

theremino
•the•real•modular•in-out•

Sistema theremino

Minimizzare l' FWHM

**Metodi per la riduzione della larghezza
delle righe con cristalli scintillatori
Nai(Tl), Csl(Tl) e BGO**

Ringraziamenti

Prima di tutto è necessario spiegare come mai in questo documento si usa la forma plurale "noi" ed è importante evidenziare che i risultati ottenuti sono frutto di un lavoro comune. Molti hanno contribuito alle ricerche, primo fra tutti **Alessio Giusti** che è stato una continua fonte di riferimento e che ha svolto buona parte degli esperimenti pratici. Ma anche **Roberto Cena** che ha aiutato a mettere in piedi le basi del sistema Theremino, senza poi dimenticare **Andrea Bosi** che ha contribuito con test del software, interessanti idee e utili suggerimenti. Ma anche **Steven Sesselman** di **Gamma Spectacular** che ha molto stimolato il miglioramento di FWHM. Nell'ultimo anno si è anche aggiunto **Nobu Komagata** che sta trasformando l'applicazione ThereminoMCA in un software iper-professionale. Poi ci sono centinaia di **sperimentatori** che per motivi di privacy non possiamo elencare qui, ma che hanno contribuito con un gran numero di prove, e ci hanno inviato molte immagini e suggerimenti. Infine c'è **Livio Cicala** (io), che ha fatto gran parte del "lavoro sporco" sintetizzando le idee e i consigli di tutti.

Abstract

Questo documento spiega i metodi che abbiamo individuato negli anni 2012 e 2013 per minimizzare il valore di FWHM dei normali cristalli scintillatori reperibili con prezzi ragionevoli. Questi cristalli sono a volte preferibili ai sistemi HPGe (Germanio iper-puro) non solo per il prezzo ma anche per poter usare sistemi leggeri e portatili (senza bombola di azoto liquido)

I cristalli scintillatori usati normalmente, Nai(Tl), Csl(Tl) e BGO, hanno una risoluzione scarsa, quasi al limite dell'usabile. Con questi sistemi è difficile distinguere le righe di isotopi con energie simili. Inoltre l'allargamento delle righe crea un tappeto di fondo che fa completamente sparire le righe basse, dovute a isotopi con bassa concentrazione e attività.

Anche minimi incrementi della risoluzione producono grandi benefici sulla visibilità delle righe quindi inizieremo con il mostrare i notevoli risultati raggiunti per poi procedere con la spiegazione delle tecniche che li hanno permessi.

Risoluzione dei cristalli e miglioramenti ottenuti

La documentazione reperibile in rete fino ad oggi considera come minimi i seguenti valori di FWHM (1):

- NaI (Tl) = 7%
- CsI (Tl) = 8%
- BGO = 11%

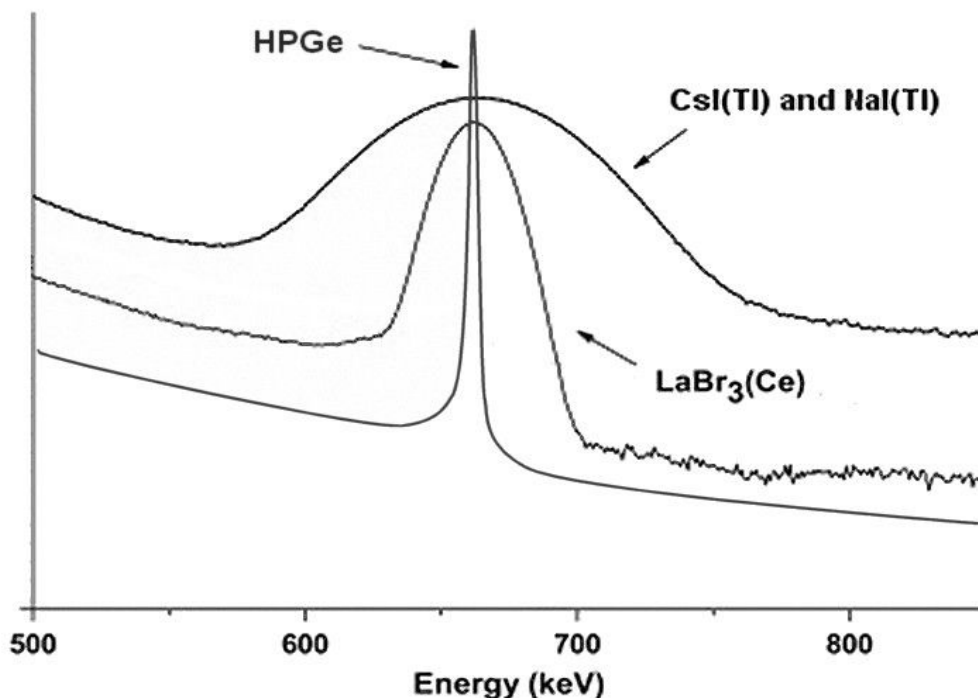
Se si applicano tutte le tecniche che presentiamo in questo documento si possono raggiungere i valori seguenti (e anche qualcosa di meglio secondo i dati di alcuni sperimentatori):

- NaI (Tl) = 5%
- CsI (Tl) = 6%
- BGO = 8%

(Nota 1) In questa pagina e in tutto questo documento i valori FWHM sono sempre riferiti al Cs-137 da 662 KeV e sono valori approssimativi, non prendeteli come costanti naturali. A seconda del software, dell'hardware, del cristallo e di come lo si accoppia con il PMT, i valori effettivi possono differire da quelli indicati, anche di un punto percentuale.

Le risoluzioni a confronto

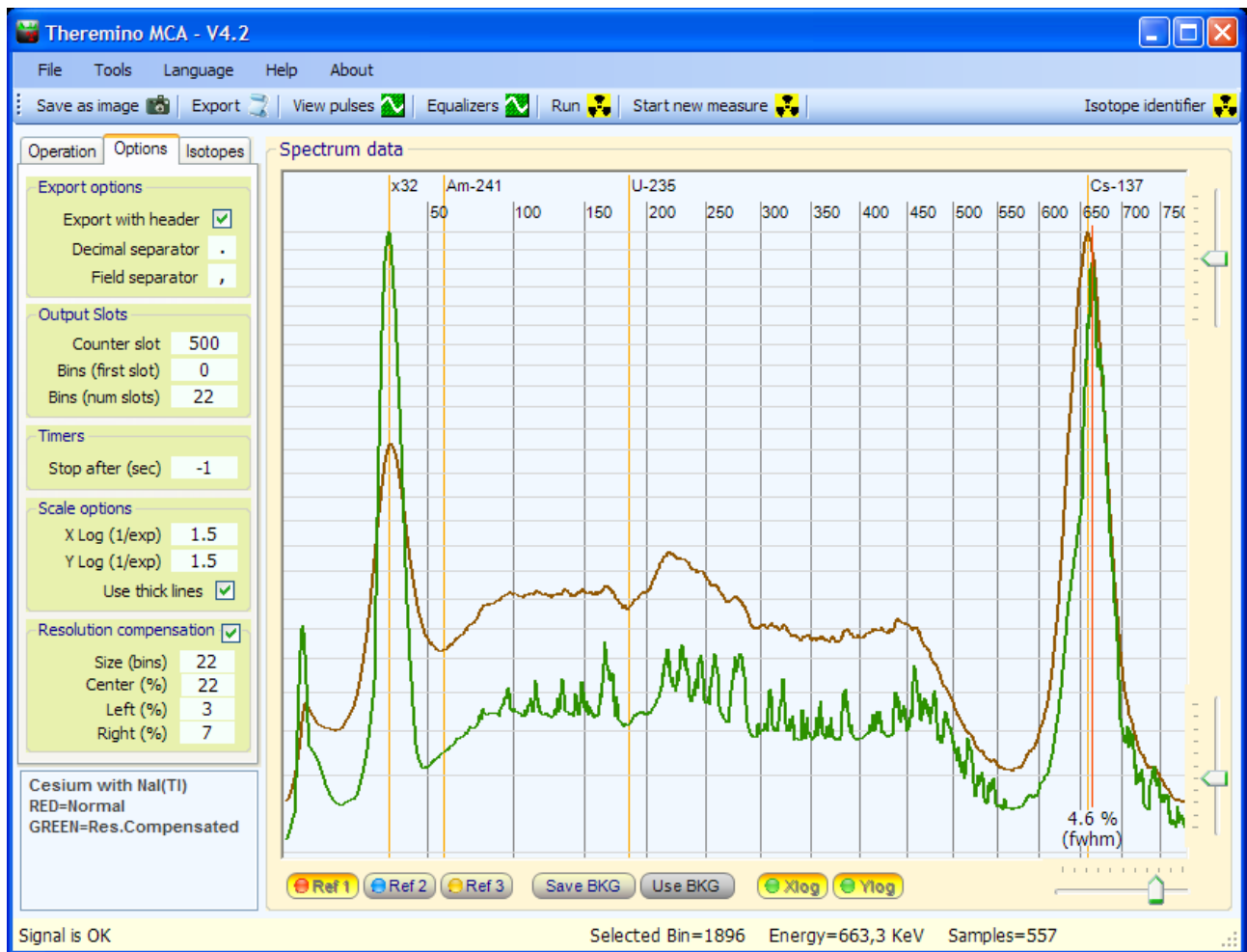
Questo è l'aspetto approssimativo delle righe prodotte dai principali tipi di rivelatori.



