

Campionamento, DFT Esperienza 9

Marco Vignati
Laboratorio di Sistemi e Segnali 2018-19.

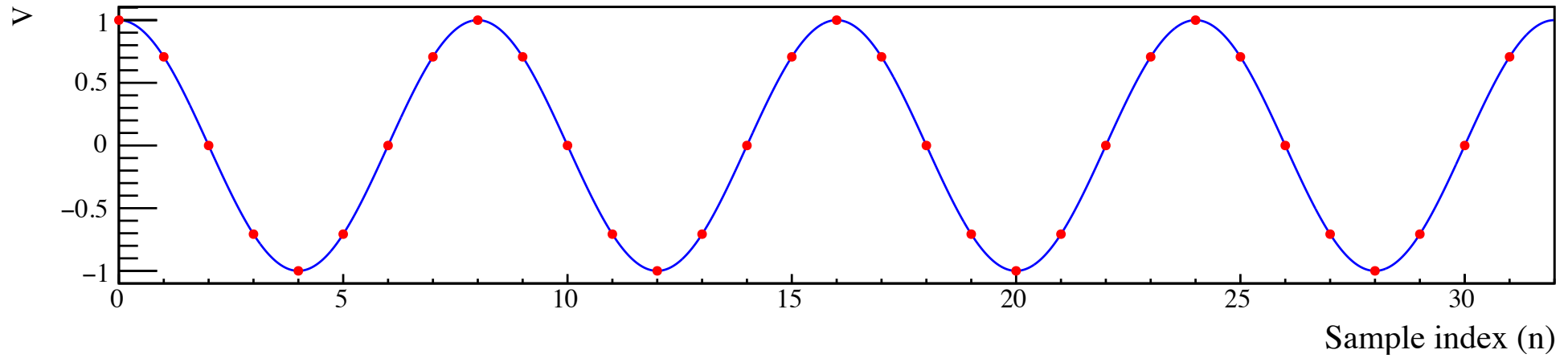
Teorema del campionamento

- Quando campioniamo un segnale con un ADC dobbiamo considerare la risoluzione in tensione e in tempo.
- La risoluzione in tensione è: $dV = V / \text{count} = \text{“scala ADC in V”} / 2^n$
- La risoluzione in tempo è legata alla frequenza di campionamento.
 - ▶ Il teorema di Shannon Nyquist stabilisce che per ricostruire un segnale sinusoidale con frequenza f , questo deve essere campionato con una frequenza superiore o uguale a $f_s = 2f$.
 - ▶ Alternativamente, dato un ADC con frequenza di campionamento f_s , la massima frequenza ricostruibile di un segnale è la cosiddetta **frequenza di Nyquist**:

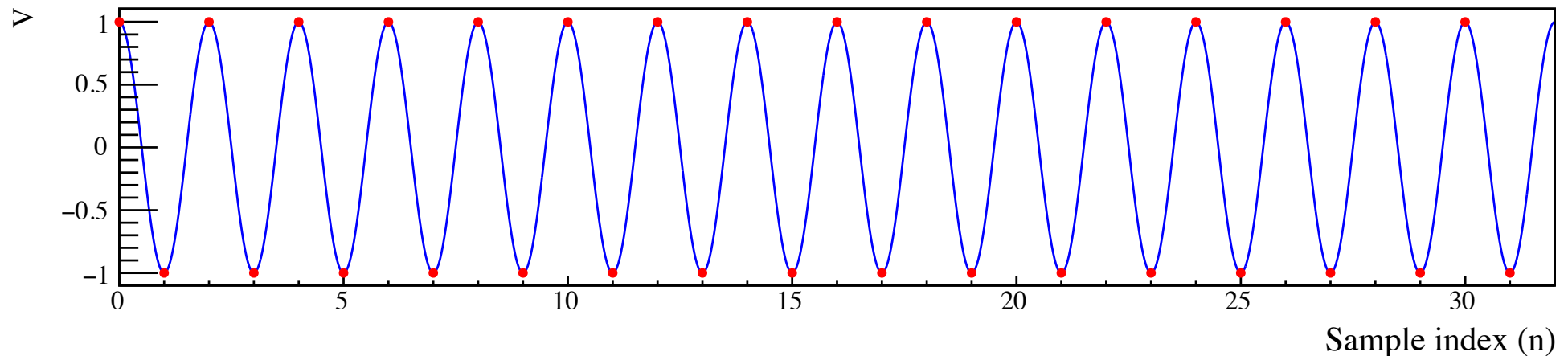
$$f_N = \frac{f_s}{2}$$

Campionamento di sinusoidi

$f < f_s/2$: Segnale ben campionato

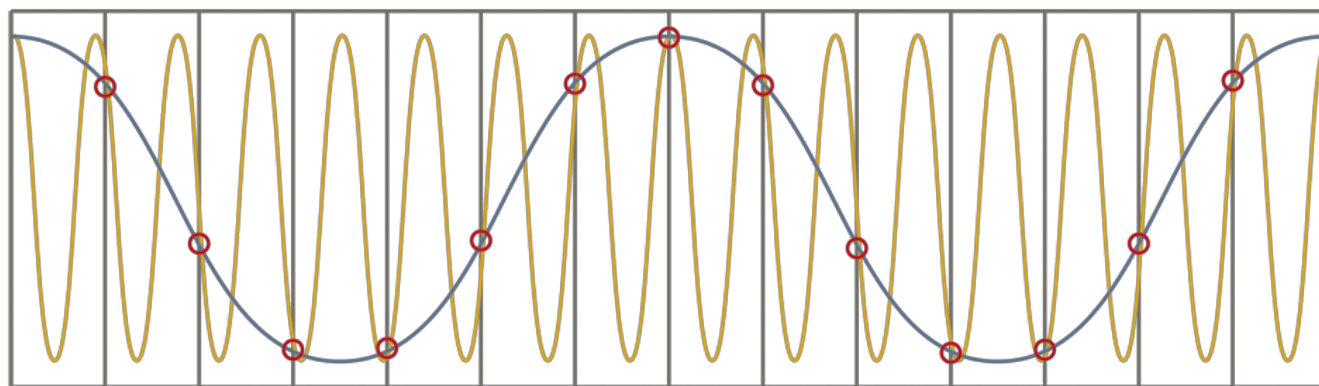
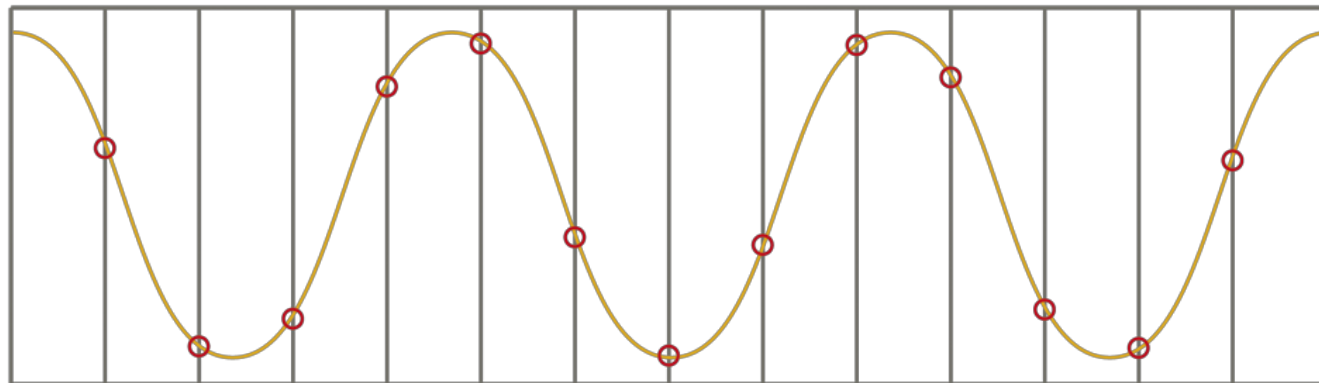


$f = f_s/2$: Segnale al limite del campionamento



Aliasing

- Quando la frequenza del segnale acquisito supera quella di Nyquist si verifica il fenomeno dell'aliasing.
- La frequenza apparente è inferiore a quella reale. Esempio:



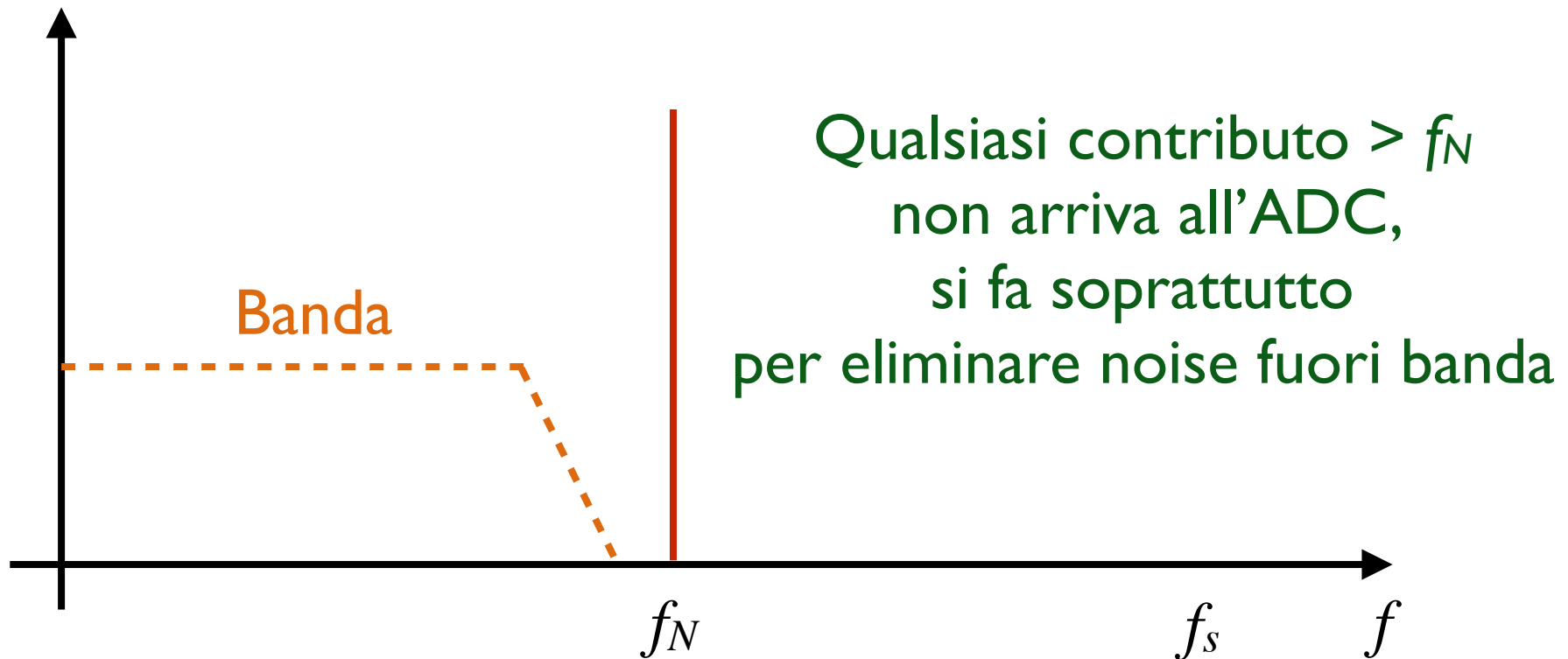
Wagon wheel effect

- Quando osserviamo le ruote delle automobili in corsa, può apparire che esse girino lentamente o addirittura all'incontrario. Questo succede perché l'occhio umano campiona ad una frequenza equivalente di ~ 30 Hz, inferiore a quella di rotazione della ruota.
- L'effetto è ancora più visibile nelle riprese video dove il campionamento è fissato a 25 frame per second.
- Esempio:
 - ▶ <https://www.youtube.com/watch?v=bl8lrqBBAXQ>



Filtri anti-alias

- L'aliasing è un fenomeno non voluto quando si campiona con un ADC
- Per evitarlo è sufficiente mettere un filtro passa-basso prima dell'ADC con frequenza di taglio inferiore a quella di Nyquist.
- Molti ADC commerciali includono il filtro anti-alias.



