

Digitalizzazione e analisi di segnali di rivelatori

Luigi Bardelli
bardelli@fi.infn.it
8 Marzo 2010

Indice

1	Introduzione	3
2	La conversione analogico-digitale veloce	3
2.1	Principi base	3
2.2	Metodi di conversione	4
3	Aliasing e interpolazione	6
4	Rumore introdotto dal processo di conversione	9
4.1	Bit, ENOB, rumore di quantizzazione	10
4.2	Cenni al problema del random sampling	11
4.3	Altri effetti	12
5	Applicazioni in Fisica Nucleare e SubNucleare	12
5.1	Confronto con i metodi analogici	13
5.2	Misure di energia e filtri digitali	14
5.3	Misure di temporizzazione	15
5.4	Analisi di forma	16
6	Acquisizione e analisi dati in laboratorio	17
6.1	L'oscilloscopio digitale	17
6.2	Apparato sperimentale	19
6.3	Il programma di acquisizione e analisi	20
6.4	Acquisizione dei segnali	21
6.5	Analisi dati	21
6.6	Il fit dei segnali	22
6.7	Elenco delle funzioni per il fit di segnali	25
6.7.1	Funzione delta con durata finita	26
6.7.2	Funzione gaussiana	27
6.7.3	Funzione gaussiana·exp	29
6.7.4	Funzione $\exp(-t/\tau)$	30
6.7.5	Funzione $t \exp(-t/\tau)$	31
6.7.6	Funzione $A_1 \exp(-t/\tau_1) + A_2 \exp(-t/\tau_2)$	32
6.7.7	Funzione $A_1 t \exp(-t/\tau_1) + A_2 \exp(-t/\tau_2)$	34
6.7.8	Funzione Scintillatore a 3 componenti	35
6.7.9	Funzione $1/(1 + t/\tau)$	37
6.7.10	Funzione "silicio"	38
6.7.11	Funzione "germanio coassiale"	40
6.8	Altre funzionalità	42
7	Approfondimenti e bibliografia	43

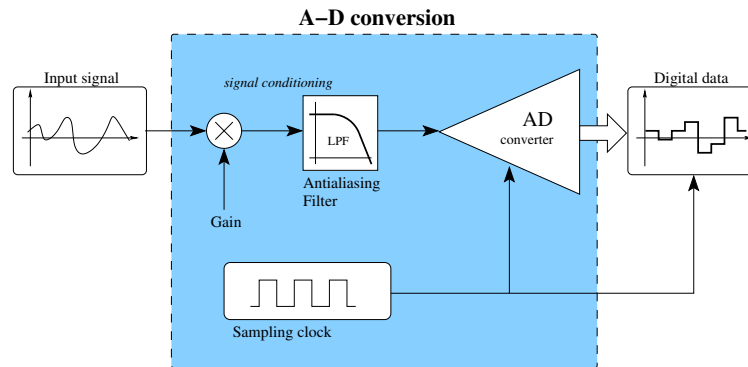


Figura 1: Schematizzazione del processo di conversione analogico-digitale

1 Introduzione

Nel corso degli ultimi anni si sta assistendo ad un notevole interesse nell'ambito della Fisica Nucleare e SubNucleare sperimentali per l'utilizzo di campionatori analogico-digitale veloci.

In queste pagine viene fornita una breve introduzione ai concetti fondamentali del campionamento analogico-digitale veloce: caratteristiche degli ADC (Sect. 2.2), aliasing (Sect. 3), varie sorgenti di rumore dovute al processo di campionamento (Sect. 4), applicazioni in misure sperimentali (Sect. 5). Nel corso del laboratorio verrà utilizzato un oscilloscopio digitale per campionare e analizzare i segnali prodotti da vari rivelatori: in Sect. 6 vengono discusse le relative tematiche.

È da notare che, mentre lo studio e la caratterizzazione delle tecniche di misura analogiche è storicamente ben consolidato (sia sperimentalmente che teoricamente), lo studio dell'applicabilità e delle prestazioni di ADC veloci in misure di Fisica Nucleare e SubNucleare è ancora oggetto di una fiorente attività di ricerca. Alcune delle tematiche presentate in queste pagine sono tuttora dibattute in letteratura: queste sezioni sono state contrassegnate col simbolo "(*)".

2 La conversione analogico-digitale veloce

In queste pagine viene data una breve descrizione delle caratteristiche dei campionatori analogico-digitali. Per approfondimenti, vedi [1, 2].

2.1 Principi base

La digitalizzazione di segnali (vedi Fig. 1) è il processo per cui a un segnale analogico $S(t)$ (variabile con continuità in funzione del tempo) viene associato un vettore $S[k]$ di valori numerici, che corrispondono

	SIGMA DELTA	SAR	PIPELINE	FLASH (Parallel)
Pick This Architecture if you want:	High resolution, low to medium speed, no precision external components, simultaneous 50/60Hz rejection, digital filter reduces anti-aliasing requirements.	Medium to high resolution (8 to 16bit), 5Msps and under, low power, small size.	High speeds, few Msps to 100+ Msps, 8 bits to 16 bits, lower power consumption than flash.	Ultra-High Speed when power consumption not primary concern?
Conversion Method	Oversampling ADC, 5-Hz - 60Hz rejection programmable data output.	Binary search algorithm, internal circuitry runs higher speed.	Small parallel structure, each stage works on one to a few bits.	$N \text{ bits} - 2^{N-1}$ Comparators Caps increase by a factor of 2 for each bit.
Encoding Method	Over-Sampling Modulator, Digital Decimation Filter	Successive Approximation	Digital Correction Logic	Thermometer Code Encoding
Disadvantages	Higher order (4th order or higher) - multibit ADC and multibit feedback DAC.	Speed limited to ~5Msps. May require anti-aliasing filter.	Parallelism increases throughput at the expense of power and latency.	Sparkle codes / metastability, high power consumption, large size, expensive.
Conversion Time	Tradeoff between data output rate and noise free resolution.	Increases linearly with increased resolution.	Increases linearly with increased resolution.	Conversion Time does not change with increased resolution.
Resolution	Component matching requirements double with every bit increase in resolution.	Component matching requirements double with every bit increase in resolution.	Component matching requirements double with every bit increase in resolution.	Component matching typically limits resolution to 8 bits.

Figura 3: Caratteristiche dei vari metodi di conversione (adattata da [4]).

elaborazione. All'aumentare della frequenza di campionamento si è costretti a sacrificare parte della risoluzione (ovvero minor numero di bit) a favore della velocità. Una tabella riassuntiva/comparativa dei principali metodi di conversione è riportata in Fig. 3. Per una descrizione dei vari metodi vedi [5], oppure le Application Notes dei costruttori (ad esempio: Sigma-Delta ADC [6], Successive Approximation Register (SAR)

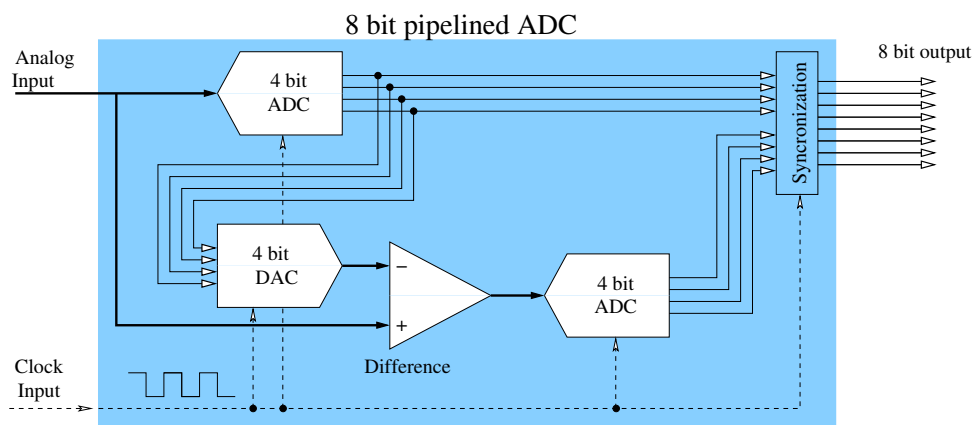


Figura 4: Struttura interna di un convertitore pipelined: il segnale da misurare viene prima convertito con bassa risoluzione. Successivamente viene convertita la differenza tra il valore convertito e l'originale.

ADC [7], Pipeline ADC [8], Flash ADC [9]).

Per quanto riguarda le applicazioni in Fisica Nucleare e SubNucleare, la necessità di analizzare con buona risoluzione segnali provenienti da rivelatori relativamente veloci limita la scelta degli ADC utilizzabili. Al fine di preservare l'informazione di ampiezza (ovvero energia) del rivelatore è necessario l'utilizzo di campionatori con una elevata risoluzione (ovvero alto numero di bit), limitando quindi la frequenza di campionamento dell'ADC utilizzato. D'altra parte, misure di temporizzazione con rivelatori veloci richiedono spesso l'utilizzo di campionatori veloci (e quindi necessariamente limitati in risoluzione). Sono quindi le caratteristiche del rivelatore da impiegare che definiscono il tipo di ADC. In pratica, per l'analisi di segnali di rivelatori quali rivelatori a stato solido, scintillatori organici, rivelatori a gas lenti, si utilizzano campionatori caratterizzati da frequenze di campionamento superiori ai 20 MSamples/s, con risoluzioni di 10–12 bit o superiori. Come si vede in Fig. 2 la grande maggioranza di questi campionatori è realizzata con tecnica pipeline. Per applicazioni riguardanti scintillatori veloci (plastici) e rivelatori a gas veloci si è invece costretti a utilizzare la tecnologia tipo Flash converter, accontentandosi di risoluzioni dell'ordine di 6–8 bit.

La struttura interna di questi due tipi di convertitori è rappresentata in Fig. 4 e Fig. 5.

3 Aliasing e interpolazione

È intuitivo come per il corretto campionamento di un determinato segnale di un rivelatore sia necessario un campionario con una "alta" velocità di campionamento in modo da campionare in "molti"

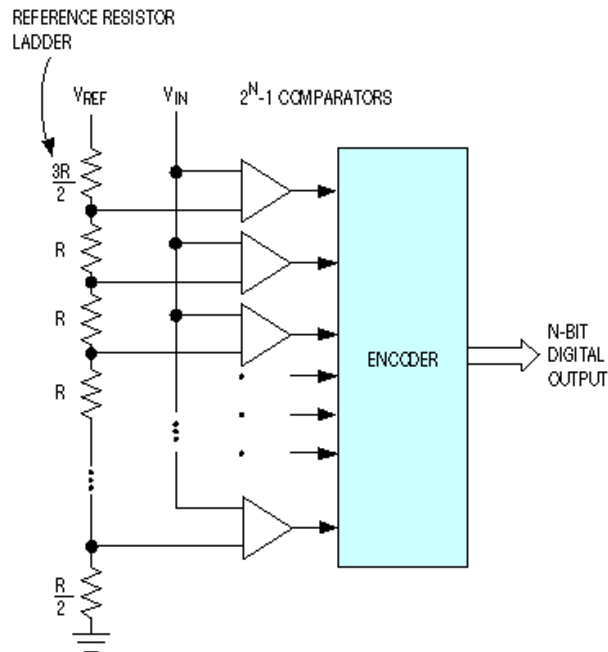


Figura 5: Struttura interna di un convertitore flash: l'ampiezza del segnale da misurare viene confrontata con $2^B - 1$ livelli (ottenuti mediante partitori resistivi di precisione) utilizzando altrettanti comparatori.