Le tecniche che proponiamo permettono di **ottenere con i normali scintillatori NaI(Tl) approssimativamente la stessa risoluzione dei ben più costosi LaBr₃(Ce)** In via teorica il nostro software con una sonda LaBr₃ potrebbe arrivare verso il 2% di FWHM e, con i futuri miglioramenti al compensatore di risoluzione che stiamo pianificando, forse si potrebbe andare verso un incredibile 1.5 %

Diminuire il valore di FWHM oltre a stringere le righe fa spuntare dal tappeto di fondo anche le righe più basse, come visibile nelle immagini delle prossime tre pagine.

Un esempio con il Cesio

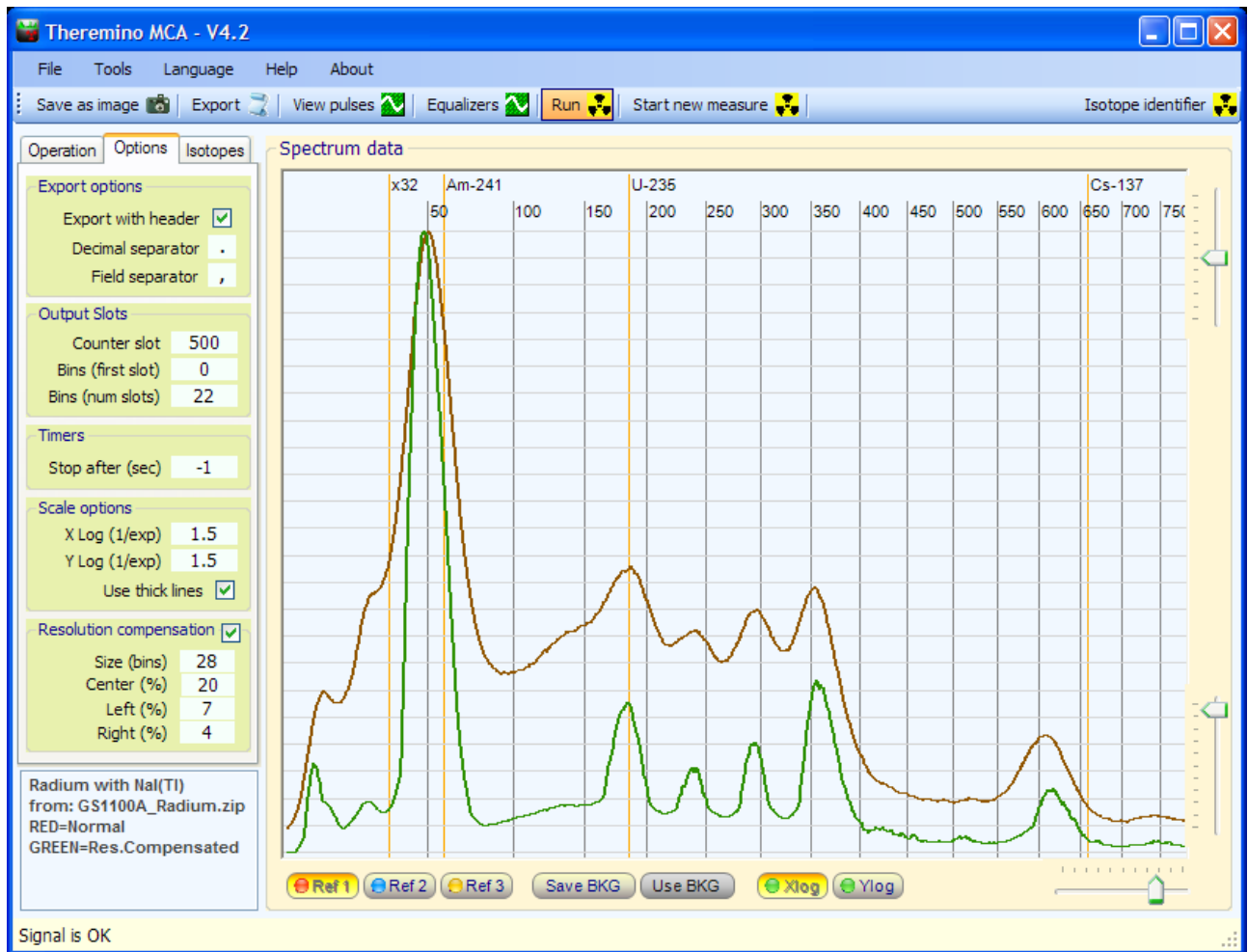


Qui si vede un campione di Cs-137 misurato con la migliore sonda Nai(Tl) a nostra disposizione (riga rossa) Il valore di FWHM relativo (non visibile in questo grafico) era del 6%

Senza poi muovere la sonda e il campione ma solo abilitando l'algorithmo di "Compensazione della risoluzione" si è ottenuto il grafico con un FWHM del 4.6% (riga verde)

Il pannello di controllo del "compensatore di risoluzione dei cristalli" è visibile nella colonna di sinistra, in basso.

Un esempio con il Radio



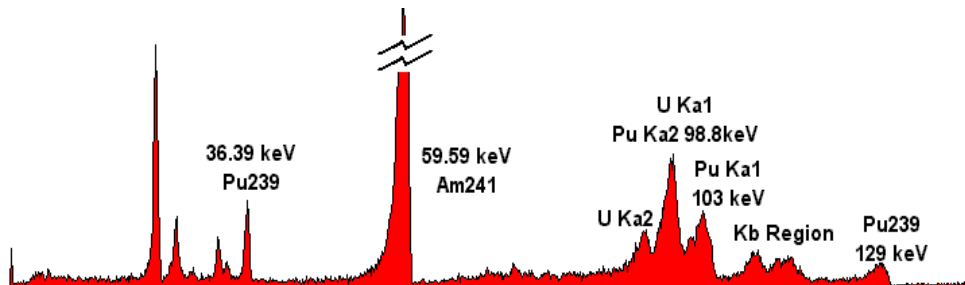
Qui si vede un campione di Ra-226 misurato con una sonda dalla risoluzione mediocre (riga rossa)

Lo stesso file di dati è poi stato letto nuovamente ma con l'algoritmo di "Compensazione della risoluzione" attivato (riga verde)

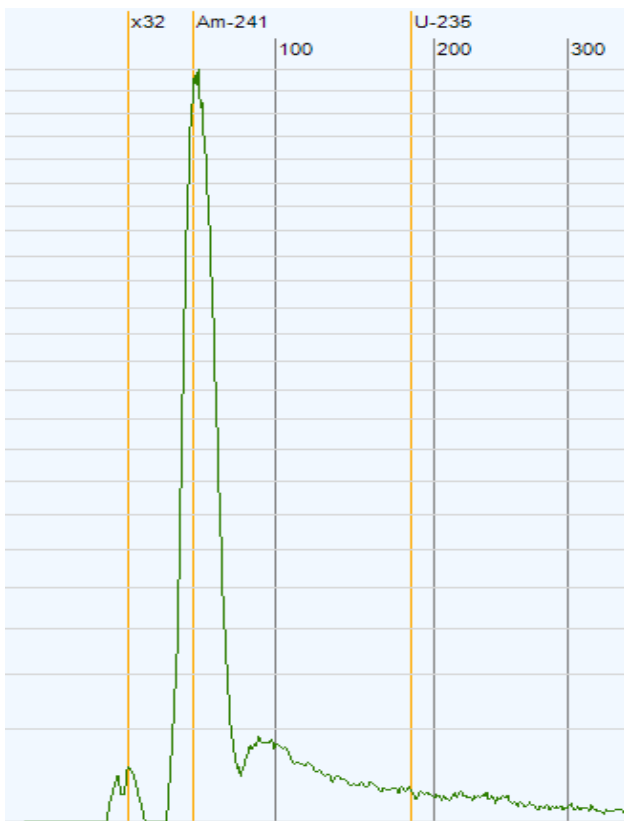
Qui non abbiamo abilitato il misuratore di FWHM, i valori di FWHM del Radio sono sempre molto alti e non si usano normalmente.