Discrete Fourier transform (DFT)

- Serie di Fourier:

$$a_k = \frac{1}{T} \int_0^T f(t) \cos\left(\frac{2\pi}{T} kt\right) dt \quad b_k = \frac{1}{T} \int_0^T f(t) \sin\left(\frac{2\pi}{T} kt\right) dt$$

- Trascrizione della serie di Fourier in forma complessa $X_k = a_k - ib_k$:

$$X_k = \frac{1}{T} \int_0^T f(t) e^{-i\frac{2\pi}{T} kt} dt \quad k = -\infty, \infty$$

- Trasformata di Fourier discreta di un segnale campionato f_n :

$$X_k = \sum_{n=0}^{N-1} f_n e^{-i\frac{2\pi}{N} kn} \quad \begin{array}{l} n = 0, \dots, N-1 \\ k = 0, \dots, N-1 \end{array}$$

- ▶ N è il numero di campionamenti ADC in una forma d'onda di durata T

Antitrasformata

- Alla trasformata complessa

$$X_k = \sum_{n=0}^{N-1} f_n e^{-i \frac{2\pi}{N} kn}$$

- Corrisponde l'antitrasformata

$$f_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i \frac{2\pi}{N} kn}$$

- Si possono incontrare normalizzazioni di (trasformata, antitrasformata) diverse da $(1, 1/N)$. Ad esempio $(1/N, 1)$ o $(1/\sqrt{N}, 1/\sqrt{N})$
 - ▶ Dipende dalle librerie sw ed è fonte di confusione ed errori (basta vedere sui vari forum sul web...)

Caratteristiche della DFT

- **Dominio del tempo:** N è il numero di campionamenti ADC di una forma d'onda di durata T
 - ▶ Ogni campionamento ha durata: $\Delta t = T/N$
 - ▶ In relazione alla frequenza di campionamento: $\Delta t = 1/f_s$
- **Dominio delle frequenze:** N è il numero di frequenze digitali
 - ▶ La quantizzazione in k corrisponde a $\Delta f = f_s/N = 1/T$.
 - ▶ Da questo discende che la frequenza minima risolvibile è $1/T$. Ovvero dipende esclusivamente dalla durata temporale e non da N o f_s .
 - ▶ La frequenza di Nyquist è $f_N = f_s/2$, che è anche la frequenza massima risolvibile.
 - ▶ In termini di indici della trasformata la frequenza di Nyquist corrisponde a: $k_N = N/2$
 - ▶ Quindi in realtà il numero di frequenze digitali è $N/2$ non N .

Esempio: DFT di un coseno

$$f_n = \cos\left(\frac{2\pi k'}{N}n\right) = \frac{e^{i\frac{2\pi k'}{N}n} + e^{-i\frac{2\pi k'}{N}n}}{2}$$

scomposizione in
frequenze
“positive” e “negative”

$$X_k = \sum_n^{N-1} \cos\left(\frac{2\pi k'}{N}n\right) e^{-i\frac{2\pi k}{N}n}$$

$$X_k = \sum_n^{N-1} \frac{e^{-i\frac{2\pi n}{N}(k-k')} + e^{-i\frac{2\pi n}{N}(k+k')}}{2}$$

$$X_k = \begin{cases} N/2, & \text{if } k = k' \\ N/2, & \text{if } k = N - k' \\ 0, & \text{otherwise} \end{cases}$$

frequenza “positiva”

frequenza “negativa”

Frequenze positive e negative

- Data la DFT

$$X_k = \sum_{n=0}^{N-1} f_n e^{-i \frac{2\pi}{N} kn}$$

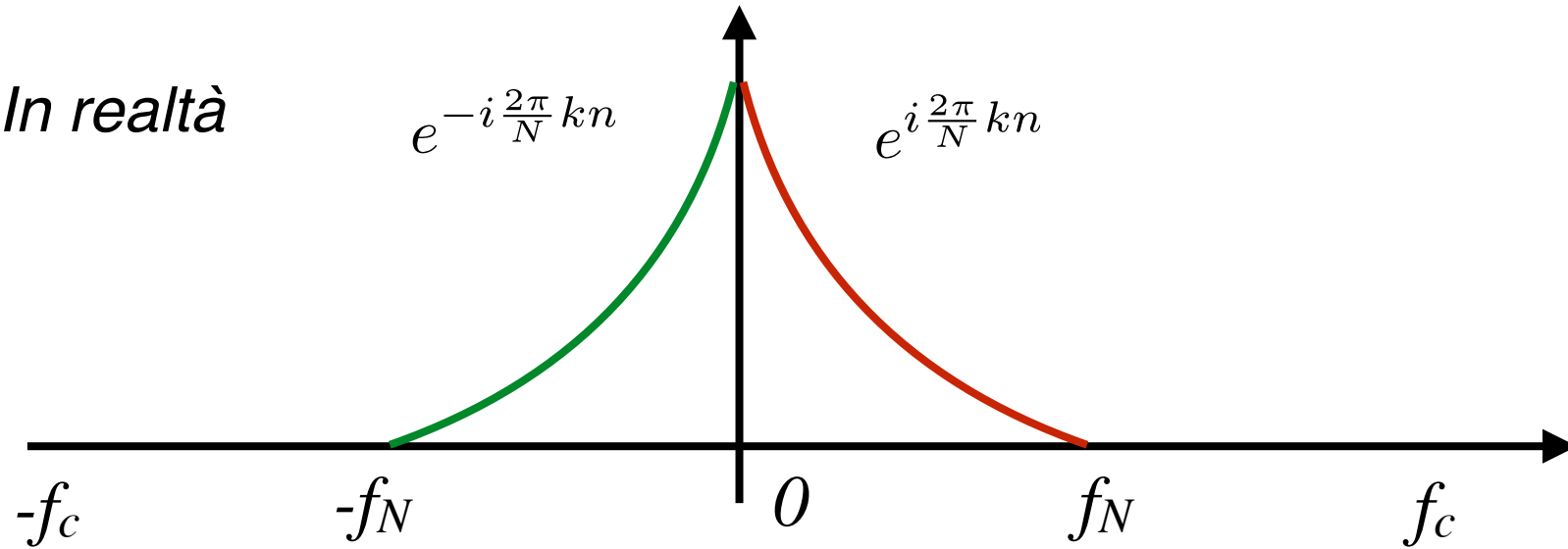
- si può dimostrare facilmente che nel caso di f_n reale:

$$X_k = X_{N-k}^* \quad \text{for } 0 < k < N/2$$

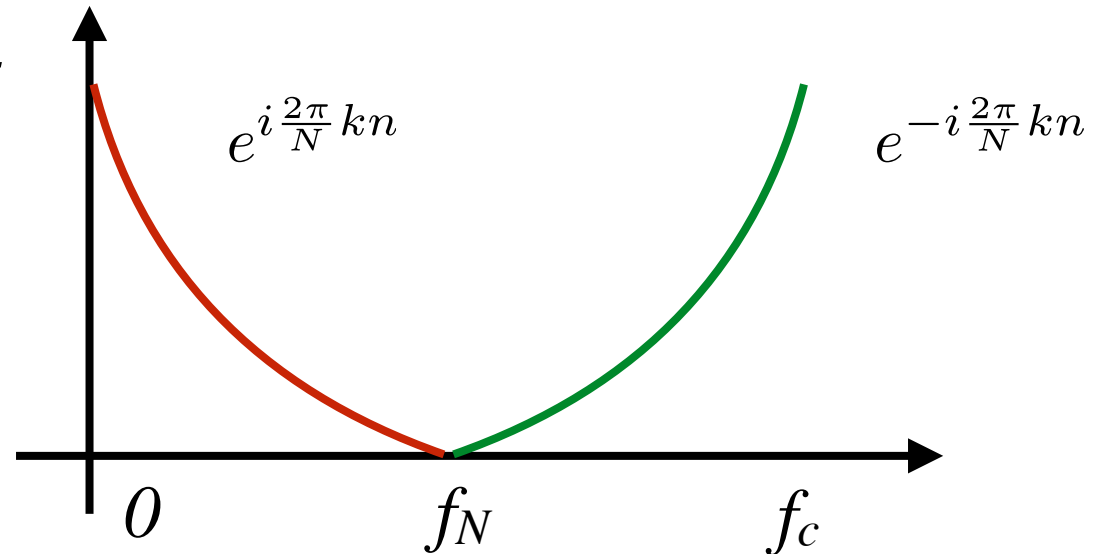
- Pertanto gli elementi con $k > N/2$ corrispondono alle “frequenze negative” e il loro valore non è indipendente da quelle positive.
 - ▶ Rappresentano il contributo $e^{-i \frac{2\pi}{N} kn}$ che insieme a $e^{i \frac{2\pi}{N} kn}$ costituisce la base per fare onde a fase variabile.
 - ▶ Invece di essere distribuite in $[-N/2+1, -1]$ si trovano in $[N/2+1, N-1]$
- Una conseguenza pratica dell’equazione precedente è: $|X_k| = |X_{N-k}|$