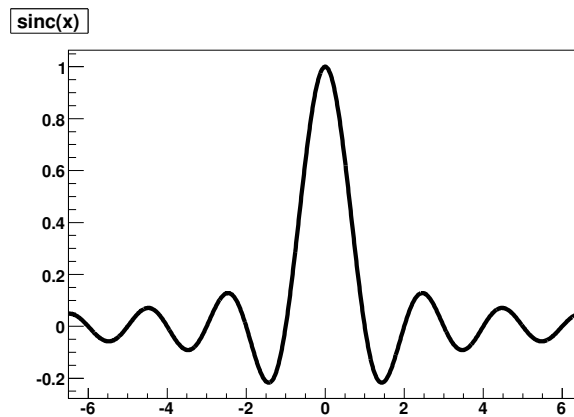


Figura 6: La funzione $\text{sinc}(x)$. Da notare che la funzione non è a supporto compatto e vale zero per x intero non nullo.

punti il segnale di interesse per poi ricostruirlo “sufficientemente” bene. Questo processo è descritto dal seguente teorema:¹

¹Questo teorema è noto anche col nome di Teorema di Nyquist, Teorema dell'interpolazione, Teorema di Shannon, Teorema di Whittaker-Nyquist-Kotelnikov-Shannon.

Teorema del campionamento Nelle ipotesi in cui:

- $S(t)$ è un segnale a banda limitata, ovvero la sua trasformata di Fourier è nulla per $|f| > f_N$, dove f_N è la frequenza di Nyquist del segnale,
- il segnale viene campionato puntualmente e senza rumore aggiuntivo a una frequenza di campionamento $f_S \geq 2f_N$,

allora il segnale $S(t)$ ad ogni tempo t è rappresentato completamente dai campioni $S[k]$ raccolti, e può essere esattamente ricostruito mediante:

$$S'(t) = \sum_{k=-\infty}^{+\infty} S[k] \cdot \text{sinc}\left(\frac{t}{\tau_{\text{clk}}} - k\right) \quad (1)$$

con la definizione $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$.

Questo teorema è alla base di tutta la trattazione matematica dei segnali prodotti da campionatori analogico digitali veloci. In Fig. 6 è rappresentata la funzione sinc.

In Fig. 7 è riportato un esempio riguardante il fenomeno dell'aliasing: i due segnali sinusoidali a 10 e 90 MHz (e quindi ovviamente a banda limitata), vengono campionati da un campionario a 100 MSamples/s. Il segnale a 90 MHz non rispetta la condizione $f_S \geq 2f_N$ e quindi non è possibile la sua corretta ricostruzione: viene a essere scambiato per un segnale a 10 MHz, da cui il termine "aliasing" che descrive questo fenomeno. Al fine di evitare questo fenomeno, in applicazioni pratiche viene introdotto un filtro analogico anti-aliasing prima dell'ingresso del campionario (vedi Fig. 1): questo è un filtro passabasso con frequenza di taglio pari a $f_S/2$ o inferiore. Così facendo, pur introducendo modificazioni al segnale, si evita di campionare componenti di segnale o di rumore con frequenza superiore alla frequenza di Nyquist del campionario. Idealmente il taglio operato dal filtro anti-aliasing deve essere caratterizzato da guadagno unitario nella banda permessa, guadagno zero nella banda proibita, e transizione tra le due zone infinitamente piccola. In realizzazioni pratiche questi requisiti possono essere soddisfatti solo approssimativamente utilizzando filtri analogici con un elevato numero di poli (tipicamente > 3).

L'espressione di Eq. 1 per la ricostruzione del segnale può venire vista come la convoluzione tra il segnale campionato $S[k]$ e un "kernel di interpolazione" (la funzione sinc). In pratica questo vuol dire ricostruire il segnale originario $S(t)$ mediante una interpolazione tra i campioni $S[k]$: un esempio è mostrato in Fig. 8. In (a) è mostrato il segnale

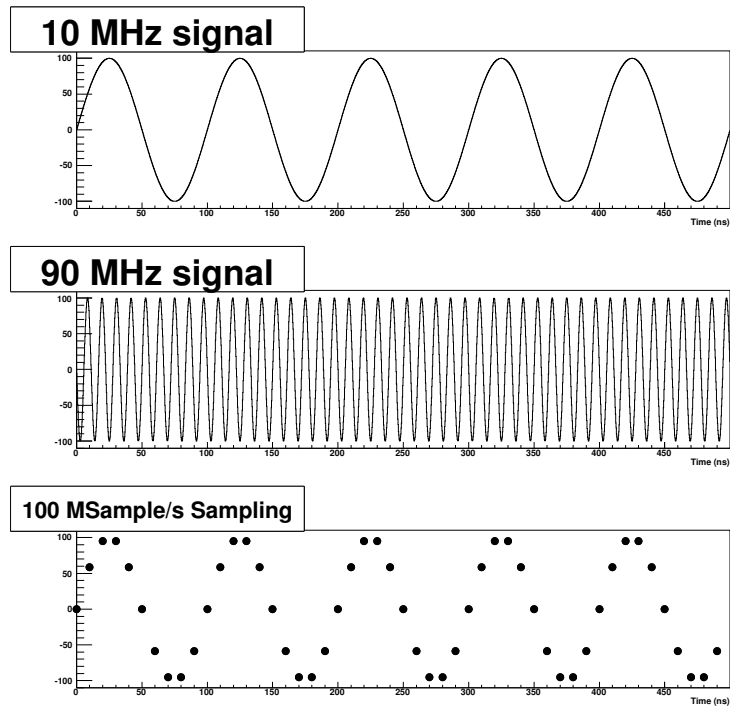


Figura 7: Esempio di aliasing: due segnali di frequenza 10 e 90 MHz producono in uscita lo stesso vettore di dati se campionati a 100 MSamples/s.

originario $S(t)$, che viene campionato producendo la sequenza $S[k]$ (b). In (c) è mostrato il risultato di una interpolazione effettuata usando un kernel di interpolazione diverso dalla funzione sinc, ovvero una semplice funzione lineare. Il segnale originario può essere ricostruito usando una interpolazione a sinc come in (d). A seconda delle applicazioni, può essere richiesta l'accuratezza dell'interpolazione esatta oppure la semplicità di una interpolazione lineare.

Per approfondimenti si rinvia a [2, 1].

4 Rumore introdotto dal processo di conversione

In queste pagine si introduce il problema del rumore introdotto dal processo di campionamento sul segnale. Questo rumore si aggiunge al rumore presente sul segnale del rivelatore e deve quindi essere sufficientemente piccolo da non influenzare le prestazioni del rivelatore medesimo.

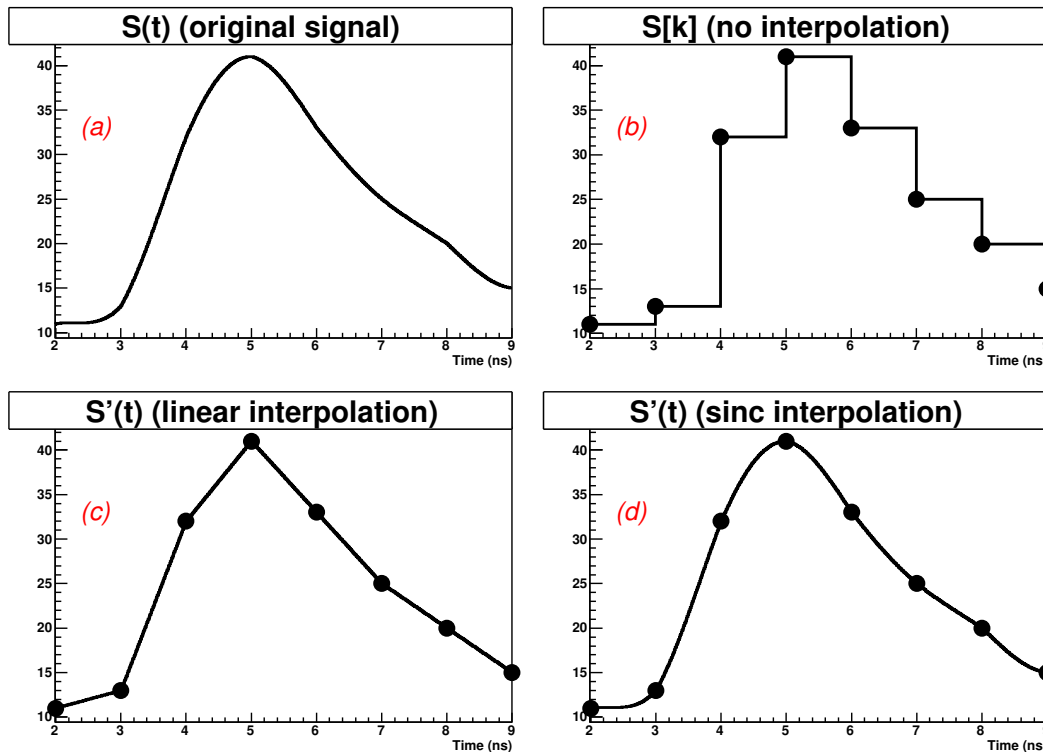


Figura 8: Esempio di interpolazione: il segnale originario (a) viene digitalizzato (b), interpolato mediante una semplice interpolazione lineare (c) o mediante una interpolazione come in Eq. 1 (d).

4.1 Bit, ENOB, rumore di quantizzazione

Il processo di quantizzazione introduce un troncamento dell'ampiezza del segnale in ingresso che può essere schematizzato come:

$$S[k] = \text{troncamento}(S_{n.q.}[k]) \quad (2)$$

dove $S_{n.q.}[k]$ è il segnale non quantizzato. A rigore siamo quindi in presenza di un processo non-lineare e quindi di difficile trattazione analitica.

Possiamo però descrivere l'effetto del campionamento come l'aggiunta di un rumore $n[k]$ sulla sequenza originaria $S_{n.q.}[k]$ tale che $S[k] = n[k] + S_{n.q.}[k]$. L'interessante proprietà di questa schematizzazione è il fatto che per una vasta categoria di situazioni sperimentali il rumore $n[k]$ può essere schematizzato come un rumore bianco e scorrelato in ampiezza con il segnale di ingresso. È quindi possibile definire la varianza del rumore di quantizzazione come $(1/12 = \text{varianza della distribuzione uniforme sull'intervallo } [0, 1])$:

$$\sigma_Q^2 = \frac{R^2}{12} \cdot \frac{1}{4^B} \quad (3)$$

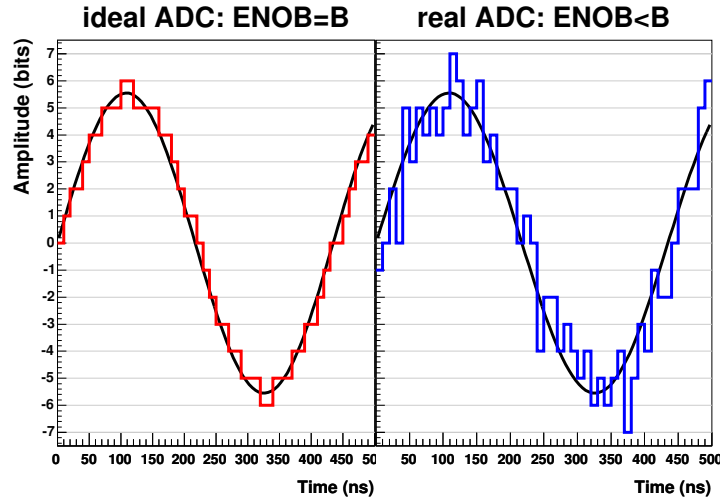


Figura 9: A sinistra: campionamento di un segnale effettuato da un campionatore ideale. A destra: lo stesso segnale campionato da un campionatore reale.

dove il range R del convertitore è stato esplicitamente considerato. La densità spettrale di questo rumore è data da:

$$w_{ADC}(\omega) = \begin{cases} \sigma_Q^2 \cdot \frac{2}{f_S} & ; \text{ if } f < f_N \\ 0 & ; \text{ otherwise} \end{cases} \quad (4)$$

Nel caso di campionatori reali, il rumore di troncamento osservato è peggiore di quello ideale previsto da Eq. 3, come mostrato in Fig. 9. Quello che si osserva è la presenza di rumore aggiuntivo dovuto a sorgenti di rumore interne all'ADC, che si va a sommare a quello idealmente previsto per la sola quantizzazione. Per tenere conto di questo effetto si definisce, in analogia con Eq. 3, il “Numero di Bit Effettivi” del campionatore (ENOB) come:

$$\sigma_{ADC}^2 = \frac{R^2}{12} \cdot \frac{1}{4^{ENOB}} \quad (5)$$

Usualmente $ENOB \simeq B - 1.5$, ad esempio un convertitore da 8 bit è caratterizzato da 6–6.5 ENOB. In tutte le applicazioni pratiche e calcoli teorici il rumore di un convertitore va quindi considerato usando Eq. 5.