Un esempio con l'Americio

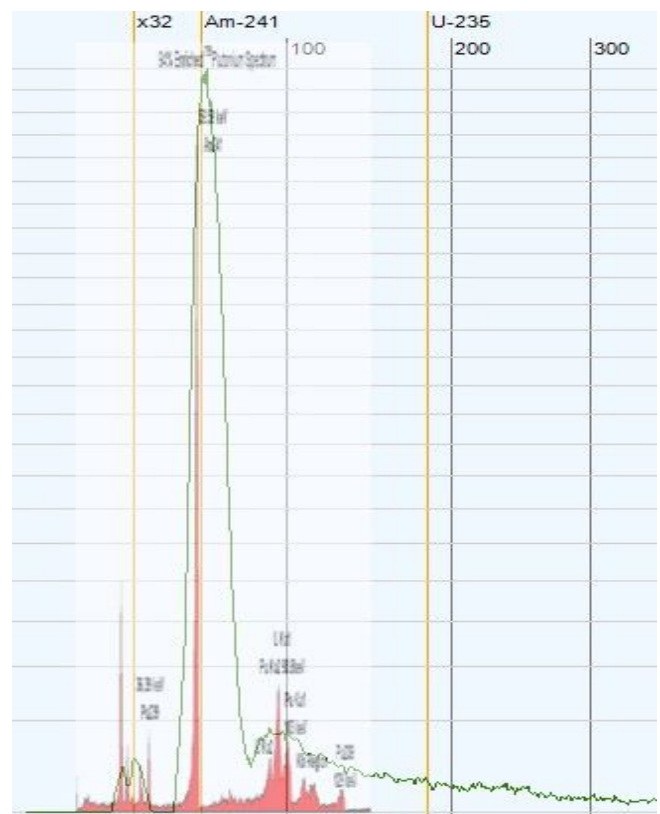
Queste immagini fatte da Paolo (happynewgeiger) mostrano che il Thermano MCA riesce a risolvere parzialmente anche le deboli righe del Pu-239 (e i raggiX prodotti dagli shell Ka1 e Ka2 di Plutonio e Uranio) che si trovano alla base della fortissima riga dell'Am-241.



Come si vedono l'Americio e le righe adiacenti con i sistemi HPGe.



Come si vedono con Nai(Tl) e ThermanoMCA



La sovrapposizione dei due grafici.

Come "raccattare" due punti di FWHM

Per i dettagli sui punti da 1 a 16 vedere le appendici da 1 a 16

- 1) Cablaggio del PMT a bassa corrente, singolo condensatore e niente cablaggi strani con zener e transistor.
- 2) Alimentazione da USB, niente alimentatori separati, niente loop di masse.
- 3) Filtro di condizionamento del segnale che crea impulsi da circa 100 uS con punta arrotondata (gaussiani)
- 4) Cancellazione del polo zero (per non far dondolare la linea di zero)
- 5) Pre amplificazione del segnale di circa 30dB dopo i filtri e prima dell'ADC
- 6) Cura maniacale nell'abbassare il rumore (max 500uV p.p, quindi niente circuiti "affollati" e pieni di fili)
- 7) Filtro passa alto prima dell'ADC (elimina il rumore a bassa frequenza e crea gli impulsi bipolari)
- 8) ADC a 16 bit (per il nostro approccio "tutto software" 14 bit sono il minimo indispensabile)
- 9) Alta frequenza di campionamento (almeno 20 campioni per ogni impulso)
- 10) Da qui in poi tutto software ad alto livello. Niente DSP, sistemi embedded, e hardware vario.
- 11) Misurazione precisa della punta dell'impulso (meglio dei SampleHold hardware su impulsi "difficili")
- 12) Compensazione dell'offset della linea di zero con BaseLine Restoring.
- 13) Individuazione dei PileUp.
- 14) Individuazione degli impulsi troppo "disturbati" (da rumore, ringing, pileup o linea di zero imprecisa)
- 15) Individuazione del numero, quantità e tipo di errori e display di controllo della forma degli impulsi.
- 16) Compensazione della risoluzione dei cristalli.

Le tecniche indicate da (1) a (15) hanno permesso di ridurre lo FWHM di circa un punto. Fino alla fine del 2012 (prima della versione 4.0 di ThereminoMCA) si riusciva ad arrivare verso il 6% di FWHM e nessun ulteriore miglioramento sembrava possibile (non molto diverso da quanto indicato nella letteratura)

La tecnica (16) "Compensazione della risoluzione dei cristalli", dalla versione 4.0 in poi (inizi del 2013) ha prodotto da sola un altro miglioramento di un intero punto.

Quali di queste tecniche influiscono sullo FWHM

Tutti i punti elencati influiscono sul valore di FWHM.

La documentazione "teorica" potrebbe far pensare che solo alcune di queste tecniche abbiano una relazione con il valore di FWHM ma in pratica (con impulsi reali e disturbati) il minimo FWHM si ottiene solo ottimizzando tutta la catena, dal raggio gamma fino al grafico.

Anche i particolari non direttamente collegabili allo FWHM influiscono, in quanto propedeutici per le tecniche successive, che altrimenti non potrebbero funzionare bene.

Per dimostrare queste affermazioni basta disabilitare, una per una, varie aree del software ThereminoMCA e vedere i cambiamenti di FWHM che ne conseguono. Per il rumore, il numero di bit dell'adc e gli altri fattori che non sono modificabili da software esistono metodi esterni al ThereminoMCA. Ad esempio aggiungere quantità crescenti di rumore bianco con la applicazione Theremino_AudioExamples.

Appendice 1 - Cablaggio del PMT

Se non si seguono questi consigli il rumore aumenta e il valore di FWHM aumenta di conseguenza.

Bassa corrente

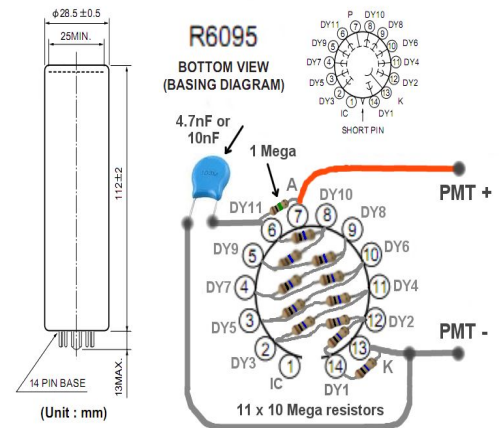
Solo con PMT a bassa corrente (resistori da 10 Mega) è possibile filtrare in modo adeguato e ridurre il rumore al minimo.

Ma in passato i produttori di sonde hanno creato molta confusione in questo campo, usando inutilmente resistori di valore molto basso (470 Kohm) che minimizzano le non linearità di misura fino a un improbabile 50000 CPS e oltre (lo specificano nelle caratteristiche).

Questo tipo di progettazione risale a quando i PMT venivano utilizzati per l'astronomia dove la luce delle stelle produceva una **corrente media migliaia di volte superiore** a quella prodotta dagli impulsi strettissimi che usiamo noi. Inoltre per usi astronomici era importante una linearità molto alta (una frazione dell'1% in modo da classificare correttamente la magnitudine delle stelle)

Ma nella spettrometria precisioni così spinte sono inutili e vengono vanificate dalla larghezza delle righe e dagli errori di linearità dei cristalli stessi, che superano il 10% (e anche il 30% alle basse energie)

Comunque se si dovessero misurare 50000 CPS il nostro consiglio non sarebbe di linearizzare la sonda ma di allontanarsi al più presto.