Frequenze positive e negative

In realtà

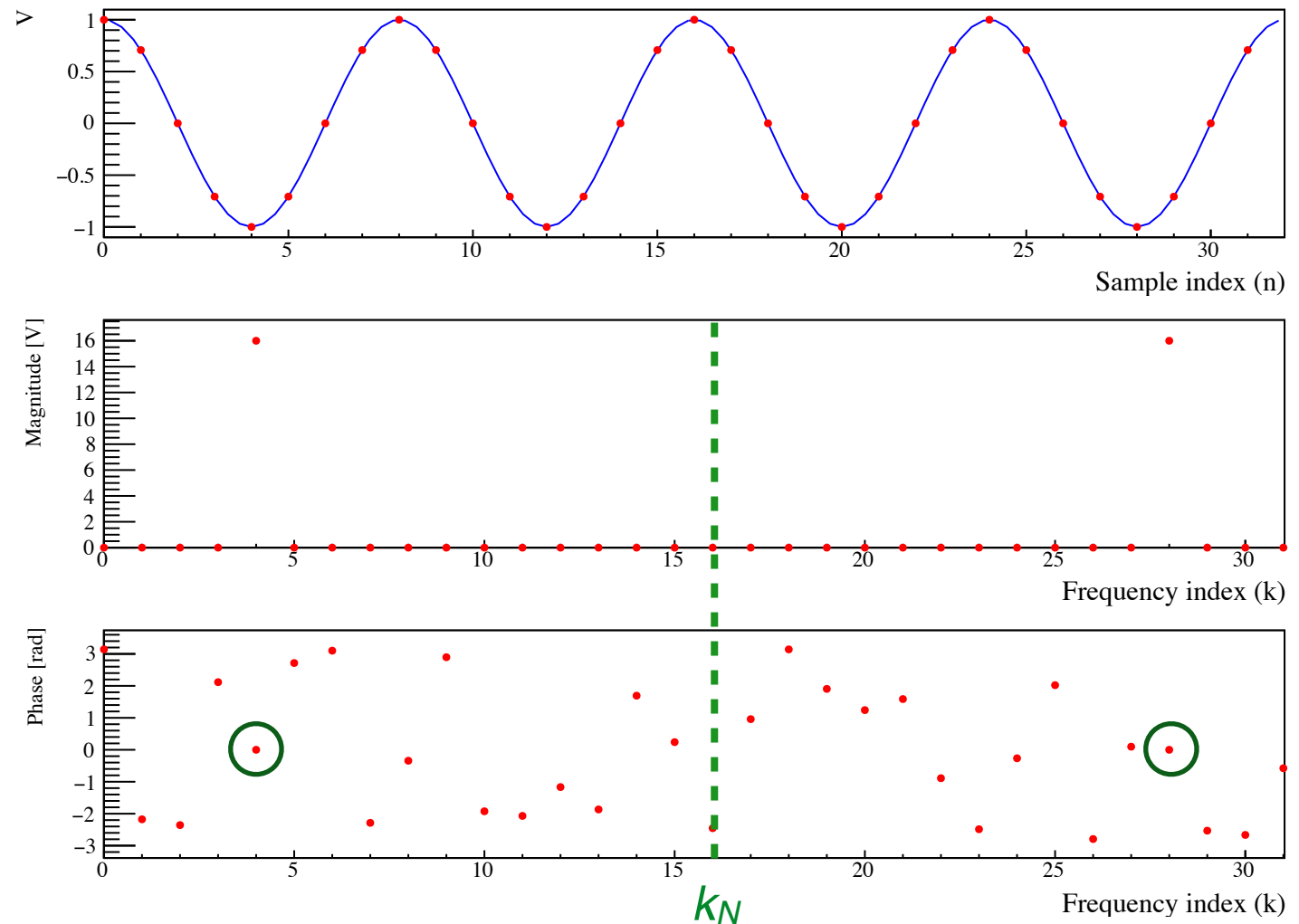


Rappresentazione DFT



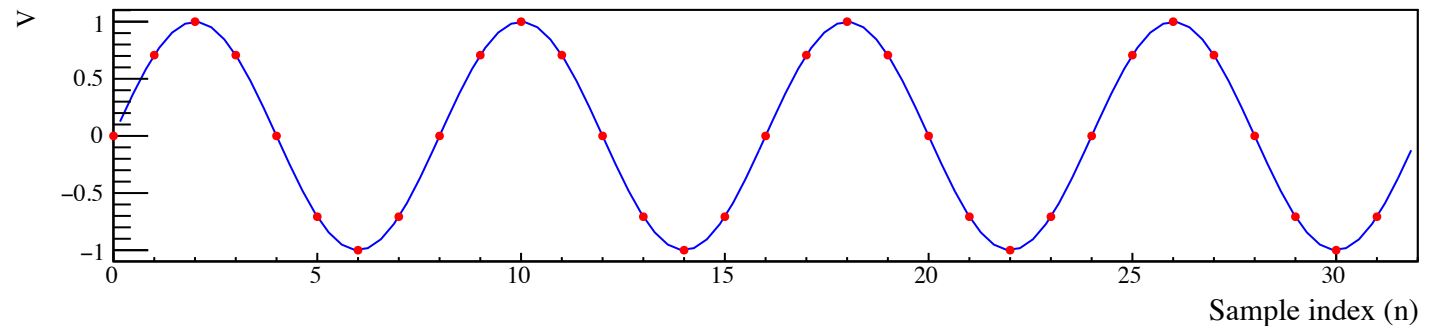
DFT di un coseno ($N = 32, k' = 4$)

$$\frac{e^{+i\frac{2\pi n}{N}k'} + e^{-i\frac{2\pi n}{N}k'}}{2}$$

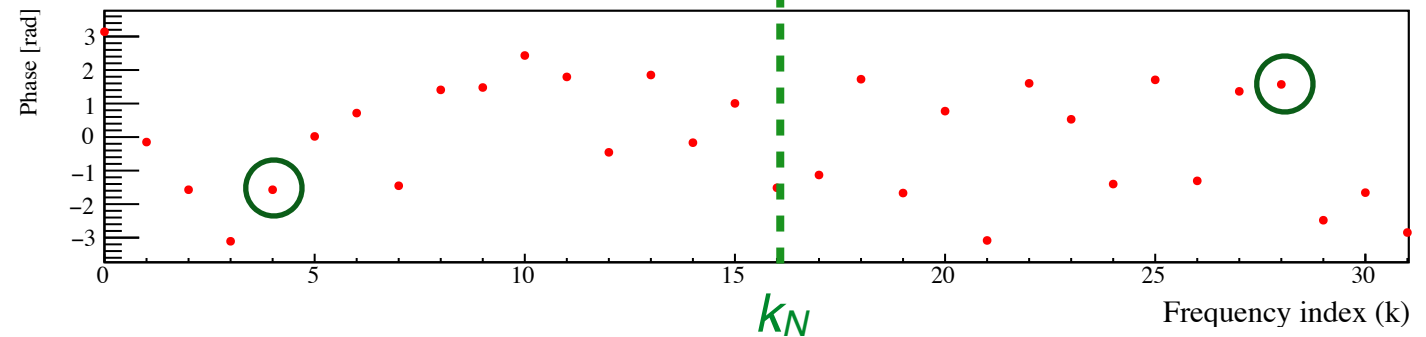
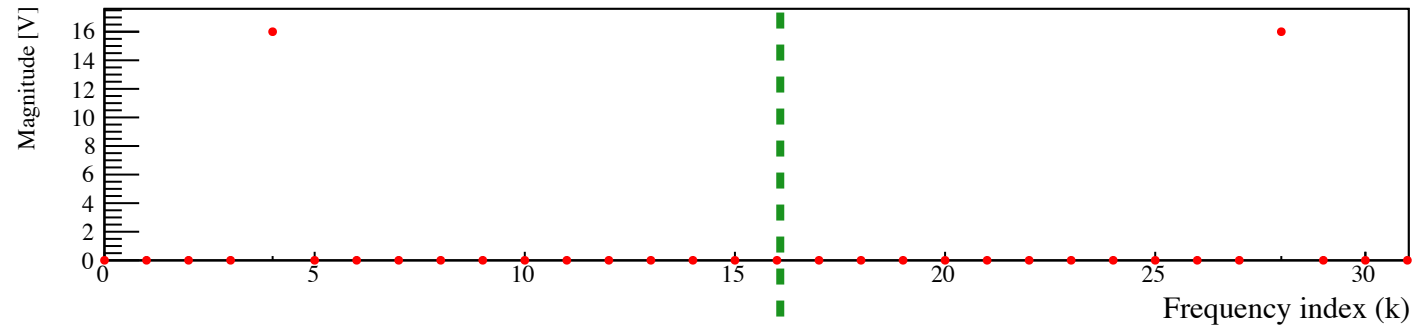


- Le fasi non sono valutabili quando Magnitude = 0, quindi il calcolo con il computer produce un numero a caso.
- Fasi uguali a zero per $k=4$ e $k=32-4$

DFT di un seno ($N = 32, k' = 4$)



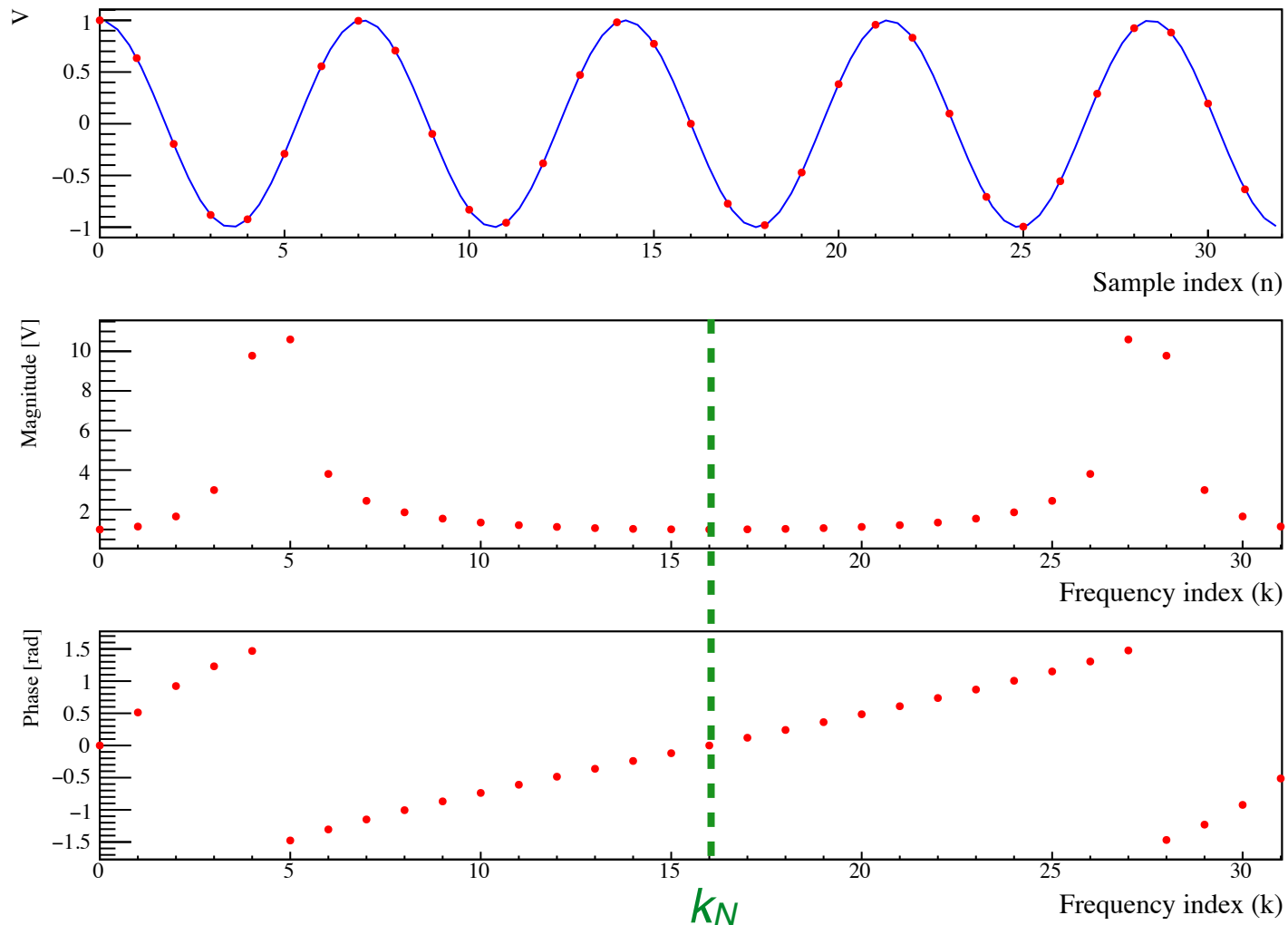
$$\frac{e^{+i\frac{2\pi n}{N}k'} - e^{-i\frac{2\pi n}{N}k'}}{2i}$$