4.2 Cenni al problema del random sampling

Nelle nostre applicazioni gli eventi da misurare non sono correlati con il clock del campionatore, ovvero non c'è una relazione di fase fissa tra il segnale misurato e il campionamento del segnale medesimo. Questo effetto, noto col nome di “random sampling”, è schematicamente

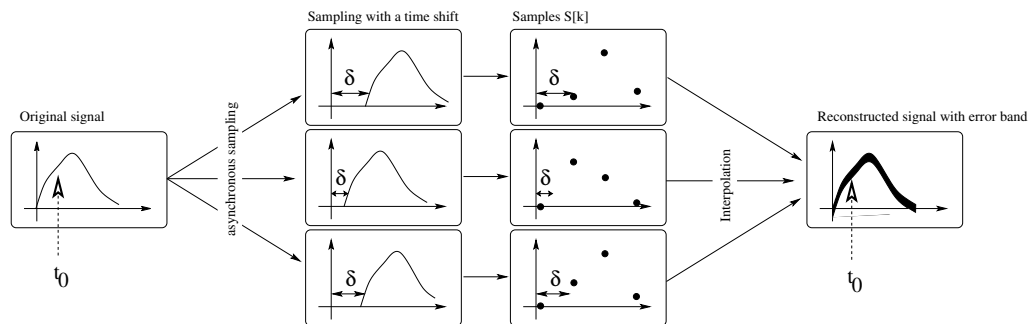


Figura 10: Effetto del “random sampling” sulla ricostruzione di un segnale. Sono mostrati tre esempi di campionamento con fase diversa dello stesso segnale originario.

mostrato in Fig. 10. Il segnale originario viene campionato con varie fasi: dopo la ricostruzione mediante interpolazione, si osserva una dispersione dei segnali, ovvero, in altri termini, la presenza di un rumore aggiuntivo. Questo rumore non è legato al numero di bit del campionatore, ma esclusivamente alla assenza di una relazione di fase precisa tra il clock del campionatore e l’arrivo dei segnali.

Nel caso di configurazioni sperimentali che rispettano tutte le ipotesi del Teorema del campionamento, questo effetto non è presente, in quanto è possibile utilizzare Eq. 1 per ricostruire il segnale originario, fase compresa. Nel caso in cui le ipotesi del teorema non siano soddisfatte (ad esempio presenza di un segnale non limitato in banda, oppure utilizzo di una interpolazione diversa da Eq. 1), questo effetto è presente e deve essere considerato durante l’analisi.

4.3 Altri effetti

Altri effetti di cui tener conto nella caratterizzazione di convertitori analogico digitali sono la non-linearità integrale (ovvero la non proporzionalità tra il segnale di ingresso e il codice convertito) e la non-linearità differenziale (ovvero la non uguaglianza nell’ampiezza dei vari canali di misura). In Fig. 11 questi due effetti sono riportati schematicamente.

In determinate applicazioni (misure di tempo e sincronizzazione a alta precisione) anche il rumore di fase e il jitter del clock del campionatore possono influenzare i risultati della misura.

5 Applicazioni in Fisica Nucleare e SubNucleare

La recente disponibilità tecnologica di campionatori analogico-digitali dotati di adeguati numero di bit e frequenza di campionamento ha reso possibile l’utilizzo pratico di questi dispositivi in applicazioni di Fisica

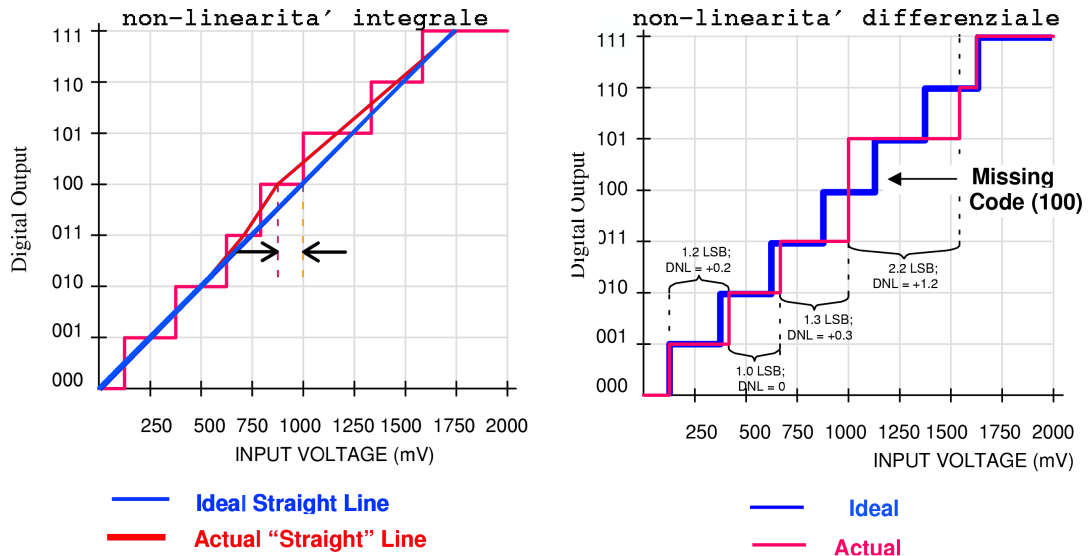


Figura 11: Sinistra: rappresentazione della non-linearità integrale. Destra: rappresentazione della non-linearità differenziale.

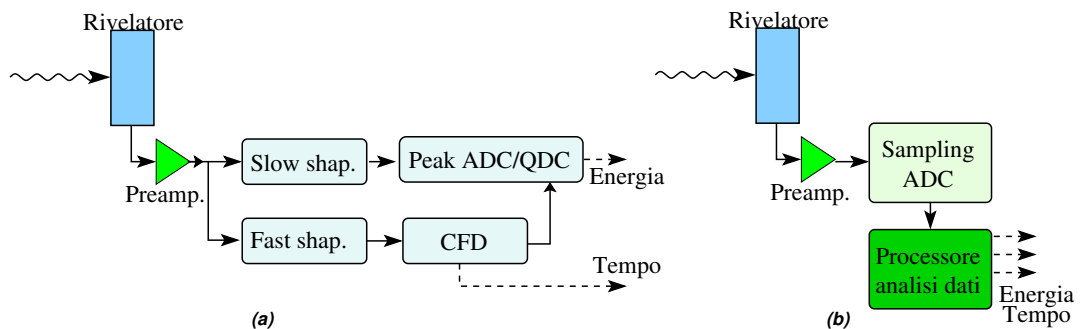


Figura 12: Confronto tra (a) una tipica catena analogica per l'analisi di un rivelatore, (b) l'equivalente catena digitale. Un unico processore può estrarre più variabili a partire dai dati digitalizzati.

Nucleare e SubNucleare. In queste pagine vengono brevemente riassunte le principali caratteristiche delle misure di energia e timing con campionatori veloci.

In queste applicazioni (vedi anche Fig. 12) viene normalmente utilizzato un solo campionatore per ogni rivelatore utilizzato, collegato direttamente all'uscita del preamplificatore. I dati provenienti dal campionatore vengono analizzati da un processore dedicato che estrae dai dati le grandezze di interesse.

5.1 Confronto con i metodi analogici

Nel caso di applicazioni a esperimenti che impiegano molti rivelatori, i

principali vantaggi rispetto ai metodi di misura analogico derivanti dall'utilizzo di sistemi di campionamento veloce sono:

Semplicità della catena elettronica Mentre nella trattazione analogica dei segnali deve essere prevista una catena elettronica separata per ogni informazione da estrarre dal segnale (vedi ad esempio misure fast-slow, misure di energia e temporizzazione,...), in una trattazione digitale dei segnali viene utilizzato un solo convertitore analogico-digitale per la digitalizzazione del segnale, risultando così in una catena elettronica notevolmente semplificata (vedi Fig. 12).

Prestazioni Le prestazioni (in termini di risoluzioni elettroniche) ottenibili da questi dispositivi possono essere superiori a quelle degli equivalenti sistemi analogici. In pratica qualsiasi algoritmo realizzabile con metodi analogici può essere realizzato con metodi digitali se necessario, mentre non è vero il viceversa.

Costi Generalmente inferiori a quelli dei corrispondenti sistemi analogici.

Uno dei principali svantaggi rispetto ai sistemi analogici è la scarsa disponibilità commerciale di sistemi di digitalizzazione dedicati per misure con rivelatori per Fisica Nucleare e SubNucleare. È quindi talvolta necessario progettare e realizzare "in casa" il sistema di campionamento e la sua integrazione con il resto dell'apparato sperimentale.

5.2 Misure di energia e filtri digitali

Il tradizionale metodo di misura dell'ampiezza di un segnale di rivelatore si basa solitamente sulla "formazione" del segnale [10], ovvero l'applicazione di un filtro analogico passa-banda. Ad esempio, in Fig. 13 è riportato l'esempio della formazione di un segnale di rivelatore (curva nera) mediante un filtro CR-RC (curva rossa). Questo processo seleziona la banda di frequenze dove le componenti di segnale sono dominanti rispetto al rumore, risultando così in un netto incremento del rapporto segnale-rumore (SNR).

Un filtro lineare e time-invariant è completamente descritto dalla sua risposta $G(t)$ a un segnale impulsivo $\delta(t)$. L'applicazione di un filtro analogico al segnale $S(t)$ può essere matematicamente descritta come un prodotto di convoluzione tra $S(t)$ e $G(t)$ per ottenere la risposta $R(t)$:

$$R(t) = \int_{-\infty}^t S(\tau)G(t - \tau)d\tau \quad (6)$$

Nell'analisi digitale dei dati è possibile eseguire digitalmente la formazione del segnale: il filtro analogico ("hardware") viene rimosso (vedi Fig. 12) e sostituito da un processore che applica ("software") la funzione di risposta del filtro ai dati digitalizzati. Viene quindi realizzata una

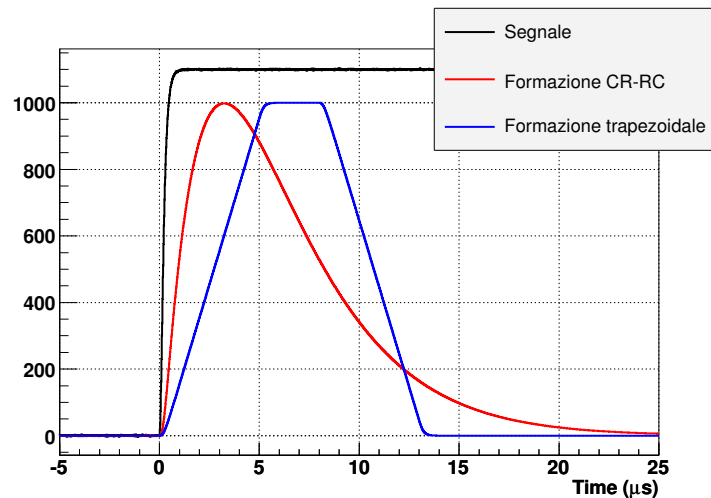


Figura 13: Esempio di formazione di un segnale di rivelatore: in rosso una formazione analogica tipo CR-RC, in blu una tipica formazione digitale (trapezoidale).

convoluzione numerica, del tutto analoga a Eq. 6:

$$R[k] = \sum_{h=-\infty}^k S[h] \cdot G[k-h] \quad (7)$$

Un notevole vantaggio rispetto ai tradizionali metodi di misura analogici è la possibilità di applicare un filtro $G[k]$ di forma *arbitraria* al segnale $S[k]$. Ad esempio, in Fig. 13 è riportata in blu la risposta di un filtro “trapezoidale” al segnale di ingresso (questo tipo di filtro non è esattamente realizzabile con metodi analogici). Questa caratteristica dei filtri digitali permette ottimizzazioni del sistema non realizzabili con le tecniche di misura analogiche, ad esempio è possibile realizzare filtri “calibrati” sui dettagli del rumore in ingresso e che massimizzano (nel senso matematico del termine) la risoluzione della misura. Questo consente di effettuare misure di energia in cui, per molte applicazioni, la risoluzione è limitata dalle prestazioni del rivelatore medesimo e non dall’elettronica di misura utilizzata.