Misurare sorgenti molto radioattive con sonde a bassa corrente

Anche le sonde cablate a bassa corrente secondo le nostre indicazioni possono misurare sorgenti molto "calde" e produrre ugualmente spettri perfettamente lineari, si procede così:

- 1) Si posiziona la sonda vicino al campione e si misurano velocemente i CPS (pochi secondi) in questa misura si ottiene il valore di un conteggio e la linearità non conta.
- 2) Ci si allontana quanto basta per portare i CPS a meno di 500 (anche molti metri se la sorgente è di grandi dimensioni) e questo è un bene sia per la sonda che per la salute di chi misura.
- 3) Si attende il completarsi dello spettro stando a distanza di sicurezza. Lo spettro risultante sarà molto più lineare di quello che sarebbe ottenuto con 50000 CPS e sonda con resistori da 470k

Condensatori di filtro della alta tensione

La doppia cella di filtro del PmtAdapter usa già abbondanti e rari condensatori da 47 nF a 1500 Volt. Per un PMT con resistori da 470k questi condensatori dovrebbero essere da un microfarad.

Non si tratta solo di costo, condensatori da 1uF a 1500Volt sarebbero enormi e anche molto pericolosi.

Niente componenti "strani"

Cablaggi con transistor e zener possono linearizzare la risposta anche con corrente del PMT al limite della saturazione, ma in spettrometria questi accorgimenti non servono. In compenso creano rumore, maggiore consumo di corrente e altri difetti.

Errori di questo genere (ad esempio linearizzare a tutti i costi) sono tipici della progettazione "scolastica" che a volte sovrastima l'importanza di alcuni particolari a scapito delle prestazioni globali.

Appendice 2 - Alimentazione da USB

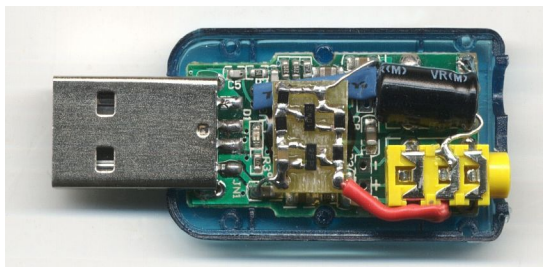
Se non si seguono questi consigli il rumore aumenta, disturba le ampiezze degli impulsi, e fa aumentare il valore di FWHM

Niente alimentazione 5 Volt separata

Se non si usa la tensione proveniente dalla USB, che segue il segnale lungo lo stesso cavo, si creano inevitabili loop di massa che provocano un aumento del rumore inaccettabile.

Usare una scheda audio modificata con alimentazione e filtro passa alto

Solo modificando una scheda audio come indicato nel file PmtAdapters si ottiene di portare la alimentazione lungo lo stesso cavo del segnale, eliminando così i loop di massa. Naturalmente non è possibile fare queste modifiche sulla scheda audio della motherboard e nemmeno su una costosa SoundBlaster.



Non ci si faccia ingannare dal loro basso costo, queste schede campionano a 192 KHz e hanno un rapporto segnale rumore di oltre 100 dB.

Per quello che serve a noi sono perfette e anche migliori di molte schede da 100 Euro e più.

Niente alimentatori "professionali"

Alimentatori collegati dalla rete elettrica, provocano inevitabili disturbi a 50Hz sul delicatissimo collegamento che porta il segnale alla scheda audio.

Inoltre questi alimentatori sono fatti per tutt'altri usi, forniscono 1mA contro i pochi uA che servono, ma in compenso hanno un rumore intollerabilmente alto. *Secondo le specifiche del costruttore questo alimentatore ha un ripple di "0.01% of full scale" quindi: $10KV / 100 * 0.01 = 1 \text{ Volt pep}$ (diecimila volte maggiore del nostro)*

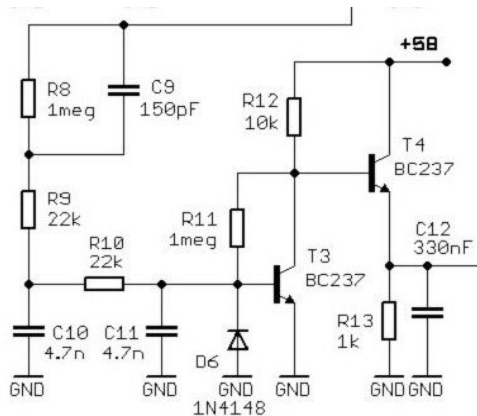


In questa immagine si vede un bellissimo alimentatore professionale da 1 mA a 10 KVolt (che fanno 10 Watt)

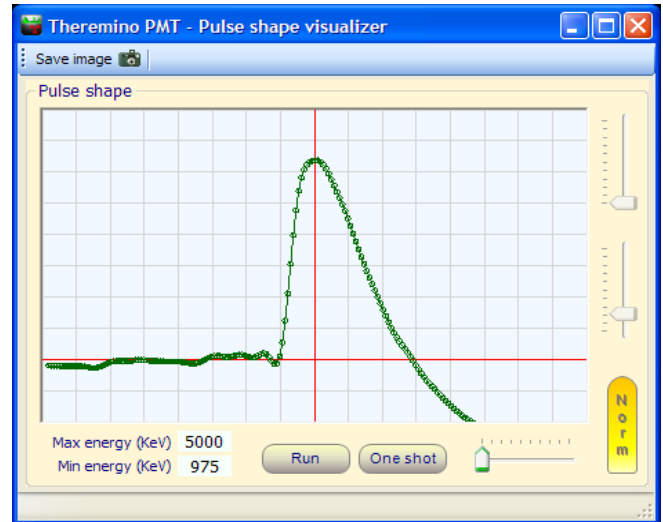
Ottimo per le esecuzioni capitali, un po' meno per un tubo PMT, appena si sbaglia "kaputt" (kaputt PMT o kaputt umano a seconda dell'errore che si fa)

Appendice 3 - Il condizionamento del segnale

Il filtro di condizionamento del segnale crea impulsi da circa 100 μs con punta arrotondata (gaussiani)



L'allargamento dell'impulso a oltre 100 μs permette di campionare l'impulso con molti punti e fare misure precise.

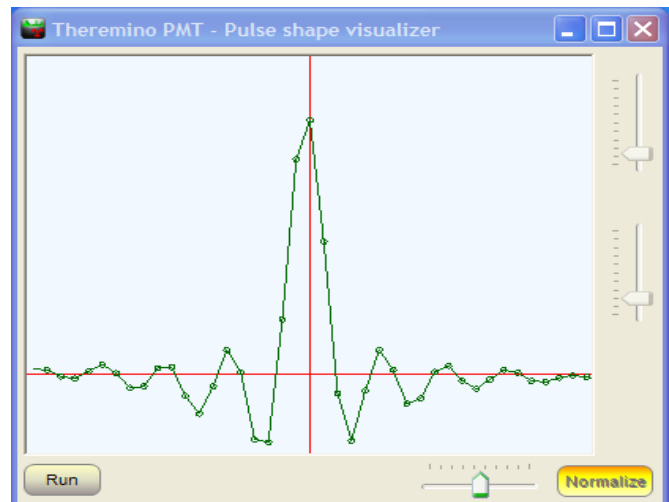


Nell'immagine qui a destra si vede invece un impulso molto stretto, da 5 μs (la scala orizzontale è allargata di venti volte)

Si noti che non è possibile individuare l'altezza della sua punta con precisione.

Un impulso con fronti ripidi genera anche molto "ringing" (le ondulazioni che precedono e seguono l'impulso) e il "ringing" rende difficile misurare con precisione la posizione dello zero.

Per misurare questo impulso con la stessa precisione dell'immagine precedente si dovrebbe usare un ADC con frequenza venti volte maggiore del nostro, quindi circa 4 MHz.