- Le fasi non sono valutabili quando Magnitude = 0, quindi il calcolo produce un numero a caso.
- Fasi uguali a $-\pi/2$ per $k=4$ e $+\pi/2$ per $k=32-4$

DFT di un coseno ($N = 32, k' = 4.5$)

- Nella realtà è improbabile che $k' = f / \Delta f$ sia esattamente un intero. k' descrive spesso un segnale analogico che non ha nulla a che vedere con la struttura di frequenza del nostro ADC.



- La DFT descrive l'onda come sovrapposizione di contributi a k intero.

Windowing

- Per ovviare al problema della risposta in k non uniforme si può applicare ai dati da trasformare una funzione ammorbidente, detta di “Window”.

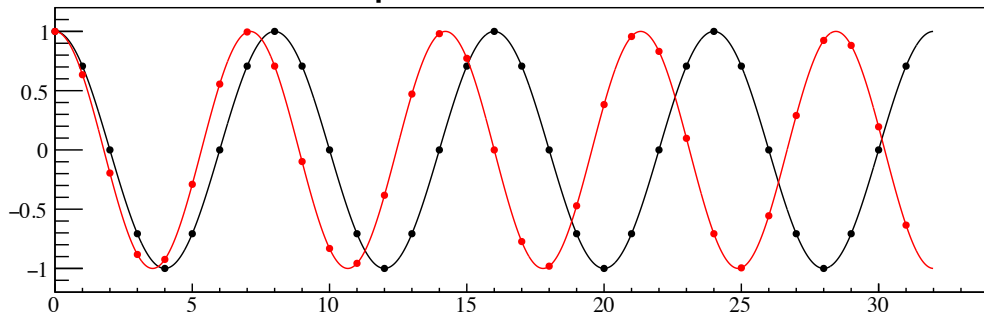
▶ Ne esistono di vari tipi: https://en.wikipedia.org/wiki/Window_function

▶ Qui mostriamo l'effetto della Blackman-Harris

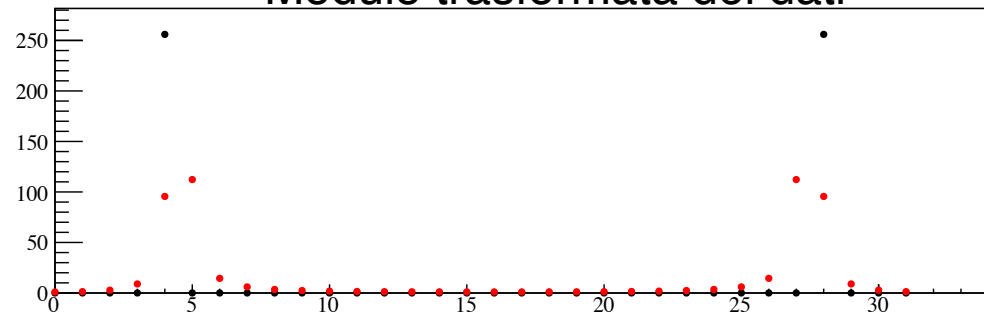
$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right) - a_3 \cos\left(\frac{6\pi n}{N-1}\right)$$

$$a_0 = 0.35875; \quad a_1 = 0.48829; \quad a_2 = 0.14128; \quad a_3 = 0.01168$$

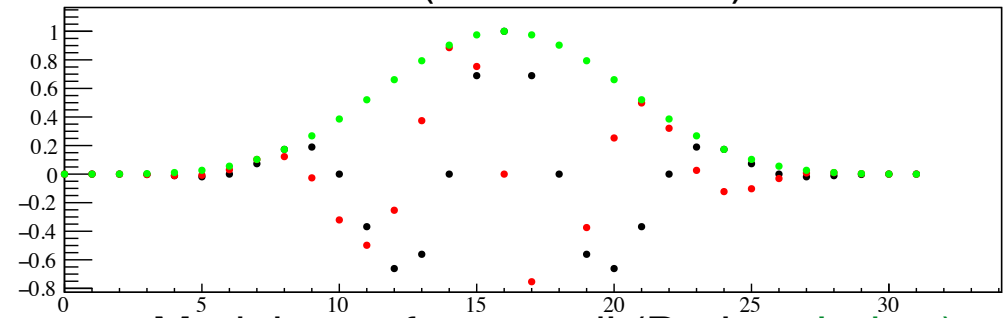
Dati campionati e funzioni ideali



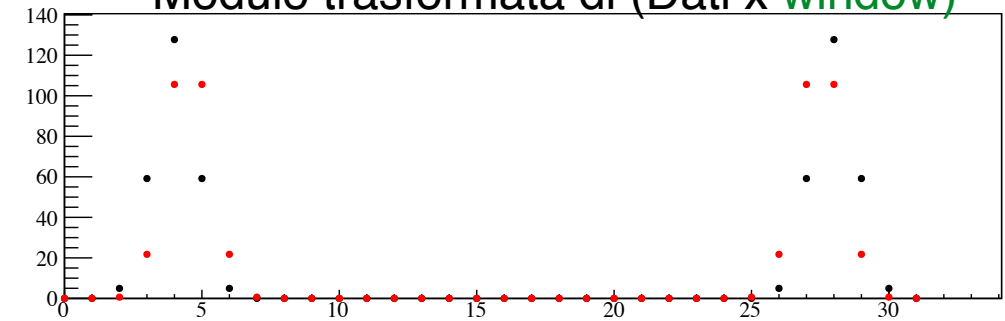
Modulo trasformata dei dati



(Dati x window)



Modulo trasformata di (Dati x window)



Normalizzazione

- Riprendiamo la trasformata del coseno di ampiezza unitaria

$$f_n = \cos\left(\frac{2\pi k'}{N}n\right) = \frac{e^{i\frac{2\pi k'}{N}n} + e^{-i\frac{2\pi k'}{N}n}}{2}$$

$$X_k = \begin{cases} N/2, & \text{if } k = k' \\ N/2, & \text{if } k = N - k' \\ 0, & \text{otherwise} \end{cases}$$

- Se vogliamo dare un significato fisico alla trasformata, cioè quello di distribuzione di ampiezza di onde a diversa frequenza,
 - ▶ Dobbiamo dividere X_k per N

Frequenza zero

- Dalla definizione della trasformata

$$X_k = \sum_{n=0}^{N-1} f_n e^{-i \frac{2\pi}{N} kn}$$

- Discende che la frequenza $k = 0$ corrisponde all'integrale del segnale

$$X_0 = \sum_n^{N-1} f_n$$

- Tutte le frequenze al disotto di $k = 1$, (cioè $f < \Delta f$), finiscono nell'integrale e sono irrisolvibili.
 - ▶ Se la DFT è normalizzata, X_0 diventa la **media**.
- X_0 è sempre reale, così come $X_{N/2}$

Potenza

- Un segnale (co)sinusoidale di ampiezza in tensione unitaria ($V=1$) genera su una resistenza unitaria ($R=1$) una potenza pari a:

$$P = \frac{\langle V^2 \rangle}{R} = \frac{1}{2}$$

- Calcolando la potenza dalla DFT del coseno precedentemente trovata

$$\sum_{k=0}^{N-1} |X_k|^2 = \frac{N^2}{4} + \frac{N^2}{4} = \frac{N^2}{2}$$

- Normalizzando la DFT:

$$\sum_{k=0}^{N-1} \frac{|X_k|^2}{N^2} = \frac{1}{2}$$

- Ovvero quello che ci aspettavamo: l'integrale del quadrato nel FD (frequency domain) è uguale a quello del TD (identità di Parseval).