Nelle applicazioni pratiche il prodotto di convoluzione di Eq. 7 viene solitamente sostituito da espressione matematiche equivalenti (filtri ricorsivi e/o buffer circolari) caratterizzate da una notevole velocità di calcolo (vedi [2]).

5.3 Misure di temporizzazione (*)

È ben noto come la realizzazione con metodi analogici di misure di temporizzazione si basi sull’utilizzo di sistemi Constant Fraction Discriminator (CFD), vedi [10]. Lo scopo di questi dispositivi è produrre un

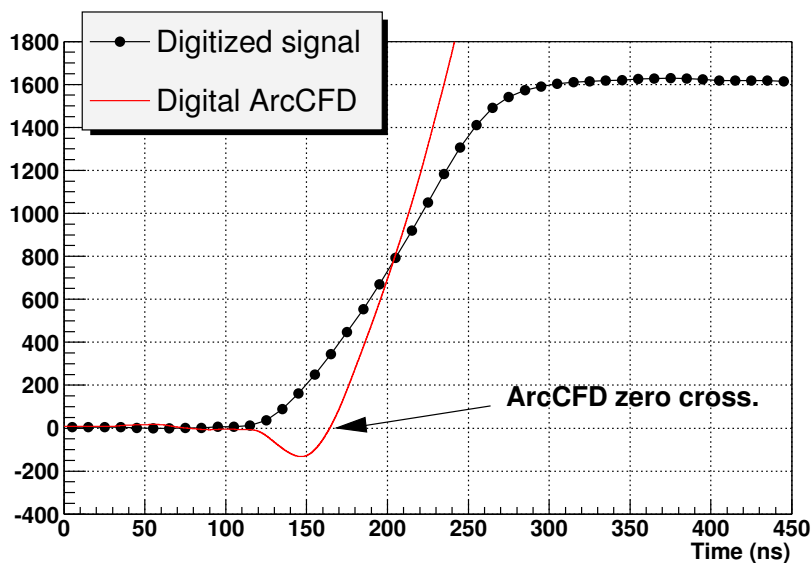


Figura 14: Esempio di misura di temporizzazione digitale: il segnale viene interpolato e digitalmente trattato in modo analogo a un CFD analogico.

segnale logico in corrispondenza all'istante di arrivo nel rivelatore della particella da misurare. Una delle principali difficoltà risiede nella necessità di ottenere una marca di tempo che sia indipendente dall'ampiezza del segnale prodotto dal rivelatore e da piccole variazioni della forma del segnale medesimo.

Per quanto riguarda le misure di temporizzazione con metodi digitali, è ad esempio possibile eseguire sui dati campionati una analisi numerica corrispondente all'operato di un CFD analogico: un esempio è riportato in Fig. 14. Il segnale (in nero) proveniente dal campionario viene numericamente analizzato in modo da simulare l'azione di un CFD (segnale in rosso). L'attraversamento di zero di quest'ultimo segnale fornisce la marca di tempo desiderata. Da notare come l'utilizzo di una adeguata interpolazione tra i campioni raccolti permetta di ottenere una risoluzione sulla misura decisamente migliore del passo di campionamento dell'ADC utilizzato. Ad esempio, in Fig. 14, i dati forniti dal digitalizzatore (indicati col simbolo ●), sono stati interpolati usando un polinomio di grado 3.

5.4 Analisi di forma (*)

○ Oltre alla proporzionalità tra ampiezza del segnale e energia della particella incidente, vari tipi di rivelatori presentano una dipendenza della *forma* del segnale dal tipo di radiazione rivelata (γ , particelle ca-

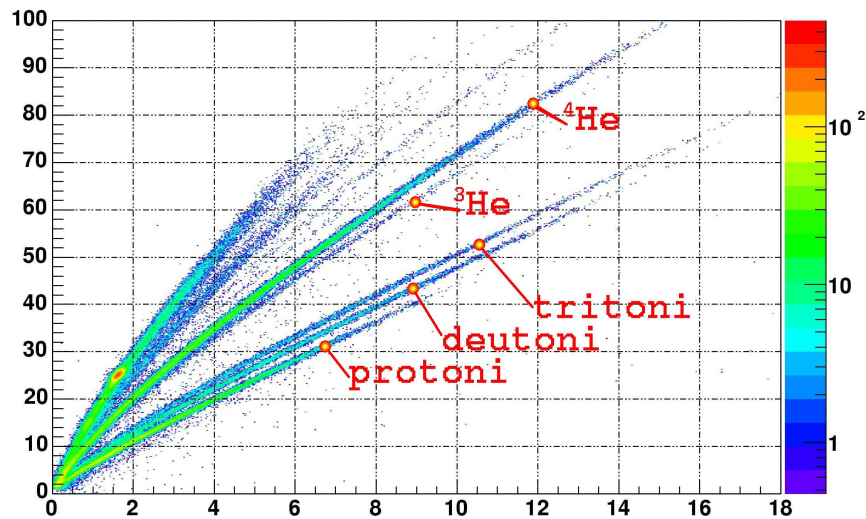


Figura 15: Esempio di analisi di forma: varie particelle cariche vengono identificate analizzando le forme dei segnali di un unico rivelatore CsI(Tl).

riche) oppure dal punto in cui l'interazione è avvenuta all'interno del volume attivo del rivelatore.

Questa variabilità della forme, se da un lato può comportare complicazioni per le misure di energia e timing (deficit balistico, perdita di risoluzione), dall'altro può essere utilmente impiegata per estrarre ulteriori informazioni sull'evento corrente. I sistemi di campionamento digitale, avendo accesso a tutti i dettagli della forma del segnale, sono in grado di eseguire analisi dedicate (fit, correlazioni fast-slow, ...) e estrarre i dati richiesti senza dover ricorrere a catene elettroniche complesse.

Ad esempio, i rivelatori a scintillazione CsI(Tl) sono caratterizzati da due principali costanti di decadimento, il cui peso relativo varia al variare della particella che è stata rivelata. In Fig. 15 è riportato un esempio di identificazione di particelle cariche ottenuta con metodi digitali. Come ulteriore esempio, i rivelatori a germanio coassiali dimostrano una elevata dipendenza della forma del segnale dal punto di interazione (vedi Fig. 16): mediante un "fit" di queste curve è possibile risalire al punto in cui è avvenuta l'interazione, purchè singola.

6 Acquisizione e analisi dati in laboratorio

6.1 L'oscilloscopio digitale

La principale differenza tra un oscilloscopio digitale e un tradizionale oscilloscopio analogico è costituita dal metodo utilizzato per la misura e visualizzazione del segnale: nel caso di un oscilloscopio digitale il segnale viene campionato da un ADC veloce del tipo discusso in pre-

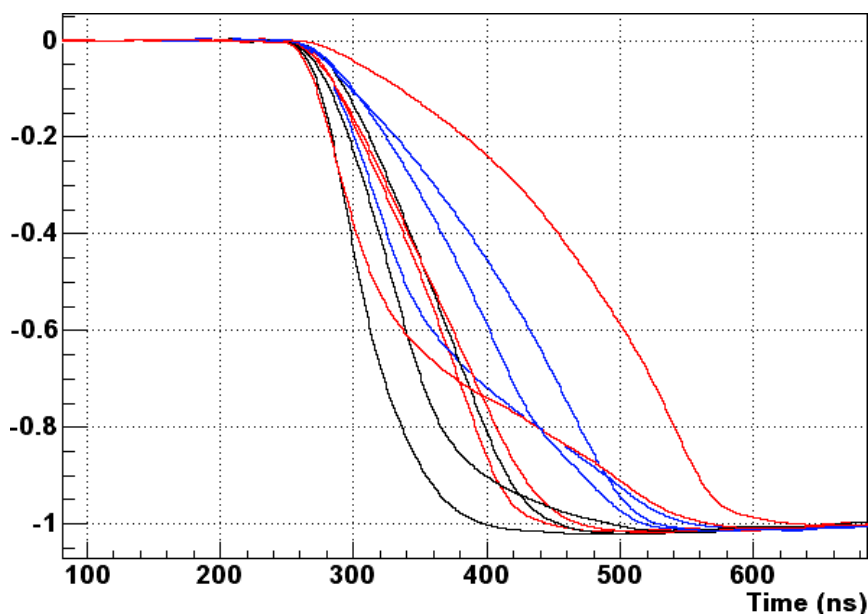


Figura 16: Variabilità dei segnali provenienti da un rivelatore a Ge coassiale: gli eventi sperimentali mostrati hanno tutti la stessa energia (1332 keV).

cedenza (frequenze di campionamento tipiche: 0.1–10 GS/s) e memorizzato in una memoria numerica (di tipo circular buffer). Questa viene continuamente riempita con nuovi campioni alla frequenza di campionamento del dispositivo fino a quando la logica di trigger dell'oscilloscopio, molto simile a quella presente negli oscilloscopi analogici, non segnala la presenza di un trigger.

In questo caso la memoria viene congelata e inizia il trasferimento della forma d'onda acquisita verso un display grafico, in maniera del tutto analoga a quanto si potrebbe fare con un normale calcolatore. Non appena il processo è terminato, la memoria diventa disponibile per accettare nuovi campioni, e il campionamento ricomincia, in attesa del prossimo trigger.

Si può vedere quindi come questo tipo di oscilloscopi permetta non solo di effettuare tutte le operazioni comunemente svolte con oscilloscopi analogici (visualizzazione delle forme, espansione delle scale orizzontali e verticali, trigger sul segnale, ...) ma anche tutta una serie di nuove funzioni. Ad esempio, è possibile memorizzare un segnale e visualizzarlo per un tempo arbitrario, eseguire calcoli sulle forme analizzate (tempo di salita/discesa, rumore, ...), eseguire trigger relativamente sofisticati (es.: coincidenze logiche).

Per poter adattare la misura alla base dei tempi richiesta, l'ADC dell'oscilloscopio digitale è normalmente capace di lavorare a varie frequenze di campionamento: quando è necessaria l'analisi e visualizzazione di segnali veloci l'ADC viene fatto lavorare a frequenze di campionamento

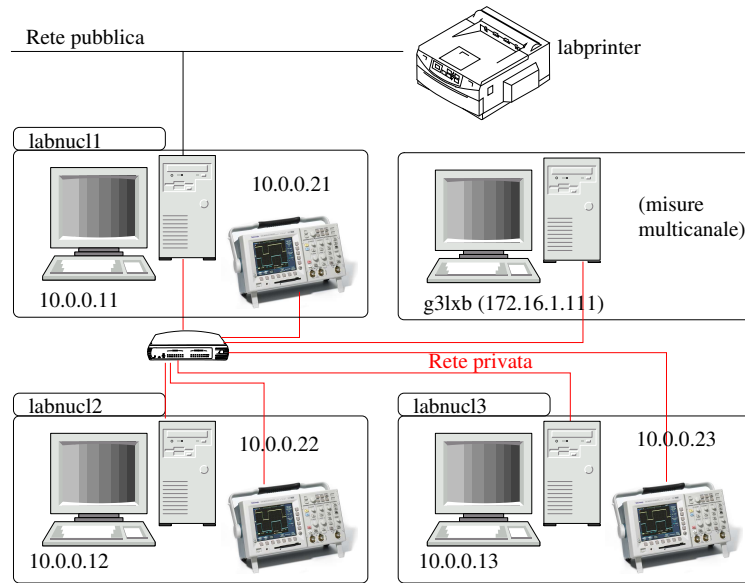


Figura 17: Disegno della rete presente in laboratorio.

elevate, mentre per segnali lenti si utilizzano frequenze di campionamento più basse. L'oscilloscopio è normalmente in grado di visualizzare, per ogni base dei tempi impostata, un numero fisso di punti (dell'ordine di 10000) e questo giustifica quindi la scelta fatta di una frequenza di campionamento variabile.

