

Corso di Laurea in Fisica

Esame di Laboratorio II – I Modulo

11/01/2021

Abstract

Si risolva il seguente esercizio, scrivendo un programma in C++. Ai fini della valutazione, il primo criterio che deve essere soddisfatto è che il codice compili senza errori ed esegua realizzando le funzionalità richieste dal testo. Per la valutazione sarà inoltre tenuto in considerazione il fatto che i codici siano scritti con ordine, utilizzando opportunamente l'**indentazione** e i **commenti**. Si richiede infine di iniziare i codici con una riga di commento contenente il comando necessario per creare l'eseguibile.

Classe Impulso

Un segnale digitalizzato ha la forma mostrata in figura: è costituito da un numero di campionamenti N_{points} in corrispondenza dei quali viene registrato un numero reale che rappresenta l'altezza del segnale. I primi n_{pretr} punti non contengono ancora l'impulso (il segnale è nullo) e si chiamano pre-trigger.

In figura sono riportati, a sinistra, la forma ideale del segnale (chiamato modello o **template**, con ampiezza unitaria); a destra la forma di un **segnale reale**: cambia l'ampiezza e compare il rumore (una fluttuazione statistica del valore del segnale).

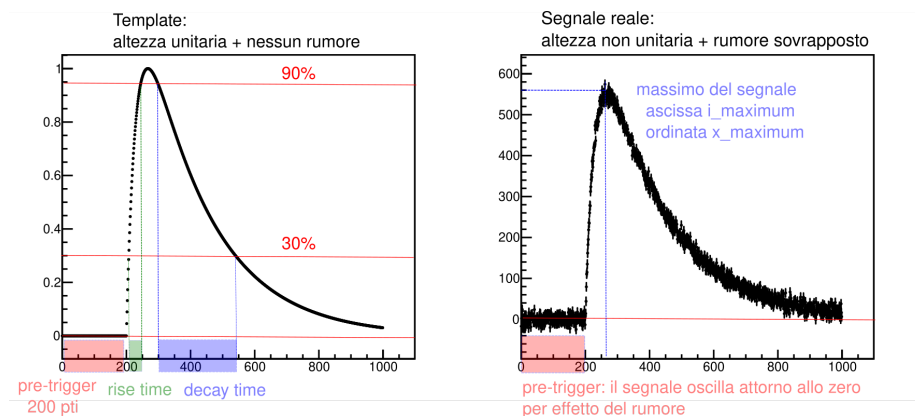


Figure 1: A sinistra il template di un segnale: rappresenta la forma del segnale in condizioni ideali (nessun rumore sovrapposto ed ampiezza unitaria). A destra un segnale reale. In entrambi i casi il segnale è nullo nella regione del pretrigger, i valori diversi da zero per il segnale reale sono dovuti al rumore.

Il segnale reale è caratterizzato da:

- un'ampiezza che nel caso dell'esercizio è il **massimo** raggiunto dall'impulso;
- un **rise-time** che corrisponde al numero di punti che servono al segnale per passare, quando sta crescendo, dal 30% al 90% della sua altezza;
- un **decay-time** che corrisponde al numero di punti che servono al segnale per passare, quando sta decrescendo, dal 90% al 30% della sua altezza;
- il valore **RMS del rumore** che viene stimato calcolando lo scarto quadratico medio rispetto a zero dei punti che appartengono alla regione di pre-trigger (in questa regione il segnale è nullo e si osserva solo il rumore).

Nel caso in esame i dati relativi ad un segnale sono registrati in un file di testo `pulse.txt` con il seguente formato: le prime due righe riportano, rispettivamente, il numero di punti campionati N_{points} e il numero di punti n_{pretr} nella regione del pretrigger. Dalla terza riga in poi (per N_{points} righe) sono riportati i numeri reali che corrispondono ai valori di tensione campionati.

Il file `template.txt` contiene, nello stesso formato, il template del segnale.

Si chiede di implementare una classe `impulso` per gestire questi dati, usando il prototipo `impulso.h` messo a disposizione nella pagina dell'esame. La classe deve avere:

- tre variabili private: `int Npoints_p`, `int npretr_p`,
`vector<double> campionamenti_p`;
- un costruttore che riempie le variabili con i dati letti in un file di testo.

La classe deve inoltre implementare i seguenti metodi:

`GetPoint()` restituisce il valore di un elemento di `campionamenti_p`;
`GetMaximum()` scrive, in due variabili passate per referenza, la posizione ed il valore dell'elemento maggiore in `campionamenti_p` (massimo del segnale);
`Maximum()` restituisce il valore dell'elemento maggiore dell'array, come calcolato con il metodo precedente;
`RMSNoise()` restituisce il valore del rumore, calcolato come RMS dei primi `npretr` punti di `campionamenti_p` (sono quelli che ancora non contengono segnale);
`GetRiseTime()` calcola la distanza – in numero di punti – che intercorre tra il 30% e il 90% del massimo del segnale, sulla porzione di segnale a sinistra del massimo
`GetDecayTime()` calcola la distanza – in numero di punti – che intercorre tra il 90% e il 30% del massimo del segnale, sulla porzione di segnale a destra del massimo
`Stampa()` stampa a schermo le quantità valutate dei metodi precedenti.
`Confronta()` effettua un test di compatibilità tra il segnale e una forma di riferimento usando il metodo del chi-quadro. Il metodo prevede che la forma di riferimento sia anch'essa un'oggetto di tipo `impulso`. L'obiettivo del test è confrontare la forma dei due segnali, quindi il metodo `Confronta` deve prima di tutto riscalarlo il template riferimento in modo che abbia la stessa altezza del segnale. Per calcolare il chi-quadro si assuma che il template sia priva di errore mentre per i punti del segnale si utilizzi come deviazione standard il valore `RMSNoise`.

Si scriva un programma principale in cui:

- sono istanziati due oggetti di tipo `impulso` a partire dai file `pulse.txt` e `template.txt`.
- Si stampano a schermo le caratteristiche dei due oggetti, utilizzando il loro metodo `Stampa`.
- Si riempiono due `TGraphErrors` e si disegnano i due segnali su due Canvas differenti, con in ascissa il numero di punti, senza errore, in ordinata il valore del campionamento, con errore pari a `RMSNoise`.
- Si calcola il fattore di normalizzazione del segnale al template usando la stima dell'ampiezza fornita da `Maximum` e si confronta il segnale letto da `pulse.txt` con il template letto da `template.txt`.