Spettro di potenza (PSD)

- A partire dalla sommatoria che ci dà la potenza

$$\sum_{k=0}^{N-1} \frac{|X_k|^2}{N^2} = \frac{1}{2}$$

- Possiamo identificare le componenti spettrali:

$$PSD_k = \frac{|X_k|^2}{N^2} \frac{1}{\Delta f} = \frac{|X_k|^2}{N f_s}$$

- ▶ Dove abbiamo diviso per la risoluzione in frequenza (quanto è largo un bin in frequenza) dato che la PSD si esprime in $[V^2/Hz]$

PSD (2)

- Dato che le frequenze al di sopra di $k=N/2$ replicano le precedenti, e dato che $k=0$ e $k=N/2$ non sono replicate, le formule da utilizzare sono:

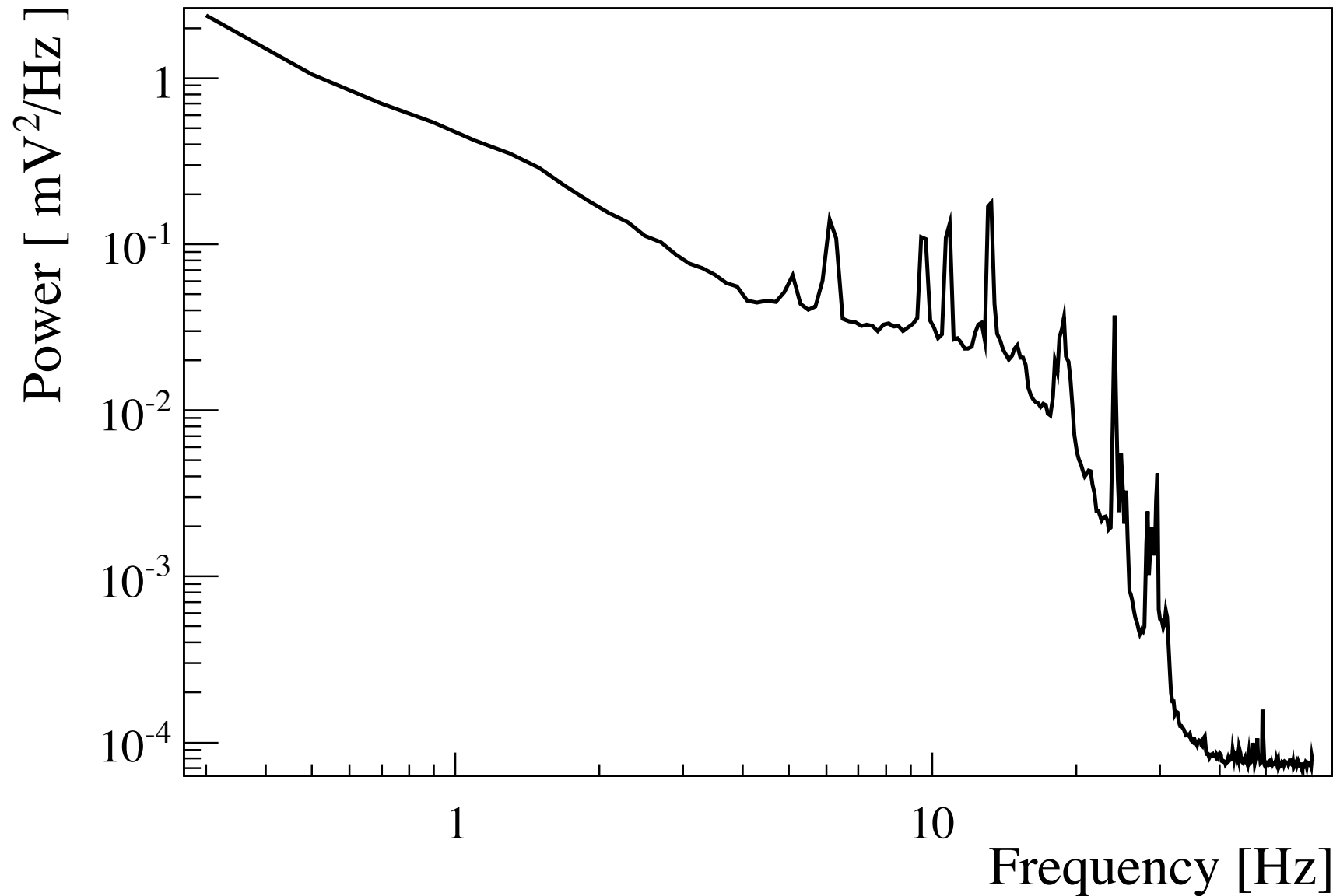
$$PSD_k = 2 \frac{|X_k|^2}{N f_s} \quad 0 < k < N/2$$

$$PSD_k = \frac{|X_k|^2}{N f_s} \quad k = 0, k = N/2$$

- Inoltre i dati vanno convertiti da conteggi a Volt. Per questo vanno moltiplicati per dV^2 .
- Infine nel grafico riportate in ascissa la frequenza fisica ($f = k \Delta f$):

$$PSD(f = k\Delta f) [V^2/Hz] = PSD_k \cdot dV^2$$

Esempio di noise PSD da DFT



- Noise power spectrum of a cryogenic detector.

Esperienza 9

SW per Arduino e processing

- Scaricate e installate i programmi dal sito di Vignati o Raggi.
 - `adc_read_5_2017.ino` (Arduino)
 - `adc_arduino_5.pde` (Processing)
- Configurate la porta di Arduino sia quella giusta (Strumenti -> Porta)
- Processing da lo Start di acquisizione ad Arduino.

Segnali sinusoidali

- Misurate con Arduino e Processing da soli lo spettro di una sinusoide generata col generatore di segnali.
 - ▶ Collegate il BNC del generatore alla breadboard.
 - ▶ Utilizzate un filo per connettere il segnale dalla breadboard al ANALOG IN A3 di Arduino.
 - ▶ Eventualmente collegate il GND della breadboard a quello di Arduino.
- Inizialmente scegliete una frequenza inferiore a quella di Nyquist dell'ADC ($9 \text{ kHz} / 2 = 4.5 \text{ kHz}$).
- Possiamo poi studiare il fenomeno dell'aliasing utilizzando segnali sinusoidali con frequenza superiore a quella di Nyquist.
- Effettuare lo stesso studio con un onda quadra



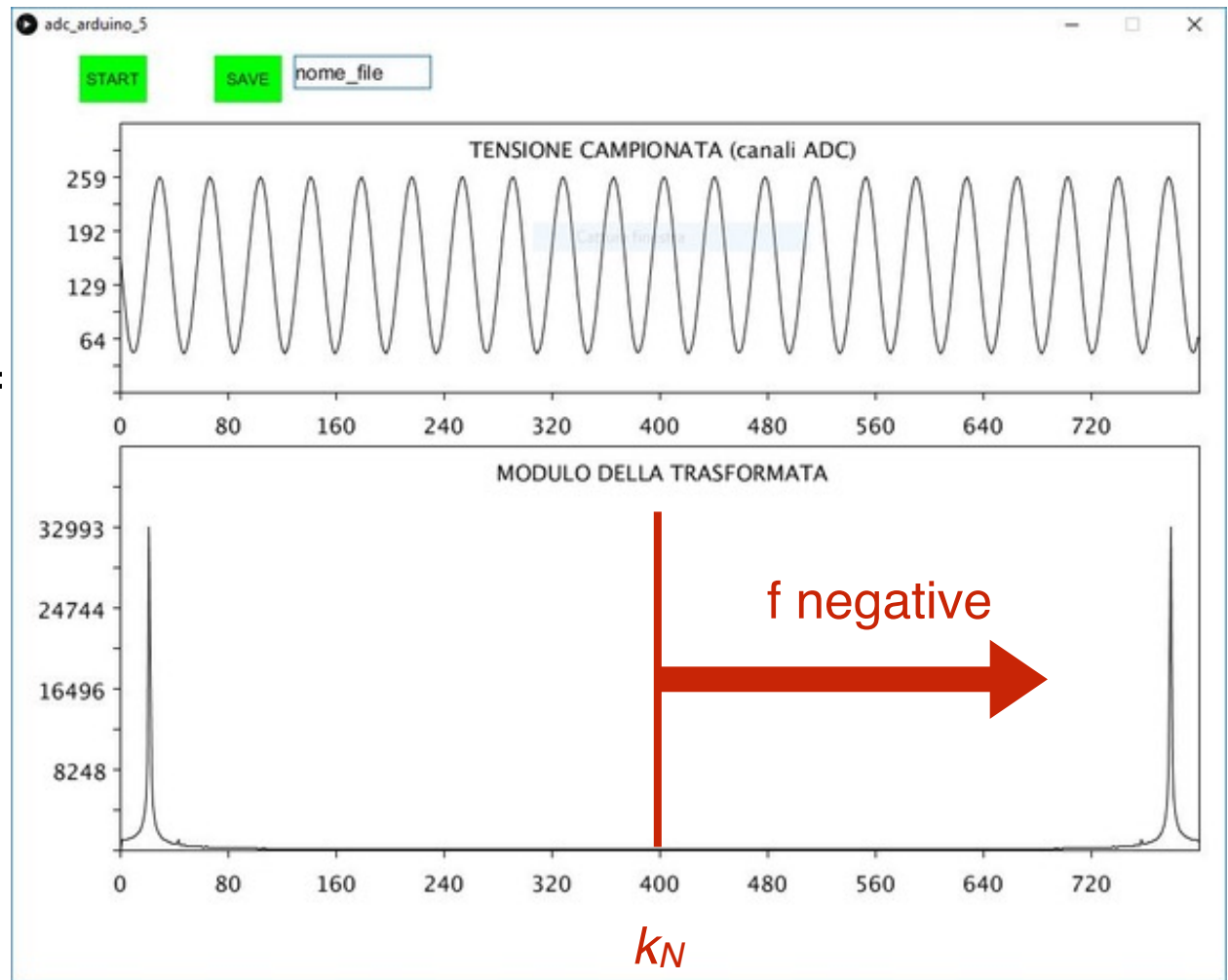
Segnale 200 Hz

- Caratteristiche Arduino:

- ▶ 1024 conteggi, 0-5 Volt,
- ▶ 800 campionamenti a circa 9 kSPS.