Un'altra peculiarità dell'oscilloscopio digitale è quella di poter estrarre i dati raccolti dal campionatore e quindi trasferirli su un computer. Questo metodo verrà ampiamente usato in laboratorio per analizzare i segnali provenienti dai vari rivelatori.

6.2 Apparato sperimentale

L'apparato sperimentale disponibile in laboratorio per la misura e analisi dei segnali di rivelatori è costituito da 3 oscilloscopi digitali Tektronix TDS 3052B (2 canali di misura, 5 GSamples/s, tecnica di conversione flash/analog-pipeline) e alcuni computer con sistema operativo Linux, collegati in rete (vedi in Fig. 17).

È possibile accedere ai dati contenuti nell'oscilloscopio via HTTP (ad esempio <http://10.0.0.21/>), analogamente a quanto viene usualmente fatto per la normale navigazione su WEB. Questo tipo di accesso è stato preferito rispetto ad altri metodi comunemente applicati in Fisica Nucleare e SubNucleare per la sua notevole semplicità di realizzazione (nel nostro caso ogni computer ha accesso a tutti gli oscilloscopi) e la disponibilità di una velocità di acquisizione modesta ma adatta ai nostri scopi.

L'acquisizione e analisi di un evento procede nel seguente modo: ini-

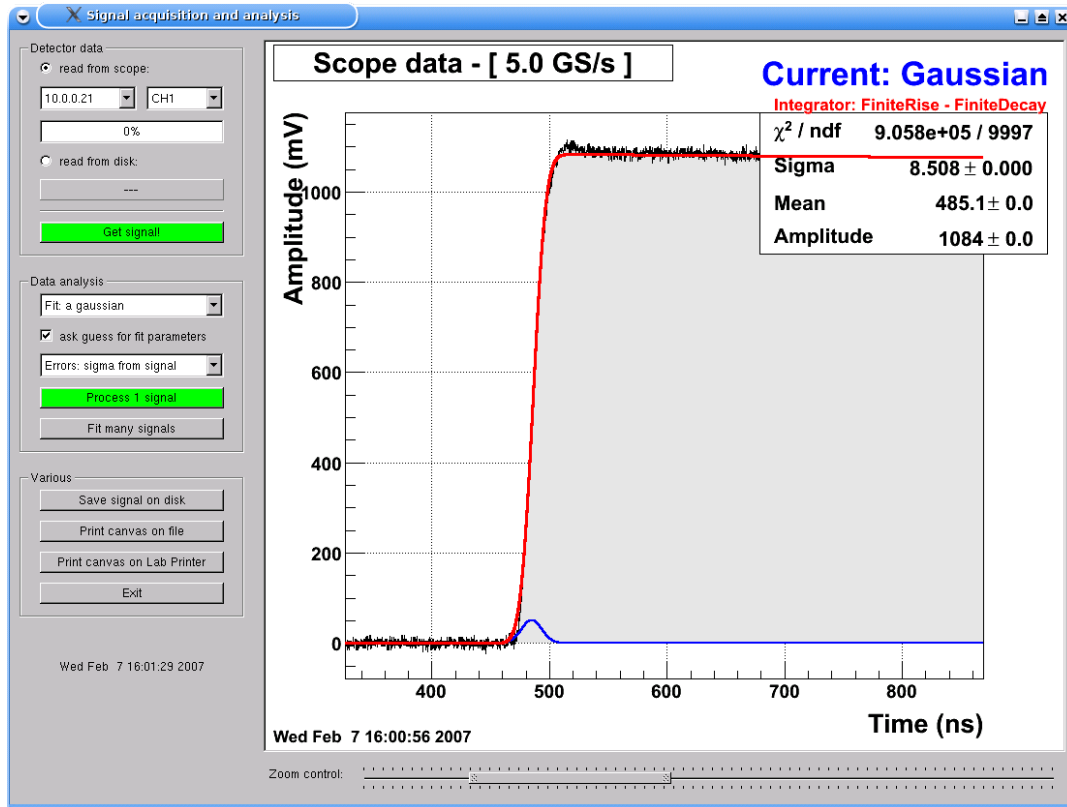


Figura 18: Schermata del programma di acquisizione e analisi dei segnali raccolti con l'oscilloscopio digitale.


zialmente l'oscilloscopio viene impostato manualmente con il tipo di trigger, scala orizzontale (temporale) e scala verticale desiderati in modo da vedere correttamente sul display dell'oscilloscopio stesso i segnali di interesse. Fatto questo, il programma di acquisizione e analisi (vedi nel seguito) richiede via HTTP i dati corrispondenti all'evento corrente all'oscilloscopio in esame. I dati vengono convertiti dal formato interno dell'oscilloscopio al formato usato (vedi oltre) e visualizzati.

6.3 Il programma di acquisizione e analisi

Il programma di analisi è stato realizzato utilizzando il software ROOT [11], un pacchetto di analisi dati sviluppato per la Fisica delle Alte Energie (principalmente LHC), basato sul linguaggio di programmazione C++. In Fig. 18 è mostrata la schermata principale. Sulla destra è presente un'area grafica ("canvas") dove vengono disegnati i segnali acquisiti e mostrati i risultati dei fit, mentre sulla sinistra sono presenti vari bottoni per accedere alle varie funzioni.



Per fare partire il programma, effettuare il log-in sul computer se necessario, e digitare sul terminale testo (solo le parti *colorate*):


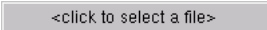
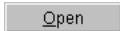

```
[laboratorio@labnucl1 ~]$ cd acq
[laboratorio@labnucl1 acq]$ ./START
```

Per uscire dal programma utilizzare il bottone . Nel caso sia necessario spegnere il computer, aprire un terminale e digitare `poweroff`. Utilizzare il comando `reboot` per riavviare.

6.4 Acquisizione dei segnali

I segnali possono o essere acquisiti via rete dall'oscilloscopio o ri-letti da disco:


lettura da oscilloscopio selezionare l'opzione  `read from scope:`. Dai due bottoni a selezione è possibile selezionare l'oscilloscopio desiderato e il canale di misura. Per acquisire un evento cliccare sul bottone  (l'indicatore di avanzamento permette di verificare il corso dell'acquisizione). Nel caso di errori, nella zona dell'interfaccia grafica immediatamente sopra l'ora, apparirà un messaggio di errore.

lettura da disco selezionare l'opzione  `read from disk:` e cliccare sul bottone  sottostante. Si aprirà una finestra di dialogo per selezionare un file `.CSV` da caricare. Cliccare su  per caricare il file. Per ri-caricarlo cliccare su .

In entrambi i casi il programma esegue una sottrazione della media dei primi 100 campioni per riportare la linea di base a zero. Terminata l'acquisizione, il segnale viene visualizzato. Alcuni parametri di interesse (baseline originaria, stima del rumore, area, ...) vengono inoltre calcolati e visualizzati in un riquadro.

6.5 Analisi dati

Una volta acquisito, da oscilloscopio o da disco, il segnale viene disegnato nel canvas: l'intervallo di tempo visualizzato corrisponde alla capacità di memoria dell'oscilloscopio, utilizzato alla frequenza di campionamento corrente. L'oscilloscopio regola automaticamente la frequenza di campionamento a seconda della base dei tempi utilizzata: intervalli di tempo piccoli vengono campionati alla massima frequenza di campionamento (5 GSamples/s per TDS 3052B), mentre frequenze inferiori sono usate per intervalli di tempo lunghi. La frequenza di campionamento dell'evento corrente è visualizzata nel riquadro in alto a sinistra (titolo del canvas).

Per visualizzare una zona specifica del segnale (zoom orizzontale o verticale) portare il cursore sull'asse fino a quando non si trasforma in : quindi cliccare e trascinare. Per tornare alla visualizzazione di tutto il

range portare di nuovo il mouse sull'asse, cliccare col destro e selezionare Unzoom. Alternativamente si può usare il "doppio slider" presente nella parte inferiore della finestra.

L'analisi vera e propria dei segnali viene condotta selezionando una delle funzionalità disponibili dal combobox . Fatto questo, cliccare sul bottone per avviare l'analisi. A parte il fit dei segnali con funzioni predefinite (vedi dopo), sono disponibili le seguenti funzionalità:

Zoom auto Questa funzione cerca di espandere la zona in cui è presente il segnale, tenendo conto della presenza di rumore.

Redraw Riaggiorna il canvas.

Derivative Calcola la derivata del segnale. Applicato a un segnale di un preamplificatore di carica permette di avere un'idea del segnale di corrente (la derivata del segnale viene filtrata da un filtro passabasso per togliere parte del rumore in alta frequenza $\propto \omega^2$ dovuto al processo di derivata).

Integral Calcola l'integrale del segnale. Applicato a un segnale di un preamplificatore di corrente o fotomoltiplicatore permette di avere un'idea dell'uscita che sarebbe stata ottenuta in presenza di un preamplificatore di carica.

Interpolate: lin 10x Esegue una interpolazione del segnale, utilizzando una interpolazione lineare (vedi Sect. 3). Il segnale interpolato viene generato con un passo temporale finito pari a 1/10 del passo di campionamento del segnale originario. Solo la zona vicina al massimo del segnale viene interpolata.

6.6 Il fit dei segnali

Come già visto durante il corso, esistono vari modelli della Fisica dei rivelatori che portano alla definizione di varie funzioni che, con vari gradi di approssimazione, sono in grado di riprodurre l'andamento dei segnali in uscita dal rivelatore. All'interno del programma di acquisizione sono presenti varie funzioni "standard" che possono essere utilizzate per il fit delle forme d'onda acquisite tramite oscilloscopio. Il fit dei segnali viene eseguito mediante il metodo dei minimi quadrati [12, 13] usando il programma di minimizzazione MINUIT [14], integrato all'interno di ROOT. Per una introduzione sistematica il problema del fit dei dati sperimentali e relativa analisi degli errori vedi [12, 13].

Riassumendo, data una funzione:

$$f(t; \vec{p}) \quad (8)$$

dove t è il tempo e \vec{p} è il vettore dei parametri della funzione, viene minimizzata la seguente espressione:

$$X^2 = \frac{1}{\sigma^2} \sum_{k=1}^N [S[k] - f(k\tau_{\text{clk}}; \vec{p})]^2 \quad (9)$$

dove τ_{clk} è il passo di campionamento utilizzato, $S[k]$ sono i dati digitalizzati in funzione dell'indice k , σ^2 è l'errore da attribuire ai campioni $S[k]$ (assunto uguale per tutti i campioni k , vedi oltre). La minimizzazione dell'espressione Eq. 9 può essere eseguita semplicemente se la dipendenza dai parametri \vec{p} è di tipo lineare. Nel caso in cui la dipendenza dai parametri sia non-lineare, è necessario fornire al programma di fit un valore iniziale (stima di "guess") per ciascuno dei parametri: la funzione da fittare viene quindi linearizzata in un intorno del punto di guess \vec{p}_g . Ad esempio, nel caso di un solo parametro p :

$$f(t; p) \simeq f(t; p_g) + \left[\frac{\partial f(t; p)}{\partial p} \right]_{p_g} \cdot (p - p_g) \quad (10)$$

È quindi possibile eseguire il fit e determinare il vettore dei parametri \vec{p} ; il valore così determinato può essere usato ricorsivamente come nuovo valore di guess \vec{p}_g . Il processo si arresta non appena le correzioni ai valori \vec{p} diventano minori di un certo predefinito valore di soglia. Nel caso di funzioni con molti parametri o poco regolari, l'accuratezza dei valori di guess forniti al programma è cruciale nello stabilire la convergenza o meno del fit ad una soluzione soddisfacente (si vogliono evitare i minimi locali di X^2).