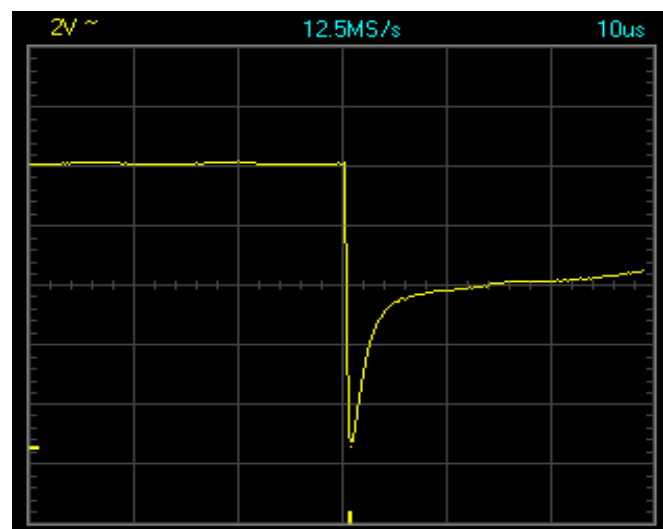


Qui invece si vede l'impulso originale di un PMT senza filtri di allargamento.

Molti MCA hardware usano questi impulsi direttamente pensando che basti usare un ADC veloce per poter misurare impulsi stretti, questo è vero ma quanto dovrebbe essere veloce?

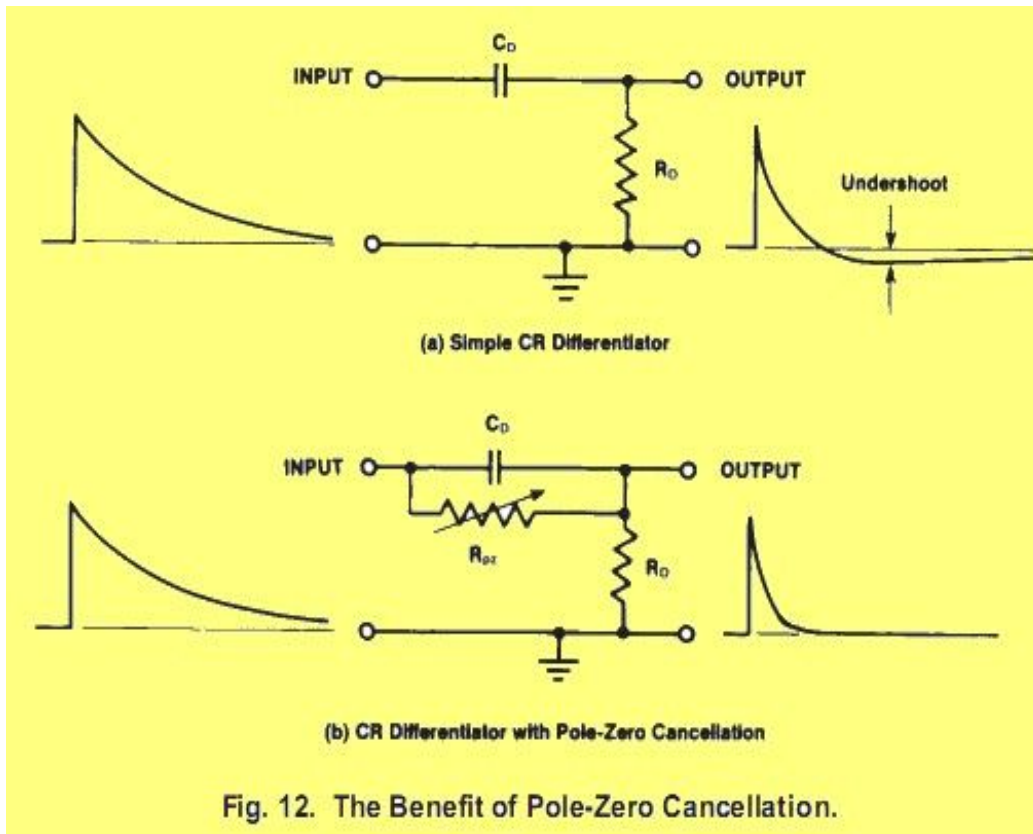
L'impulso è stretto e la punta è a spillo (non con aspetto gaussiano) qui appare da 1 μs , ma la parte a spillo della punta dovrebbe essere da 0.5 μs o meno.

Per avere la stessa nostra precisione ci vorrebbe un ADC da 40 MHz (i nostri 198 KHz per 100 μs divisi per 0.5 μs) Ma nessun sistema per la spettrometria gamma usa ADC da 40MHz!



Appendice 4 - Cancellazione del polo "zero"

La cancellazione del polo zero minimizza il dondolio della linea di zero e facilita il compito del Baseline Restorer.



La "Pole-Zero Cancellation", ben descritta nella letteratura sull'argomento, elimina lo "Undershoot" e produce un impulso che torna a zero nel più breve tempo possibile.

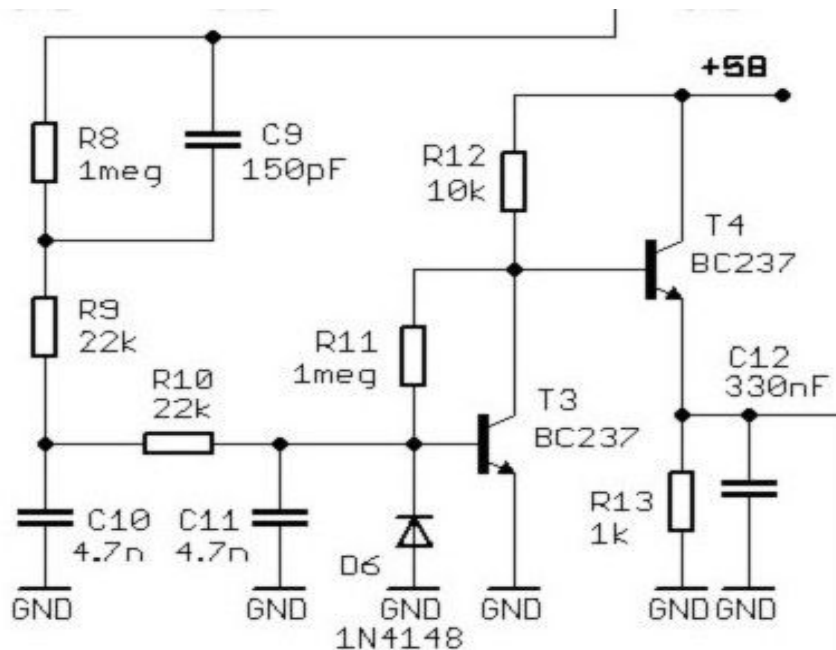
Avere impulsi senza undershoot è un prerequisito essenziale perché il filtro passa alto possa poi produrre impulsi perfettamente "bipolari"

Il filtro passa alto e gli impulsi bipolari verranno descritti nelle pagine seguenti.

Appendice 5 - Preamplificazione

Il segnale proveniente dal PMT dopo aver passato i filtri e prima di andare all'ADC deve venire amplificato di circa 30 dB e abbassato di impedenza.

Senza il preamplificatore non si potrebbe sfruttare tutta la dinamica dell'ADC, il rumore aumenterebbe e farebbe aumentare il valore di FWHM



Il transistor T3 amplifica il segnale che si è abbassato molto nel filtro.

Il transistor T4 e il condensatore da 330nF abbassano l'impedenza d'uscita, minimizzando il rumore che il conduttore del segnale può raccogliere per accoppiamento con il conduttore di alimentazione e con i disturbi esterni provocati dalla rete elettrica.

Questo preamplificatore riporta il segnale al livello più adatto per l'ADC.