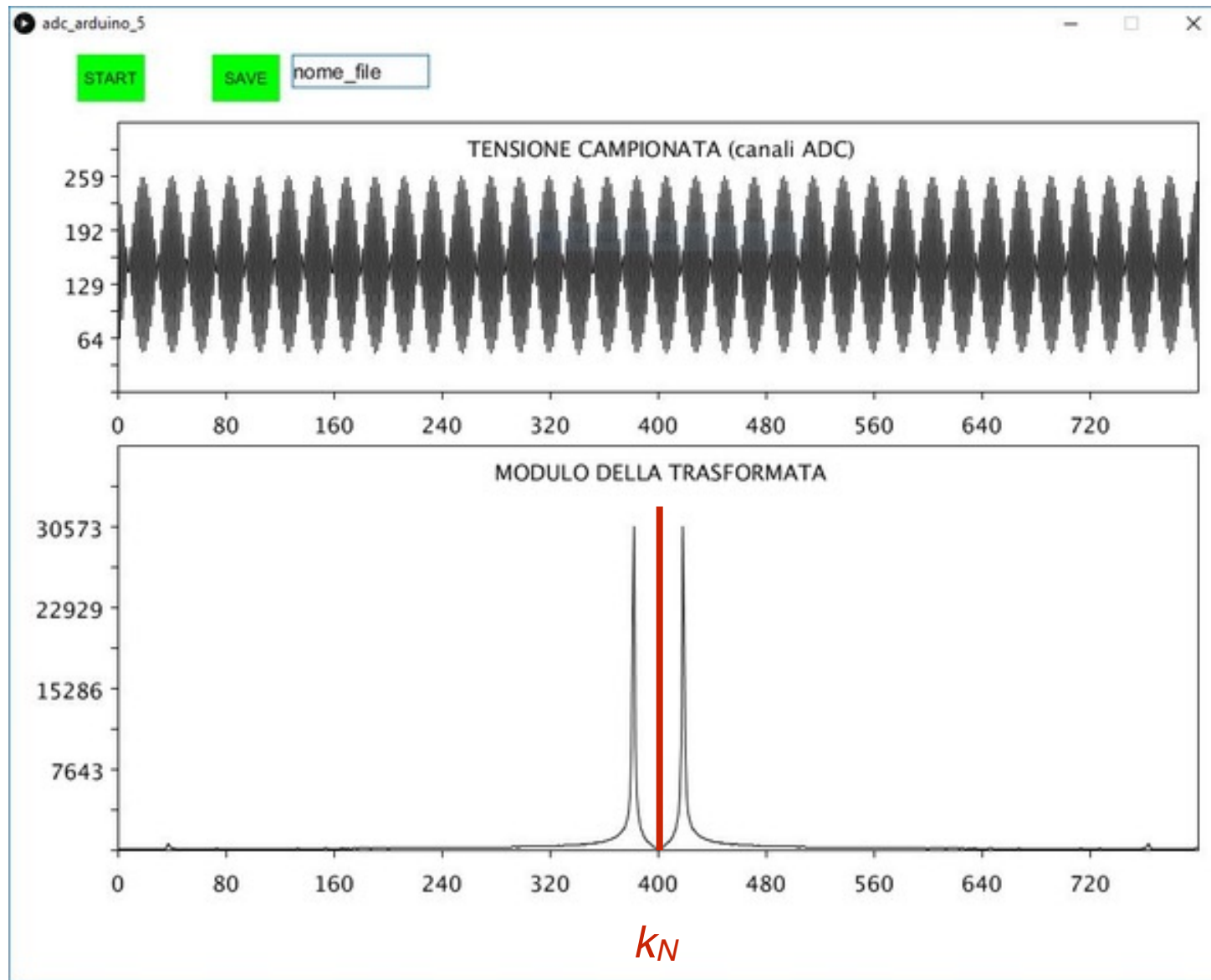
- Pertanto

- ▶ $\text{Volt} / \text{count} = 5 / 1024 = 4.88 \text{ mV} / \text{count}$
- ▶ $N = 800$
- ▶ $f_s = 9 \text{ kHz}$
- ▶ $T = N / f_s = 88.9 \text{ ms}$
- ▶ $\Delta t = 1 / f_s = 111 \text{ us}$
- ▶ $\Delta f = f_s / N = 11.3 \text{ Hz}$
- ▶ $f_N = 4.5 \text{ kHz}$
- ▶ $k_N = 400$



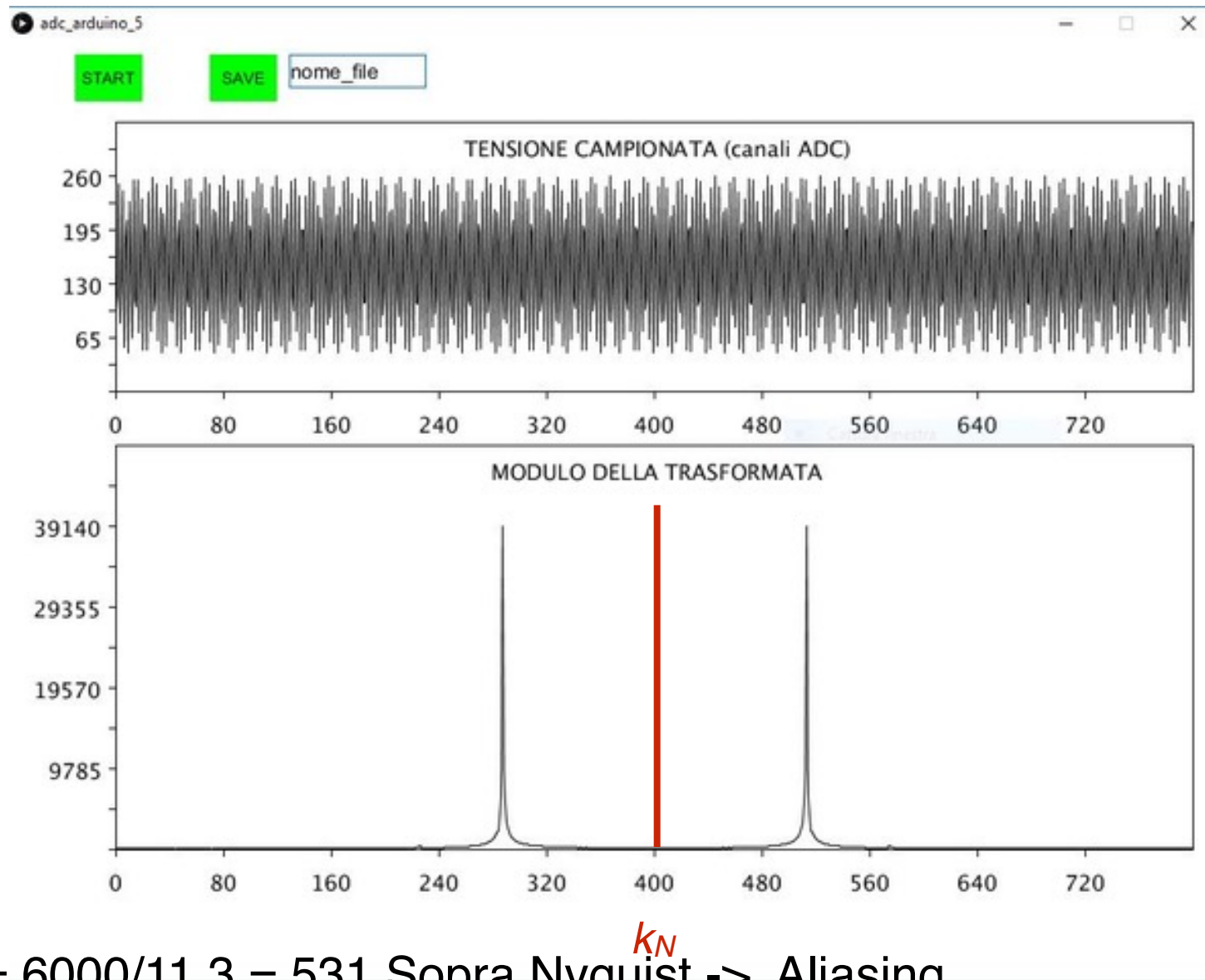
- $k_{\text{segnale}} = 200 \text{ Hz} / 11.3 \text{ Hz} = 17.7$

Segnale 4.2 kHz



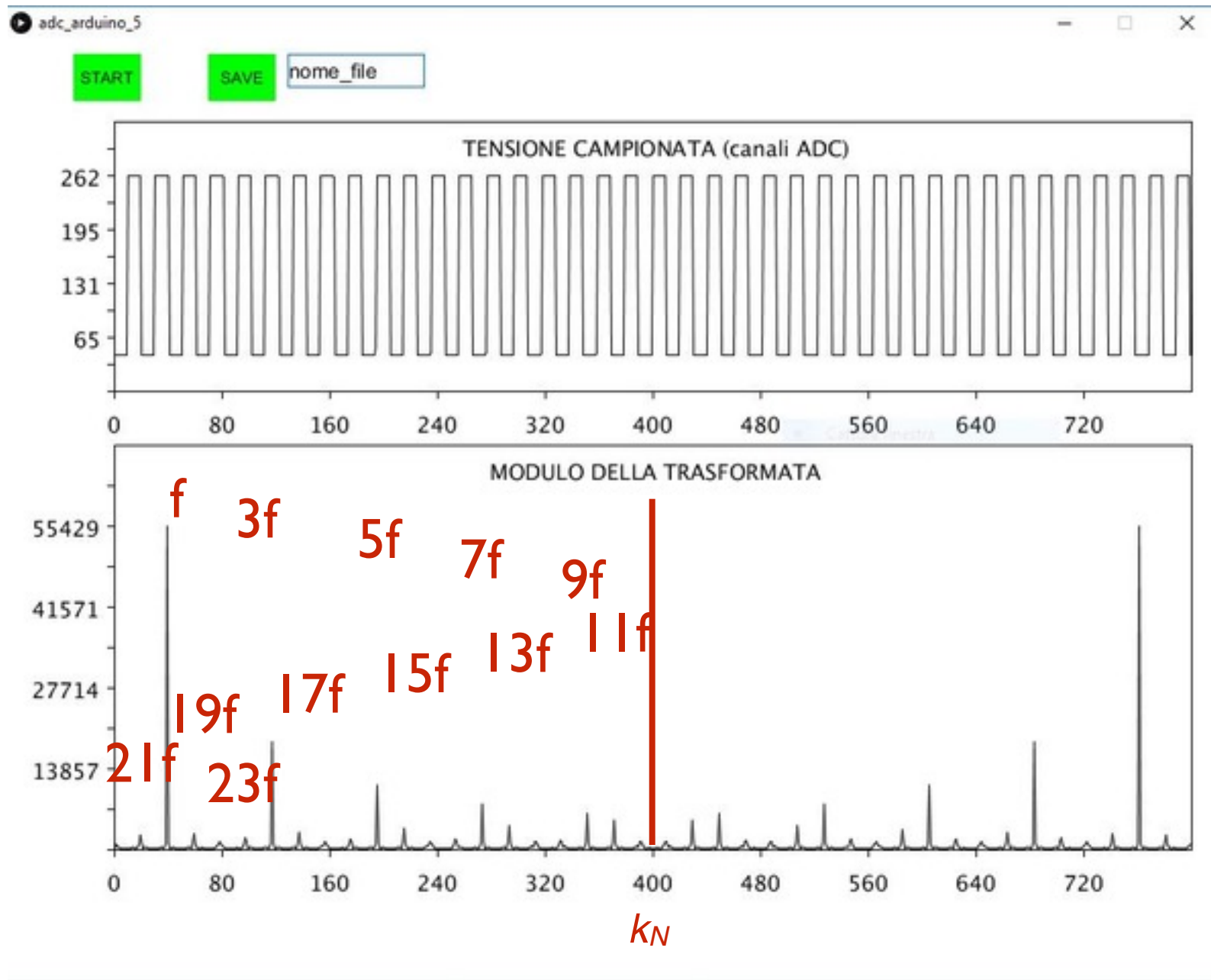
- $k_{\text{segnale}} = 4200/11.3 = 371$