Una volta eseguito il fit il programma visualizza graficamente la funzione risultante e i valori (con errore) dei vari parametri. Viene inoltre visualizzato il valore di X^2 e il numero di gradi di libertà del fit.

L'utilizzo di Eq. 9 presuppone la validità di varie ipotesi sul segnale $S[k]$ in esame: in particolare, si fa l'ipotesi che la distribuzione del rumore sui campioni sia bianca (ovvero assenza di correlazione tra il rumore sul campione $S[k]$ e gli altri), stazionaria e gaussiana. Sotto queste ipotesi σ non dipende dall'indice di campione k ed è inoltre possibile utilizzare il valore di X^2 , normalizzato ai gradi di libertà NDF del sistema, per valutare la bontà del fit, ovvero la validità delle assunzioni fatte sul segnale. È però necessario tenere conto delle seguenti osservazioni:

- Il programma di fit assume la presenza di un rumore *rigorosamente* bianco sul segnale, con l'opzione di utilizzare varianza del segnale sulla linea di base o un valore unitario. Nel secondo caso è necessario riscalarlo il valore di X^2 fornito dal programma per tener conto del reale valore di σ^2 . È inoltre possibile, nel caso in cui si stiano fittando segnali di scintillatori, introdurre le relative fluttuazioni poissoniane: in questo caso sarà richiesta l'ampiezza del singolo fotolelettore.
- La presenza di rumore non bianco sul segnale del rivelatore può influenzare notevolmente il valore di X^2 fornito dal programma, a tal

punto da rendere errato l'utilizzo di tale valore per un test quantitativo sulla qualità del fit. Nei casi in cui si richieda una corretta valutazione quantitativa della qualità del fit, è necessario tener conto della completa funzione di autocorrelazione tra i campioni $S[k]$. Esempi di fit con dati correlati sono discussi ad esempio in [12, 13, 15].

- Le funzioni utilizzate per il fit sono “idealizzate”, ovvero sono solitamente ottenute mediante un trattamento semplificato della Fisica del rivelatore e dell'elettronica collegata, e non tengono inoltre conto di vari fattori sperimentali (ad esempio la banda passante finita dei componenti utilizzati, o la presenza di parziali riflessioni sulle linee).

Non è quindi possibile usare il valore di X^2/NDF come un test rigoroso della qualità del fit, ma solo come un indicatore semiquantitativo. Per analoghi motivi, gli errori sui parametri non vanno considerati come statisticamente corretti (ovvero come intervalli di confidenza per il parametro in questione) ma solo come stime.

Il fit del segnale acquisito viene scomposto in due “componenti”: l'originario segnale di corrente $i(t)$ prodotto dal rivelatore in esame e la risposta $G_{preamp}(t)$ dell'elettronica seguente (fotomoltiplicatore, preamplificatore, ...). Quindi la funzione $f(t; \vec{p})$ di Eq. 8 può essere scritta nel seguente modo:

$$f(t; \vec{p}) = i(t; \vec{p}) * G_{preamp} \quad (11)$$

All'interno del programma di analisi sono presenti alcune funzioni di risposta G_{preamp} dell'elettronica di preamplificazione: gli eventuali parametri di G_{preamp} (ad esempio il tempo di salita del preamplificatore) *non* vengono fittati dal programma, ovvero vanno misurati in precedenza.

Per eseguire un fit del segnale è necessario selezionare la funzione da fittare mediante il combobox e poi cliccare sul bottone . Nel caso in cui la richiesta dei valori di guess sia abilitata mediante ask guess for fit parameters, si apre il dialogo in Fig. 19. In questo dialogo è possibile impostare, oltre al valore di guess per ciascuno dei parametri della funzione selezionata, anche le caratteristiche del preamplificatore usato e i limiti per il fit del segnale. Cliccando su viene eseguito il fit. Nel caso la casella ask guess for fit parameters sia disabilitata, il fit viene eseguito usando l'ultima guess fornita per i parametri. Il programma memorizza su disco l'ultima serie di valori di guess fornita per ogni funzione. A fit terminato viene visualizzata in rosso la funzione fittata, e in blu il segnale di corrente del rivelatore.

A seconda dei valori impostati (o non impostati) per il tempo di salita o decay, è possibile includere nel fit vari tipi di elettronica di preamplificazione. Ecco qualche esempio:

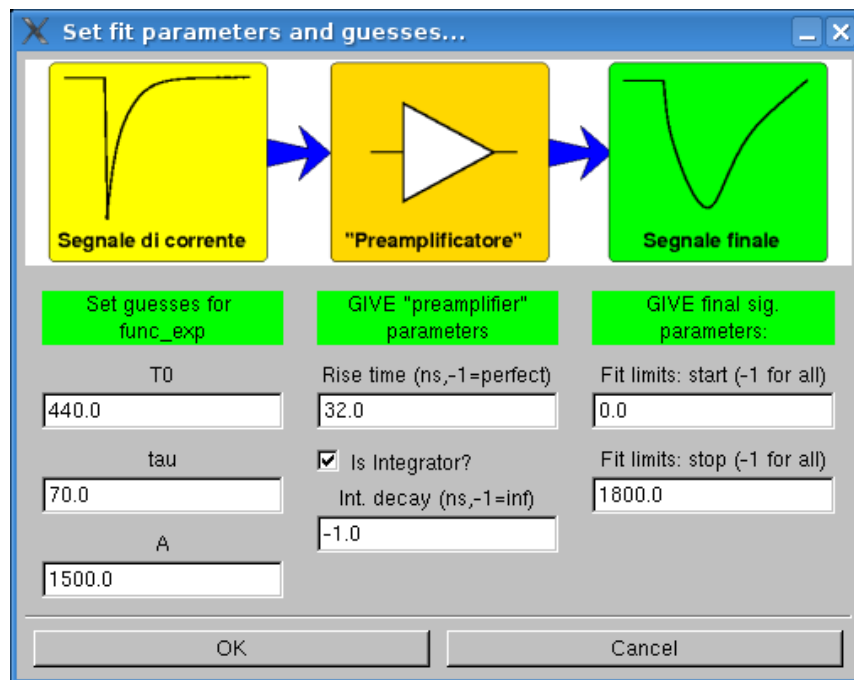


Figura 19: Dialogo per fornire i valori di guess della funzione, le caratteristiche del preamplificatore e i limiti del fit. Esempio per il fit di una funzione tipo $\exp(-t/\tau)$.

Caso desiderato	T.rise (ns)	Is Integrator?	T.decay (ns)
Preamp. ideale	-1	<input type="checkbox"/>	-
Integratore ideale	-1	<input checked="" type="checkbox"/>	-1
Preamp. reale	valore	<input type="checkbox"/>	-
Integr. reale senza decay	valore	<input checked="" type="checkbox"/>	-1
Integr. reale con decay	valore	<input checked="" type="checkbox"/>	valore

NOTA: il tempo di salita finito viene schematizzato mediante un filtro RC passabasso a 2 poli, mentre il tempo di decadimento mediante un decadimento esponenziale.

Utilizzando il bottone è possibile acquisire da oscilloscopio un certo numero di eventi, fittare ciascuno di essi con la funzione selezionata e quindi visualizzare le distribuzioni dei parametri del fit.

6.7 Elenco delle funzioni per il fit di segnali

Nel seguito viene presentato un elenco delle funzioni attualmente disponibili: per ognuna di esse, oltre ad una immagine e all'espressione matematica, è incluso il listato del programma che la realizza.

Ognuna di queste funzioni vuole essere, secondo un certo modello, una rappresentazione del segnale *di corrente* di un certo rivelatore.

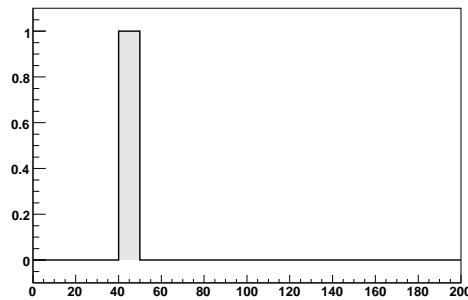
Il programma è diviso in due parti:

Definizione della funzione Qui risiede la definizione vera e propria: dato il tempo t (ovvero $x[0]$) e il vettore dei parametri, viene calcolato il valore della funzione.

Inizializzazione Qui la funzione viene inizializzata, in modo che possa essere utilizzata dal resto del programma. Vengono inoltre impostati i nomi dei vari parametri.

Da notare che molte funzioni includono un parametro t_0 che corrisponde all'istante di arrivo della particella rispetto al tempo zero dell'evento corrente (arbitrario). La presenza di una linea di base diversa da zero non è inclusa nelle funzioni.

6.7.1 Funzione delta con durata finita



Funzione del tipo:

$$i(t) = \begin{cases} A & \text{if } t > t_0 \text{ or } t < t_0 + t_W \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

```

0  /*******
1  func_delta.C: definition of a delta-like function
2  *****/
3
4  double func_delta(double *x, double *pars)
5  {
6      // the time variable:
7      double t=x[0];
8
9      // parameters:
10     double t0    =pars[0];
11     double A     =pars[1];
12     double deltaw =pars[2];
13     // function definition:

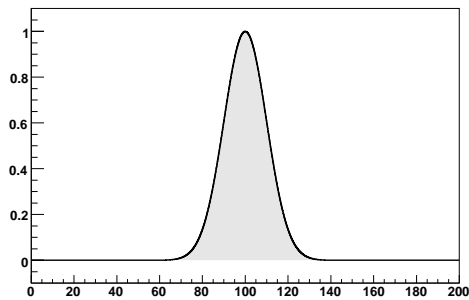
```

```

14  if(t<t0)
15      return 0;
16  t= t-t0;
17
18  if(t>deltaw) return 0;
19
20  return
21      A/deltaw;
22  }
23
24  /*****
25  /*****
26  class Func_Delta: public FitSignal{
27  public:
28      // create function
29  Func_Delta(): FitSignal(func_delta, // a name
30      func_delta, // the C function
31      3) // the number of parameters
32  {
33      //set names
34      SetFuncName(Finite #delta);
35      SetParName(0,T0);
36      SetParName(1,A);
37      SetParName(2,Width);
38  }
39  };
40  /*****

```

6.7.2 Funzione gaussiana

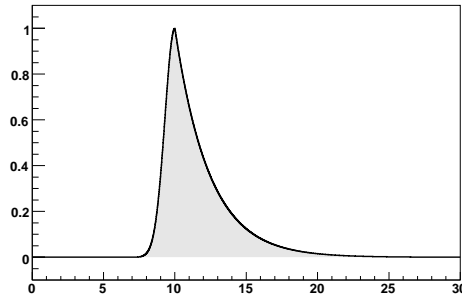


Funzione del tipo:

$$i(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(t - t_m)^2}{2\sigma^2} \quad (13)$$

```
1  *****
2  func_gaus.C: definition of a gaussian function
3  *****/
4
5  double func_gaus(double *x, double *pars)
6  {
7      // the time variable:
8      double t=x[0];
9
10     // parameters:
11     double sigma =pars[0];
12     double mean   =pars[1];
13     double A      =pars[2];
14
15     // function definition:
16     return
17         A/( sqrt(2*M_PI)*sigma )*
18         exp(-(t-mean)*(t-mean)/(2.*sigma*sigma));
19 }
20
21 *****/
22 *****/
23 class Func_Gaus: public FitSignal{
24 public:
25     // create function
26     Func_Gaus(): FitSignal(func_gaus,// a name
27         func_gaus, // the C function
28         3) // the number of parameters
29     {
30         //set names
31         SetFuncName(Gaussian);
32         SetParName(0,Sigma);
33         SetParName(1,Mean);
34         SetParName(2,Amplitude);
35     }
36 };
37 *****/
```