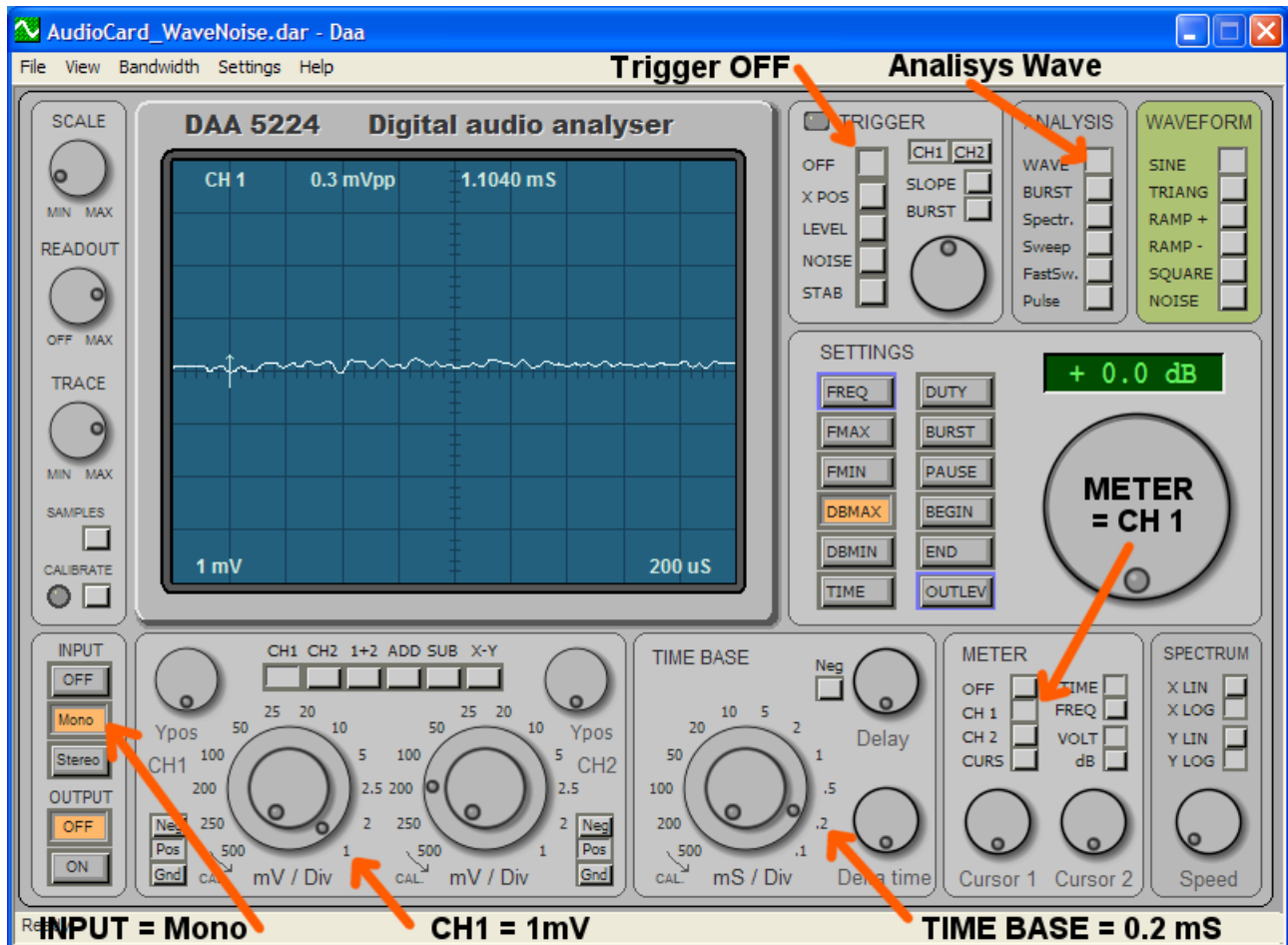
Se il segnale supera 1 Volt l'ADC satura se invece è troppo basso si perde dinamica e aumenta il rumore.

Con la giusta regolazione del valore dell'alta tensione, e quindi della amplificazione del PMT, si massimizza la dinamica e si ottiene il massimo rapporto tra segnale e rumore possibile (qualcosa di più di 100 dB)

Appendice 6 - Il rumore

Come spiegato ripetutamente è necessaria grande cura nell'abbassare il rumore, quindi niente circuiti "affollati" e pieni di fili e niente DSP, scale mobili o altre diavolerie hardware.

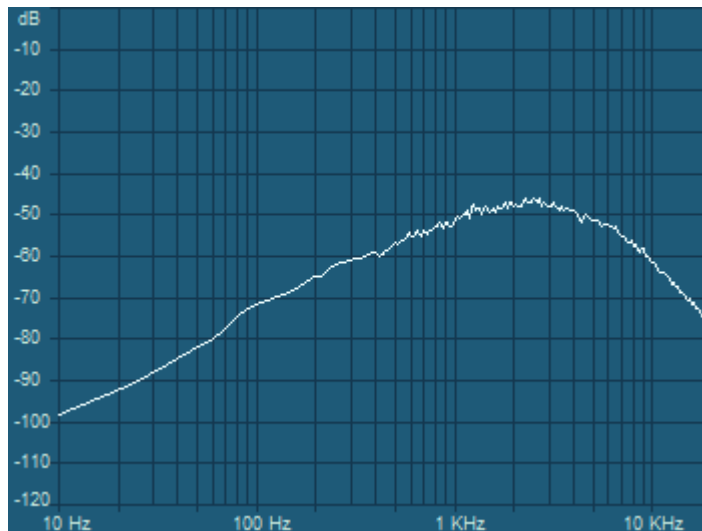
Maggiore è il rumore, maggiori saranno le incertezze nella misura della ampiezza degli impulsi e maggiore sarà lo FWHM .



Se si seguono le nostre indicazioni il rumore non deve superare i 500uV p.p, meglio se si arriva a 300 uV p.p. come in questa immagine.

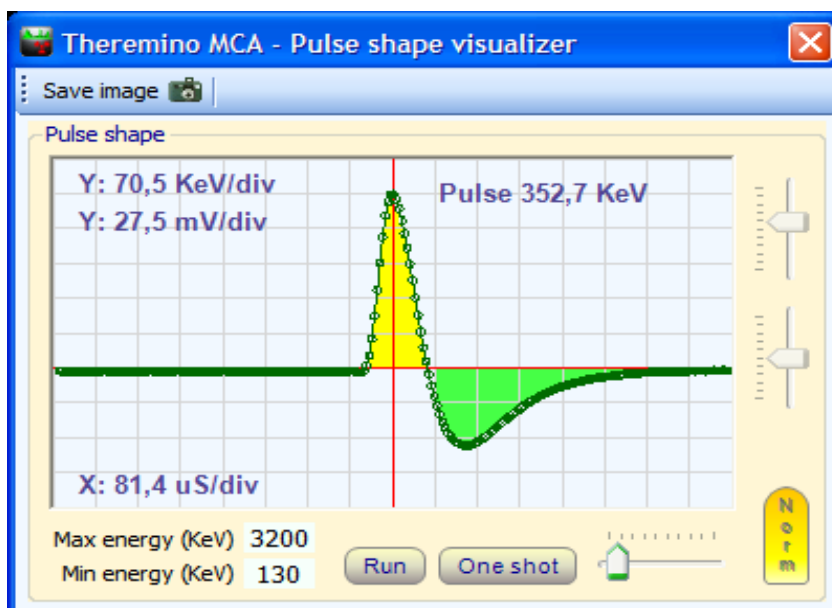
Le misure di tensione sono riferite ad una taratura con ADC che satura a +/- 1 Volt (approssimativamente le condizioni che si ottengono se si regola il mixer e si disabilita l'ADC come indicato nel file PMtAdapters.pdf)

Appendice 7 - Il filtro passa alto



Questo è lo spettro medio degli impulsi dopo il circuito di condizionamento del segnale. Si noti che gran parte della energia è concentrata nella zona da 2 a 3 KHz.

Questa considerazione ci ha permesso di modificare la scheda audio con un filtro passa alto a 3 KHz che produce molti vantaggi tra cui la minimizzazione del rumore a bassa frequenza e la eliminazione dell'onnipresente ronzio a 50 o 60Hz proveniente dalla rete elettrica.



Il filtro passa alto, oltre a eliminare il rumore a bassa frequenza, trasforma l'impulso in "bipolare" (area della zona gialla uguale all'area della zona verde)

Gli impulsi bipolari garantiscono un ritorno a zero veloce e la quasi totale eliminazione dei dondolamenti della linea di zero. Questo facilita il successivo lavoro del "Baseline restorer"

Appendice 8 - I bit dell'ADC

Per il nostro approccio "tutto software" 14 bit sono il minimo indispensabile.

Ma 16 bit permettono di lavorare comodamente, senza la continua preoccupazione della saturazione (se si amplifica troppo) o della perdita di dinamica (se si amplifica poco)

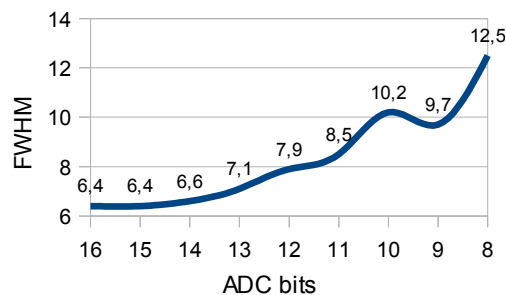
Con 12 bit lo FWHM aumenta di oltre un punto e da 10 bit in giù aumenta moltissimo.