Segnale 6 kHz

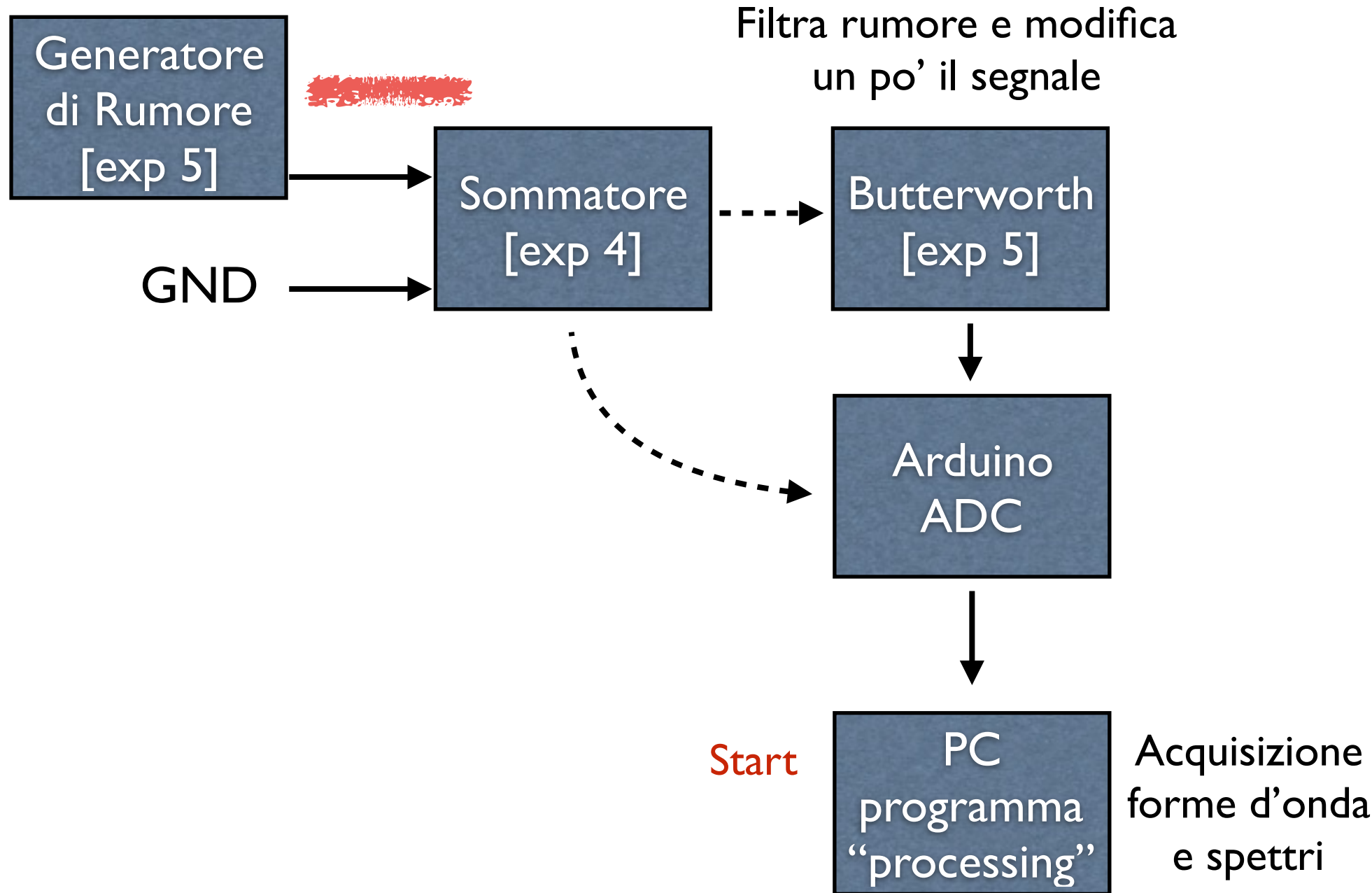


- $k_{\text{segnale}} = 6000/11.3 = 531$ Sopra Nyquist ^{k_N} -> Aliasing
- $k_{\text{apparente}} = 400 - 131 = 269$ (si tratta come un rimbalzo su k_N)

Onda quadra $f = 450$ Hz



Misura Rumore



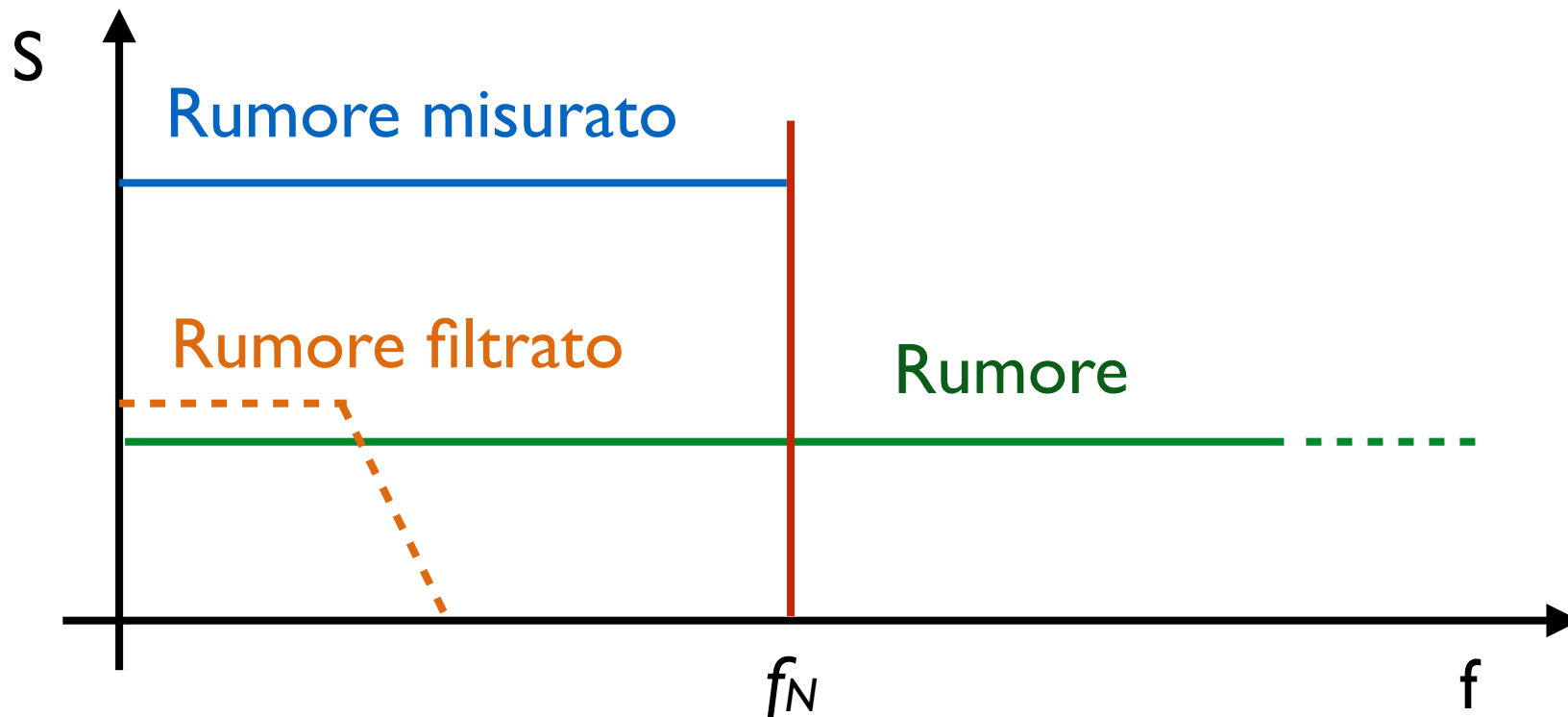
Misura di rumore

- Studiate l'andamento in frequenza del rumore da solo staccando il segnale dal sommatore e mettendolo a massa.
- Data la sua natura casuale è sensato prendere piu' forme d'onda ($M = 5 \div 10$) per poi ricavare lo spettro facendo la media dei moduli quadri della trasformata.

$$S_k = \sqrt{\frac{\sum_{m=0}^{M-1} |X_k|^2}{M}}$$

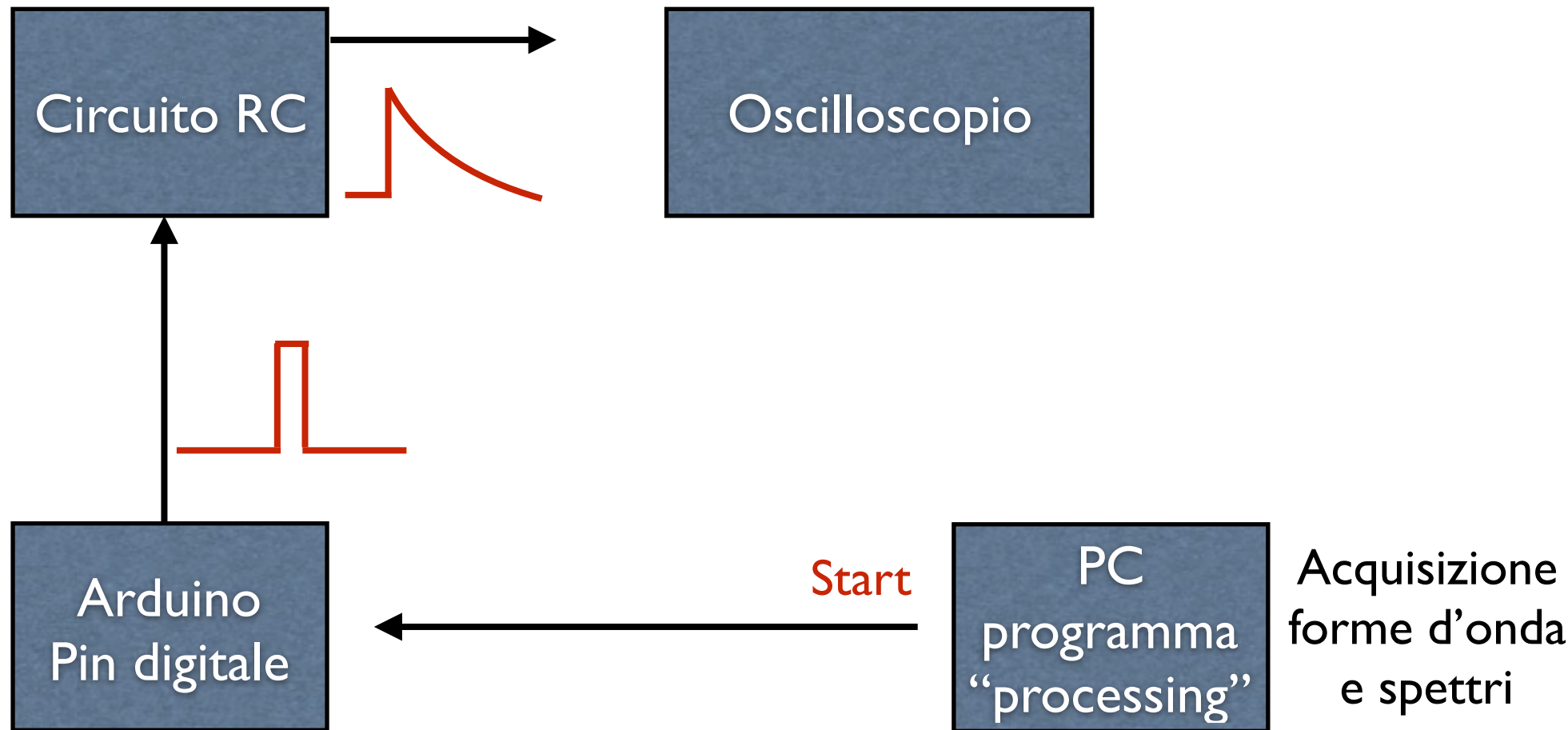
- ▶ Con Processing salvare i dati in formato txt [i, (V(i) Re(i), Im,(i), Mod(i)]
- ▶ Salvare ogni campione m su un file diverso e poi, off-line, fare la media (con un programma in c, oppure con Open Office).
- ▶ Se siete fichi applicate anche le istruzioni di slide "PSD (2)".
- Effettuate questa misura con e senza filtro passa basso. Ricordatevi nel confronto che il filtro ha un guadagno.