6.7.3 Funzione gaussiana·exp



Funzione del tipo:

$$i(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(t-t_m)^2}{2\sigma^2}\right] \cdot \exp\left(-\frac{t}{\tau}\right) \quad (14)$$

```

1  *****
2  func_gausexp.C: definition of a gaus(t)*exp(t) function
3  *****
4
5  double func_gausexp(double *x, double *pars)
6  {
7      // the time variable:
8      double t=x[0];
9
10     // parameters:
11     double sigma =pars[0];
12     double mean =pars[1];
13     double A =pars[2];
14     double tau =pars[3];
15
16     // function definition:
17     if(t<mean)
18         return
19             A/( sqrt(2*M_PI)*sigma )*
20             exp(-(t-mean)*(t-mean)/(2.*sigma*sigma));
21     return
22         A/( sqrt(2*M_PI)*sigma )* exp(-(t-mean)/tau);
23 }
24
25 *****
26 *****
27 class Func_Gausexp: public FitSignal{
28 public:
29     // create function
30     Func_Gausexp(): FitSignal(func_gausexp,// a name
31         func_gausexp, // the C function
32         4) // the number of parameters

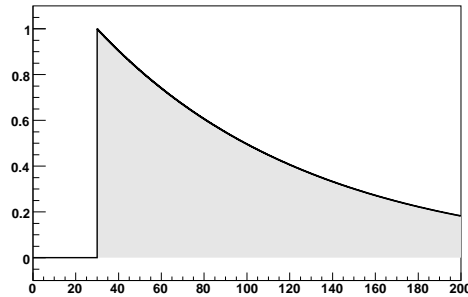
```

```

33 {
34     //set names
35     SetFuncName(Gaus(t)*exp(t));
36     SetParName(0,Sigma);
37     SetParName(1,Mean);
38     SetParName(2,Amplitude);
39     SetParName(3,Tau);
40 }
41 };
42 /*****/

```

6.7.4 Funzione $\exp(-t/\tau)$



Funzione del tipo:

$$i(t) = \begin{cases} \frac{A}{\tau} \exp\left(-\frac{t-t_0}{\tau}\right) & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

```

0 /*****
1 func_exp.C: definition of a exp function
2 *****/
3
4 double func_exp(double *x, double *pars)
5 {
6     // the time variable:
7     double t=x[0];
8
9     // parameters:
10    double t0    =pars[0];
11    double tau   =pars[1];
12    double A     =pars[2];
13
14    // function definition:
15    if(t<t0)
16        return 0;
17    t= t-t0;

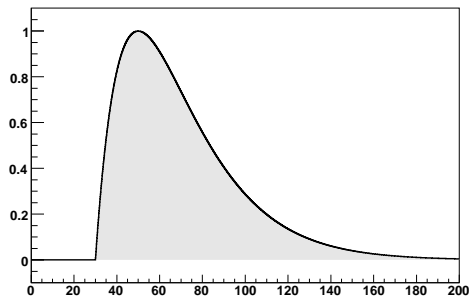
```

```

18
19     return
20     A /tau*exp(-t/tau);
21 }
22
23 /**/
24 /**/
25 class Func_Exp: public FitSignal{
26 public:
27     // create function
28     Func_Exp(): FitSignal(func_exp,// a name
29         func_exp, // the C function
30         3) // the number of parameters
31     {
32         //set names
33         SetFuncName(exp(-t/#tau));
34         SetParName(0,T0);
35         SetParName(1,tau);
36         SetParName(2,Area);
37     }
38 };
39 /**/

```

6.7.5 Funzione $t \exp(-t/\tau)$



Funzione del tipo:

$$i(t) = \begin{cases} A \cdot \frac{t}{\tau^2} \exp\left(-\frac{t-t_0}{\tau}\right) & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

```

1 /**/
2 func_texp.C: definition of a exp function with
3 same tau for rise and fall time
4 /**/
5
6 double func_texp(double *x, double *pars)

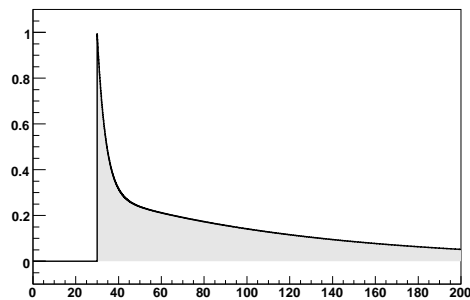
```

```

7 {
8  // the time variable:
9  double t=x[0];
10
11  // parameters:
12  double t0      =pars[0];
13  double tau     =pars[1];
14  double A       =pars[2];
15
16  // function definition:
17  if(t<t0)
18      return 0;
19  t= t-t0;
20
21  return
22      A *t/(tau*tau)*exp(-t/tau);
23 }
24
25 /*****
26 /*****
27 class Func_TExp: public FitSignal{
28 public:
29  // create function
30  Func_TExp(): FitSignal(func_texp, // a name
31      func_texp, // the C function
32      3) // the number of parameters
33  {
34      //set names
35      SetFuncName(t*exp(-t/#tau));
36      SetParName(0,T0);
37      SetParName(1,tau);
38      SetParName(2,Area);
39  }
40 };
41 /*****

```

6.7.6 Funzione $A_1 \exp(-t/\tau_1) + A_2 \exp(-t/\tau_2)$



Funzione del tipo:

$$i(t) = \begin{cases} \frac{A1}{\tau_1} \exp\left(-\frac{t-t_0}{\tau_1}\right) + \frac{A2}{\tau_2} \exp\left(-\frac{t-t_0}{\tau_2}\right) & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

Questa funzione è utilizzabile per rivelatori in cui il segnale di corrente sia scomponibile in due componenti, come ad esempio alcuni scintillatori.

```

1  /*****
2  func_expexp.C: definition of a exp+exp function
3  *****/
4
5  double func_expexp(double *x, double *pars)
6  {
7      // the time variable:
8      double t=x[0];
9
10     // parameters:
11     double t0      =pars[0];
12     double tau1    =pars[1];
13     double tau2    =pars[2];
14     double A1      =pars[3];
15     double A2      =pars[4];
16
17     // function definition:
18     if(t<t0)
19         return 0;
20     t= t-t0;
21
22     return
23     A1/tau1*exp(-t/tau1)
24     +
25     A2/tau2*exp(-t/tau2);
26
27 }
28
29 /*****
30 *****/
31 class Func_Expexp: public FitSignal{
32 public:
33     // create function
34     Func_Expexp(): FitSignal(func_expexp, // a name
35         func_expexp, // the C function
36         5) // the number of parameters
37     {
38         //set names
39         SetFuncName(A1*exp(-t/#tau 1)+A2*exp(-t/#tau 2));
40         SetParName(0,T0);
41         SetParName(1,tau 1);
42         SetParName(2,tau 2);
43         SetParName(3,Area 1);

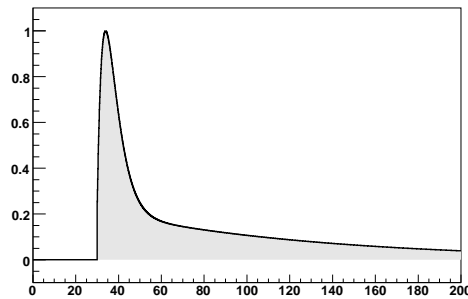
```

```

44     SetParName(4,Area 2);
45   }
46 };
47 /*****

```

6.7.7 Funzione $A_1 t \exp(-t/\tau_1) + A_2 \exp(-t/\tau_2)$



Funzione del tipo:

$$i(t) = \begin{cases} \frac{A_1}{\tau_1} \cdot t \cdot \exp\left(-\frac{t-t_0}{\tau_1}\right) + \frac{A_2}{\tau_2} \exp\left(-\frac{t-t_0}{\tau_2}\right) & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Simile alla precedente, ma con un termine lineare in t .

```

1  /*****
2  func_expexp.C: definition of a t*exp+exp function
3  *****/
4
5  double func_texpexp(double *x, double *pars)
6  {
7    // the time variable:
8    double t=x[0];
9
10   // parameters:
11   double t0    =pars[0];
12   double tau1  =pars[1];
13   double tau2  =pars[2];
14   double A1    =pars[3];
15   double A2    =pars[4];
16
17   // function definition:
18   if(t<t0)
19     return 0;
20   t= t-t0;
21

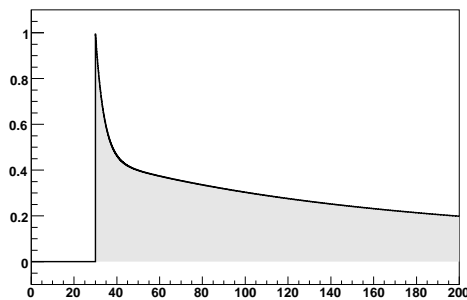
```

```

22  return
23      t*A1/tau1*exp(-t/tau1)
24      +
25      A2/tau2*exp(-t/tau2);
26
27  }
28
29  /*****
30  /*****
31  class Func_TExpexp: public FitSignal{
32  public:
33      // create function
34      Func_TExpexp(): FitSignal(func_texpexp, // a name
35          func_texpexp, // the C function
36          5) // the number of parameters
37      {
38          //set names
39          SetFuncName(A1*t*exp(-t/#tau 1)+A2*exp(-t/#tau 2));
40          SetParName(0,T0);
41          SetParName(1,tau 1);
42          SetParName(2,tau 2);
43          SetParName(3,Area 1);
44          SetParName(4,Area 2);
45      }
46  };
47  /*****

```

6.7.8 Funzione Scintillatore a 3 componenti



Funzione del tipo:

$$i(t) = \begin{cases} \frac{A1}{\tau_1} \cdot t \cdot \exp\left(-\frac{t-t_0}{\tau_1}\right) + \frac{A2}{\tau_2} \exp\left(-\frac{t-t_0}{\tau_2}\right) + \frac{A3}{\tau_3} \exp\left(-\frac{t-t_0}{\tau_3}\right) & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

```

1  /******
2  func_expexpexp.C: definition of a exp+exp function
3  *****/
4
5  double func_expexpexp(double *x, double *pars)
6  {
7      // the time variable:
8      double t=x[0];
9
10     // parameters:
11     double t0      =pars[0];
12     double tau1    =pars[1];
13     double tau2    =pars[2];
14     double tau3    =pars[3];
15     double A1      =pars[4];
16     double A2      =pars[5];
17     double A3      =pars[6];
18
19     // function definition:
20     if(t<t0)
21         return 0;
22     t= t-t0;
23
24     return
25         A1/tau1*exp(-t/tau1)
26         +
27         A2/tau2*exp(-t/tau2)
28         +
29         A3/tau3*exp(-t/tau3)
30         ;
31
32 }
33
34 *****/
35 *****/
36 class Func_Expexpexp: public FitSignal{
37 public:
38     // create function
39     Func_Expexpexp(): FitSignal(func_expexpexp,// a name
40         func_expexpexp, // the C function
41         7) // the number of parameters
42     {
43         //set names
44         SetFuncName(A1*exp(-t/#tau 1)+A2*exp(-t/#tau 2)+A3*exp(-t/#tau
35)
45         SetParName(0,T0);
46         SetParName(1,tau 1);
47         SetParName(2,tau 2);
48         SetParName(3,tau 3);
49         SetParName(4,Area 1);
50         SetParName(5,Area 2);

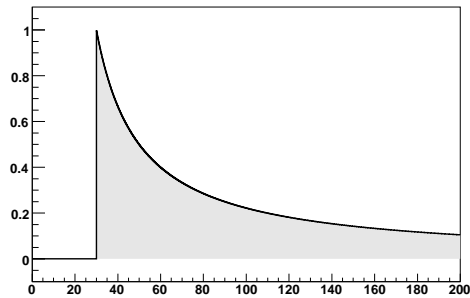
```

```

51     SetParName(6,Area 3);
52   }
53 };
54 /*****