Abbiamo verificato le precedenti affermazioni eliminando progressivamente i bit in software e vedendo l'incremento di FWHM risultante. Per ripetere la nostra prova si deve aggiungere la riga seguente nella funzione "DataArrived" del modulo "WaveRecorder".

```
WaveInBuffer(i) = k * Cint(WaveInBuffer(i) / k)
```

```
// con k=2,4,8,16,32,64,128,256 per degradare l'ADC a 15,14,13,12,11,10,9,8 bit
```

ADC bits	FWHM
16	6,4
15	6,4
14	6,6
13	7,1
12	7,9
11	8,5
10	10,2
9	9,7
8	12,5



Per eliminare ogni possibile effetto casuale questo test è stato fatto sempre nelle stesse condizioni, con lo stesso tempo e usando più volte lo stesso file di dati (il file Cs-137 fornito da Steven Sesselman e reperibile sul nostro sito) Chi ripettesse il test tenga però presente che sono valori "approssimativi" molti fattori (i parametri del "baseline" ad esempio) possono modificarli anche di molti punti. I valori a 14, 15, 16 bit cambiano poco ma i valori con ADC a 11, 10, 9 e 8 bit sono molto dipendenti dai parametri. Si noti anche che a 9 bit lo FWHM è nuovamente sceso, questo indica che è intervenuto qualche diverso meccanismo di errore che potrebbe essere eliminati scartando gli impulsi relativi.

Scartando gran parte degli impulsi (oltre il 70%) il valore a 8 bit è sceso dal 12.5% a circa il 10%. Ma poi anche scartando quasi tutti gli impulsi, e quindi rallentando terribilmente le misure, non si sono ottenuti ulteriori miglioramenti.

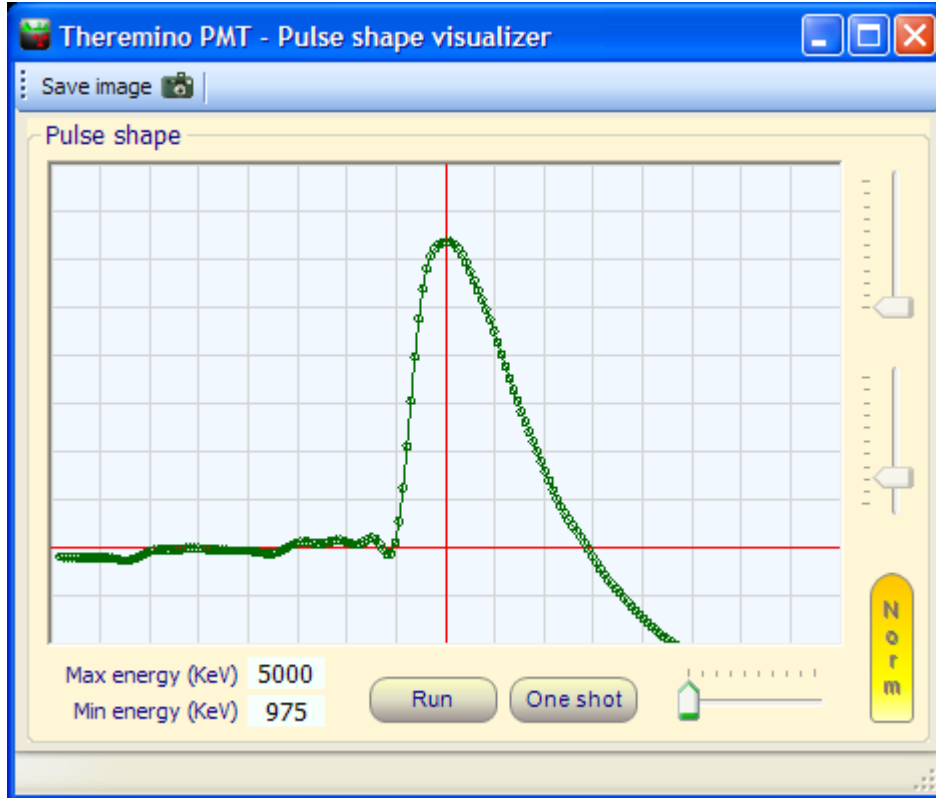
Comunque, anche se impreciso e dipendente da molto fattori, questo test conferma in pieno la nostre prove del 2012, cioè che l'ADC deve avere almeno 14 bit.

Alcuni documenti asseriscono che potrebbero bastare 8 bit (con sliding scale) ma in quelle condizioni poi si dovrebbe fare tutto in hardware e usare un sample-hold hardware. Questi stessi documenti dimenticano di scrivere (o non se ne rendono conto) che così facendo si sono auto-imposti limitazioni praticamente non risolvibili, che impediscono di abbassare lo FWHM.

Componenti analogico-digitali con fronti ripidi, come sliding-scale e sample-hold, sono difficilissimi da far convivere con il delicato segnale analogico e aggiungono molto rumore, probabilmente anche dieci volte il rumore che noi consideriamo già troppo.

Appendice 9 - Alta frequenza di campionamento

La frequenza di campionamento deve essere paragonabile alla larghezza degli impulsi, noi usiamo una frequenza di 192 KHz con impulsi larghi oltre 100 uS. I venti campionamenti per impulso che ne risultano permettono di misurare bene, oltre alla altezza dell'impulso, anche altre importanti caratteristiche, come la sua forma, la quantità di rumore e la pendenza della linea di base.



Si noti in questa immagine il gran numero di campionamenti (pallini verdi) in prossimità della punta.

Se non si allargassero gli impulsi sarebbero larghi una frazione di uS e per ottenere le nostre prestazioni si dovrebbe usare un ADC da oltre 40 MHz. E gli ADC da 40 MHz sono difficili da usare, consumano molto e hanno pochissimi bit, 8 o 10 se va bene.

E se anche si campionasse a 40MHz poi si dovrebbero poter analizzare in continuazione gli oltre 40 Mega Byte di dati al secondo che ne escono fuori. Con normali PC e tanto meno con dispositivi embedded questo non si potrebbe fare.

Come fanno dunque i sistemi hardware? Usano un sample hold hardware e decidono in hardware quando farlo scattare. E così tutto il fine lavoro algoritmico che facciamo per trovare "esattamente" la punta dell'impulso, e misurarlo con la massima precisione possibile, lo si affida a due transistor che scattano con poca intelligenza.

I metodi hardware di campionamenti sono troppo grezzi, funzionano bene solo quando gli impulsi sono distanziati tra loro, grandi e puliti, ma appena c'è rumore scattano un pochino prima o un pochino dopo... e questi pochini che noi non sbagliamo sono appunto quello che ci permette di raschiare qualche frazione di FWHM in più. E tante frazioni di FWHM alla fine fanno quell'intero punto percentuale che siamo riusciti a ottenere.

Appendice 10 - Da qui in poi tutto software

L'elettronica analogica e digitale è affascinante, ma si deve usarla con la maggiore parsimonia possibile, si deve convertire il segnale in digitale subito, meno componenti discreti si saldano e meglio è.

Gli algoritmi fatti in hardware fanno quello che possono, stretti da una parte dalla limitazione di complessità, dall'altra dal rumore che gli gira intorno e senza dimenticare le correnti di perdita, le capacità parassite, il ringing, la rete a 50Hz, gli errori di progettazione, l'umido, i trimmer che grattano e il tempo che li degrada.

Fare gli algoritmi in digitale permette tutt'altra precisione di calcolo e totale assenza di rumore ed è proprio così che noi riusciamo a recuperare tanti piccoli decimi di FWHM.

Il nostro approccio è possibile solo facendo tutto in software, con una CPU in grado di fare moltiplicazioni floating point in un solo ciclo di clock, che possa applicare algoritmi pesanti, buffer circolare e filtri digitali su tutti i campioni, uno per uno.

Da escludere che si possa fare questo con sistemi embedded, DSP, ATMEL, ARM o PIC (anche della serie 32) e nemmeno con i telefonini. Non hanno la potenza necessaria e non hanno ambienti di sviluppo adeguati allo sviluppo di software così complesso.

Abbiamo impiegato anni per ottenere questi risultati pur lavorando con un ambiente di sviluppo ad alto livello e ogni facilitazione possibile. Fare le stesse cose in un DSP, in un ATMEL, un PIC o in un telefonino sarebbe come scrivere un libro al buio e con una coda di struzzo al posto della penna.