Cosa aspettarsi



- Perché il rumore misurato ha un livello più alto di quello originale?
- Se avete tempo convertite i grafici da indice k a frequenza, in questo modo riconoscerete facilmente il taglio del Butterworth.

Creazione di un segnale impulsivo



- Controllare il segnale in uscita dal pin digitale di Arduino.
- Controllare il segnale in uscita dal circuito RC ($10 \text{ k}\Omega \times 1 \text{ uF} = 10 \text{ ms}$)
- Se i segnali sono brutti probabilmente non avete connesso i GND.

Forma attesa del segnale impulsivo

- Da Arduino esce un'onda quadra molto stretta, circa $\Delta T = 2$ ms, con ritardo di circa 10 ms rispetto allo start dell'acquisizione ADC.
- Possiamo rappresentarlo come la sovrapposizione di 2 onde quadre e applicare Laplace:

$$f(t) = A[\theta(t) - \theta(t - \Delta t)] \quad f(s) = A \left[\frac{1}{s} - \frac{1}{s} e^{-s\Delta t} \right]$$

- Moltiplicando per la funzione di trasferimento del passa basso.

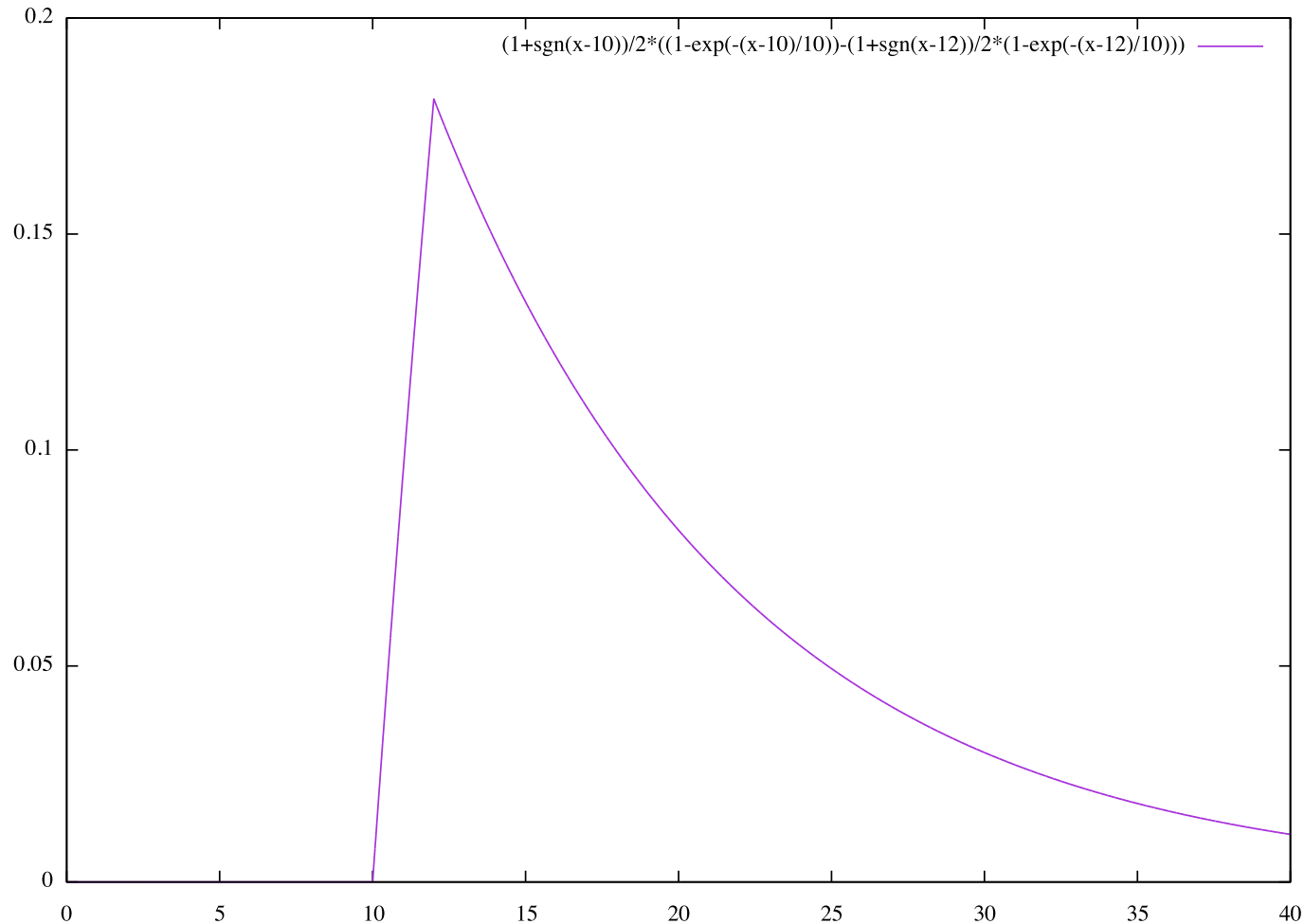
$$P(s) = f(s) \frac{1}{1 + s\tau}$$

$$P(t) = A\theta(t)[1 - e^{-t/\tau}] - A\theta(t - \Delta t)[1 - e^{-(t-\Delta t)/\tau}]$$

- Quindi in pratica l'impulso è:

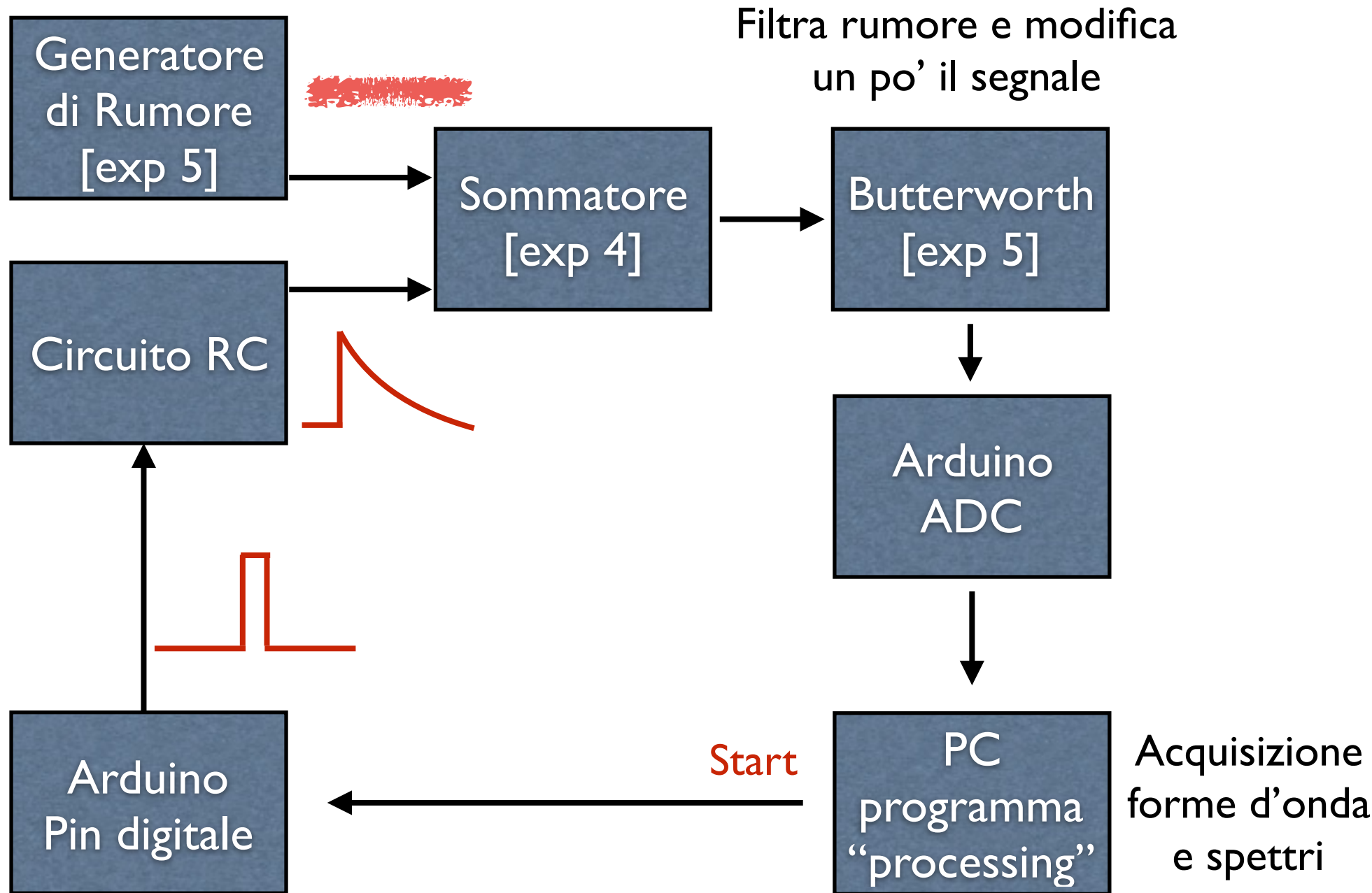
$$P(t) = \begin{cases} A[1 - e^{-t/\tau}], & \text{if } t < \Delta t \\ A[e^{\Delta t/\tau} - 1]e^{-t/\tau}, & \text{if } t \geq \Delta t \end{cases}$$

Segnale impulsivo simulato



- Il massimo vale $A [1-e^{-t/\Delta t}] = 0.18 \text{ A}$

Circuito completo



Misure segnale e noise

- Analogamente a quanto fatto per il rumore, studiate lo spettro in frequenza del segnale da solo staccando il rumore dal sommatore e mettendolo a massa.
 - ▶ Effettuate questa misura con e senza filtro passa basso.
- Attaccate segnale e rumore al sommatore.
 - ▶ Controllate l'uscita del sommatore con l'oscilloscopio.
 - ▶ Acquisite le forme d'onda nel dominio tempo con Arduino.
 - ▶ Effettuate questa misura con e senza filtro passa basso.
 - ▶ Ammirate e traete conclusioni.