```

6.7.9 Funzione $1/(1 + t/\tau)$



Funzione del tipo:

$$i(t) = \begin{cases} A / [1 + \frac{t-t_0}{\tau}] & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

Questa funzione puo' essere utilizzata per schematizzare la risposta di rivelatori come ad esempio un contatore a gas a geometria cilindrica:

```

1  /*****
2  func_log.C: definition of a A*log(1+t/tau) function
3  *****/
4
5  double func_log(double *x, double *pars)
6  {
7    // the time variable:
8    double t=x[0];
9
10   // parameters:
11   double t0   =pars[0];
12   double tau  =pars[1];
13   double A    =pars[2];
14
15   // function definition:
16   if(t<t0)
17     return 0;
18   t= t-t0;
19
20   return
21     A *1/(1+t/tau);

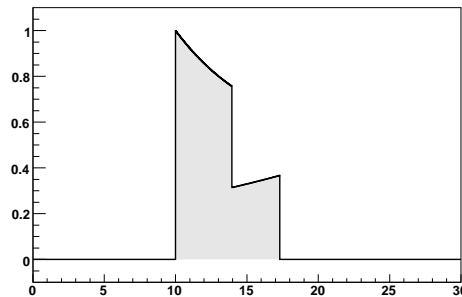
```

```

22 }
23
24 /**
25 /**
26 class Func_Log: public FitSignal{
27 public:
28 // create function
29 Func_Log(): FitSignal(func_log,// a name
30     func_log, // the C function
31     3) // the number of parameters
32 {
33 //set names
34 SetFuncName(1/(1+t/#tau));
35 SetParName(0,T0);
36 SetParName(1,tau);
37 SetParName(2,A);
38 }
39 };
40 /**

```

6.7.10 Funzione “silicio”



La funzione è somma dei contributi di lacune e elettroni:

$$i(t) = \begin{cases} A \cdot [i_e(t - t_0) + i_h(t - t_0)] & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

dove:

$$i_e(t) = \frac{e\mu_e b}{d} (a + bx_0) \exp(-\mu_e bt) \quad \text{if } t < t_e$$

$$i_h(t) = \frac{e\mu_h b}{d} (a + bx_0) \exp(+\mu_h bt) \quad \text{if } t < t_h$$

con:

$$t_e = \frac{1}{\mu_e b} \log \frac{a + bx_0}{a}$$

$$t_h = \frac{1}{\mu_h b} \log \frac{a + bd}{a + bx_0}$$

Questa funzione tiene conto della diversa velocità di drift di lacune e elettroni e della geometria del campo elettrico per un rivelatore a silicio. $d - x_0$ e' la profondita' del punto di interazione, d lo spessore del rivelatore. Il campo elettrico viene calcolato a partire dalla tensione applicata, tensione di svuotamento, e spessore del rivelatore.

```

1 #ifndef _FUNC_SI_
2 #define _FUNC_SI_
3 *****
4 func_silicio.C: function x silicio
5 *****/
6
7 double func_silicio(double *x, double *pars)
8 {
9     // the time variable:
10    double t=x[0];
11
12    // parameters:
13    double t0      =pars[0];
14    double x0      =pars[1];
15    double A       =pars[2];
16    double Vappl   =pars[3]; // fixed, non viene fittato, V
17    double Vdepl   =pars[4]; // fixed, non viene fittato, V
18    double d       =pars[5]; // fixed, non viene fittato, um
19
20    if(t<t0)
21        return 0;
22    t -= t0;
23
24    t *= 1e-9; // conversione sec <--> ns
25    x0 *= 1e-6; // conversione um <--> m
26    d  *= 1e-6; // conversione um <--> m
27
28    const double a=-(Vappl+Vdepl)/d;
29    const double b=2*Vdepl/(d*d);
30
31    //-- valori dello Knoll --
32    const double mu_h=480 *1e-4; // da cm**2/(V s) a m**2/(V s)
33    const double mu_e=1350*1e-4; // da cm**2/(V s) a m**2/(V s);
34
35
36
37
38    double tend_e =- 1/(mu_e*b) * log( (a+b*d)/(a+b*x0) );
39    double tend_h =- 1/(mu_h*b) * log( (a+b*x0)/a );
40
41    // corrente:
42    double i=0;
43    if(t<tend_e)
44        i += - mu_e*(a+b*x0)*exp(-t*mu_e*b);
45    // i += mu_e*(a+b*x0)*exp(+t*mu_e*b);

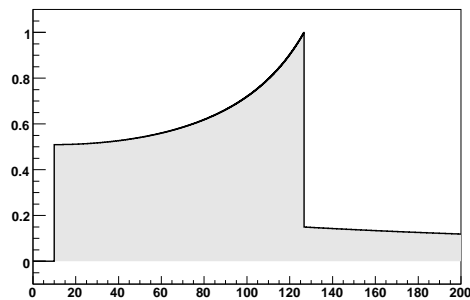
```

```

46  if(t<tend_h)
47      i += - mu_h*(a+b*x0)*exp(+t*mu_h*b);
48      //  i += mu_h*(a+b*x0)*exp(-t*mu_h*b);
49      return A*i;
50  }
51
52  /***/
53  /***/
54  class Func_Silicio: public FitSignal{
55  public:
56      // create function
57      Func_Silicio(): FitSignal(func_silicio,// a name
58          func_silicio, // the C function
59          6) // the number of parameters
60      {
61          //set names
62          SetFuncName(silicio);
63          SetParName(0,T0);
64          SetParName(1,Int.depth (um));
65          SetParName(2,Amplitude);
66
67          SetParName(3,Vappl (V));
68          SetParName(4,Vdepl (V));
69          SetParName(5,Thickness (um));
70
71          FixParameter(3, 300);
72          FixParameter(4, 120);
73          FixParameter(5, 300);
74
75          SetParLimits(1,0,1000);
76      }
77  };
78  /***/
79  #endif

```

6.7.11 Funzione “germanio coassiale”



La funzione è somma dei contributi di lacune e elettroni:

$$i(t) = \begin{cases} A \cdot [i_e(t - t_0) + i_h(t - t_0)] & \text{if } t > t_0 \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

dove:

$$i_e(t) = \frac{ev_e}{\log(r_2/r_1)} \frac{1}{r_0 - v_e t} \quad \text{if } t < t_e$$

$$i_h(t) = \frac{ev_h}{\log(r_2/r_1)} \frac{1}{r_0 + v_h t} \quad \text{if } t < t_h$$

con:

$$t_e = \frac{r_0 - r_1}{v_e}$$

$$t_h = \frac{r_2 - r_0}{v_h}$$

Questa funzione tiene conto della diversa velocità di drift di lacune e elettroni per un rivelatore a germanio coassiale. r_0 è il raggio del punto di interazione, v_e la velocità degli elettroni (in cm/s, assunta costante). La velocità delle lacune è' assunta essere circa il 90% di quella degli elettroni.

```

1  /*****
2  func_germanio2.C: function x germanio
3  *****/
4
5  double func_germanio(double *x, double *pars)
6  {
7      // the time variable:
8      double t=x[0];
9
10     // parameters:
11     double t0      =pars[0];
12     double r0      =pars[1];
13     double A       =pars[2];
14     double ve      =pars[3];
15     const double gr1 =pars[4];
16     const double gr2 =pars[5];
17     if(t<t0)
18         return 0;
19     t -= t0;
20
21
22     double vh=ve;;
23
24     double tend_e=(r0-gr1)/ve;
25     double tend_h=(gr2-r0)/vh;
26
27     t *= 1e-9; // conversione sec <--> ns
28

```

```

29 // corrente:
30 double i=0;
31 if(t<tend_e)
32 {
33     i +=ve/(1-ve*t/r0);
34 }
35 if(t<tend_h)
36 {
37     i += vh/(1+vh*t/r0);
38 }
39 return A*i/r0;
40 }
41
42 /*****
43 /*****
44 class Func_Germanio: public FitSignal{
45 public:
46     // create function
47     Func_Germanio(): FitSignal(func_germanio, // a name
48         func_germanio, // the C function
49         4+2) // the number of parameters
50 {
51     //set names
52     SetFuncName(Germanio);
53     SetParName(0,T0);
54     SetParName(1,Int.Radius (cm));
55     SetParName(2,Amplitude);
56     SetParName(3,El. vel. (cm/s));
57
58     SetParName(4,R1 (cm)); FixParameter(4, 0.3);
59     SetParName(5,R2 (cm)); FixParameter(5, 3);
60
61     SetParLimits(1,0.3,3);
62     SetParLimits(3,1e5,1e8);
63     FixParameter(3, 9e6);
64
65 }
66 };
67 /*****

```

6.8 Altre funzionalità

Save signal on disk

I segnali acquisiti vengono salvati in formato .CSV (Comma Separated Value), un formato di testo molto diffuso e leggibile o importabile dalla grande maggioranza dei programmi. È quindi poi possibili ricaricarli all'interno del programma o esportarli su altri computer.

Print canvas on file

Una copia del canvas corrente (contente anche l'im-

immagine della funzione fittata, se presente) viene salvata su disco in un formato grafico a scelta tra .EPS, .PS, .GIF.

Print canvas on Lab Printer

Il canvas corrente viene stampato direttamente sulla stampante del laboratorio.

7 Approfondimenti e bibliografia

Per quanto riguarda le informazioni tecniche sulla realizzazione di sistemi di campionamento analogico-digitale, si rimanda direttamente ai siti WEB dei costruttori, ad esempio Analog Devices [16], MAXIM [17] e altri, con particolare riferimento alle "Application Notes". Per una breve descrizione delle caratteristiche generali del funzionamento degli ADC, vedi [1, 4]. Per una introduzione dal carattere poco formale all'analisi dei dati digitalizzati si rimanda a [18]. Per una analisi matematicamente rigorosa si rinvia invece a [2]. Le varie proprietà dei rivelatori e la loro analisi mediante metodi analogici sono descritti in [10].

Riferimenti bibliografici

- [1] Analog Devices. Fundamentals of sampled data systems, application note an282. Technical report, <http://www.analog.com>.
- [2] A.V. Oppenheim and R.W. Schaffer. *Digital Signal Processing*. Prentice Hall, 1975.
- [3] Analog Devices. Adc selection guide. Technical report, <http://www.analog.com>.
- [4] MAXIM. Application note an2094. Technical report, <http://pdfserv.maxim-ic.com/en/an/AN2094.pdf>.
- [5] G.Poggi. *Esperimenti di Eletticità e Magnetismo*. Dipartimento di Fisica, Università di Firenze, 1988.
- [6] MAXIM. Demystifying sigma-delta adcs, application note an1870. Technical report, <http://pdfserv.maxim-ic.com/en/an/AN1870.pdf>.
- [7] MAXIM. Understanding sar adcs, application note an1080. Technical report, <http://pdfserv.maxim-ic.com/en/an/AN1080.pdf>.
- [8] MAXIM. Understanding pipelined adcs, application note an1023. Technical report, <http://pdfserv.maxim-ic.com/en/an/AN1023.pdf>.
- [9] MAXIM. Understanding flash adcs, application note an810. Technical report, <http://pdfserv.maxim-ic.com/en/an/AN810.pdf>.

- [10] Glenn F. Knoll. *Radiation detection and measurement*. Wiley, 2. ed. edition, 1989.
- [11] <http://root.cern.ch> The ROOT analysis framework.
- [12] G.Cowan. *Statistical Data Analysis*. Clarendon Press, 1998.
- [13] E.Brandt. *Statistical and Computational Methods in Data Analysis*. North Holland, 1983.
- [14] CERN. Minuit, function minimization and error analysis reference manual. Technical report, <http://wwwasdoc.web.cern.ch/wwwasdoc/minuit/minmain.html>.
- [15] Particle Data Group. Statistics. <http://pdg.lbl.gov/2005/reviews/strpp.pdf>.
- [16] Analog Devices. Technical report, <http://www.analog.com>.
- [17] MAXIM. Technical report, <http://www.maxim-ic.com>.
- [18] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. available on <http://www.dspguide.com/>.