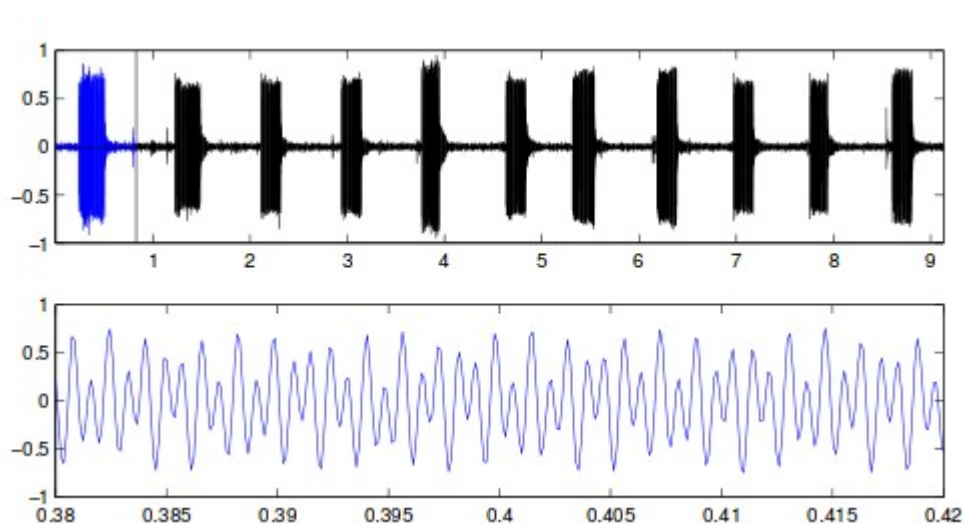
Non posso fare la trasformata discreta di Fourier dell'intero segnale perchè non riuscirei a distinguere la sequenza dei tasti (potrei riconoscere la presenza di alcuni ma non posso ricostruire quando sono stati schiacciati nella sequenza). Di seguito la rappresentazione di una parte dello spettro dell'intero segnale,

```
p = abs(fft(y));
f = (0:n-1)*(Fs/n);
plot(f,p);
axis([500 1700 0 600])
```

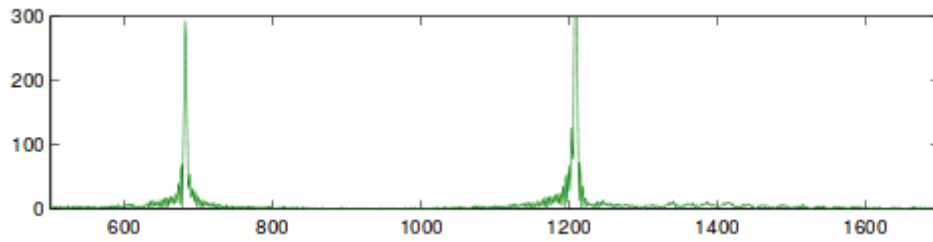


NOTA. Attenzione per il calcolo delle frequenze perchè nell'esempio con i dati di El Nino il campionamento era di ampiezza 1 (un mese), qui le frequenze vanno riscalate $f = (0:n-1)*(Fs/n)$; (per l'esempio di El Nino $Fs=1$).

Per risolvere il problema occorre suddividere il segnale in 11 parti e per ogni segmento considerare la trasformata discreta di Fourier ed analizzare le massime frequenze presenti, per esempio selezioniamo una prima parte del segnale,



con spettro,



Da cui si deduce la presenza delle frequenze “dominanti” 697 e 1209, quindi il primo tasto schiacciato è il tasto “1”. Si prosegue per i restanti segmenti di segnale.

NOTA. Il materiale di questo esercizio è preso da C. Moler, *Numerical Computing with MATLAB*, MathWorks Ed.