Chi pensa sia possibile dovrebbe provare a farlo, il nostro software è tutto OpenSource...

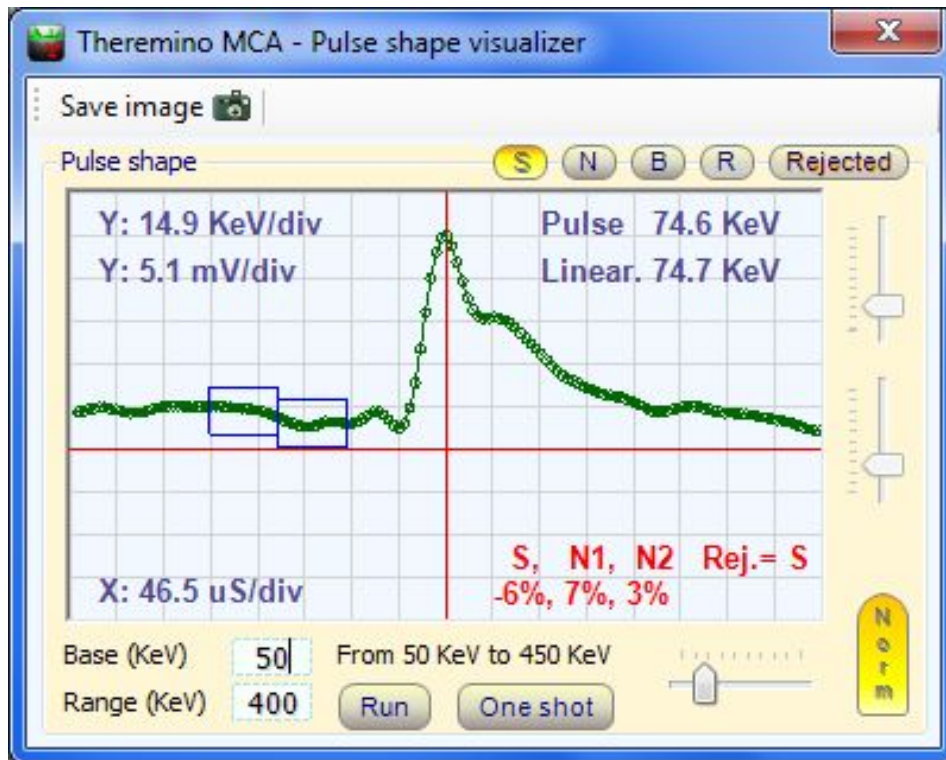
Poi ci sarebbe un piccolo particolare... perché?

- 1) Il discorso sulla "potenza aggiunta dall'hardware" non convince, piuttosto sarebbe già un miracolo riuscire a fare tutto quello che facciamo adesso.
- 2) Hardware veloce, come ad esempio un DSP, produrrebbe tanto di quel rumore digitale che i nostri 200uV, conquistati a fatica, non si potrebbero più ottenere.
- 3) Aggiungere hardware, aumenterebbe costi e difficoltà di costruzione, senza nessun vantaggio in cambio.

Se il "vantaggio" che si desidera ottenere è un oggetto portatile allora consigliamo un NetBook o un Tablet o un eeeBox che oltretutto forniscono alimentazione, tastiera e schermo decentemente grande e anche batterie, nonché caricabatterie a norme CE. Senza dover fare tutto a mano, per poi ritrovarsi con una scatola da scarpe piena di fili e, tutto sommato, anche meno portatile di un NetBook.

Chi dovesse avere dei dubbi sulle nostre affermazioni consideri che nei passati trenta anni è stato provato di tutto e nessuno è sceso molto sotto il 7%, invece con il nostro approccio si arriva facilmente al 6% e anche verso il 5%.

Appendice 11 - Misurazione precisa dell'impulso



Si provi a misurare l'altezza di questo impulso sono 4.5 quadretti o 5 quadretti?

Difficile dirlo vero?

E bastano errori di **un ventesimo di quadretto** per aggiungere un **1% di FWHM**

Maggiore è la precisione con cui si misura l'altezza degli impulsi e minore è il valore di FWHM risultante.

A volte la presenza di rumore e l'affollamento di molti impulsi vicini possono rendere difficile misurare l'altezza.

Anche la presenza di una linea di zero dondolante o la presenza di ringing nella zona che precede l'impulso possono peggiorare la precisione di misura.

E anche una frequenza di campionamento bassa (pallini verdi poco fitti) aumenta l'incertezza delle misure.

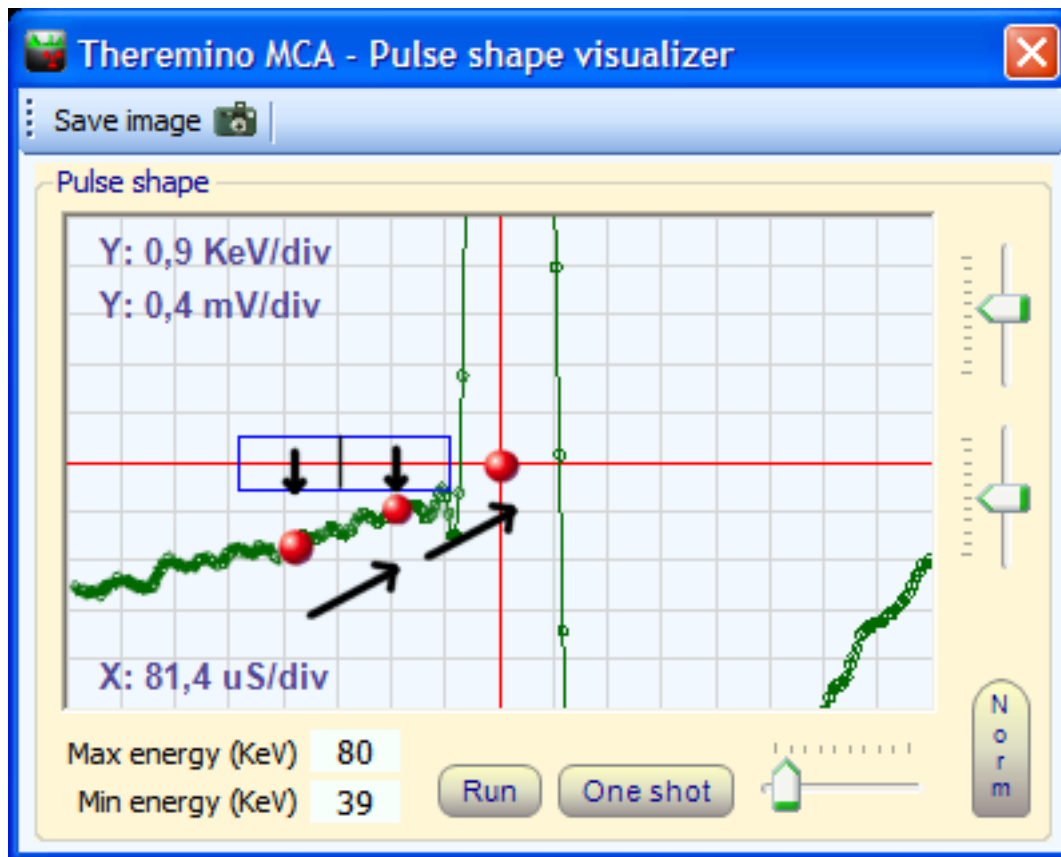
E lo stesso succede se i bit dell'ADC sono pochi.

Se l'errore di misura è piccolo l'impulso finisce in un bin appena accanto alla riga e aumenta il valore di FWHM. Se invece l'errore è grande (come nei casi in cui si sbaglia a individuare il centro dell'impulso) l'impulso finisce in un bin completamente sbagliato e aumenta il tappeto di rumore.

Nei sistemi che lavorano senza un visualizzatore di impulsi e di errori il numero di impulsi misurati in modo impreciso è molto più grande di quello che normalmente si pensa. Anche noi all'inizio non ce ne rendevamo conto ma, dopo avere stanato una per una tutti le cause di errore, è diventato evidente che nei sistemi "normali" il numero di impulsi sbagliati può essere anche oltre il 10%. E questa è una delle cause dell'alto tappeto di rumore che tutti considerano normale.

Appendice 12 - Il Baseline Restoring

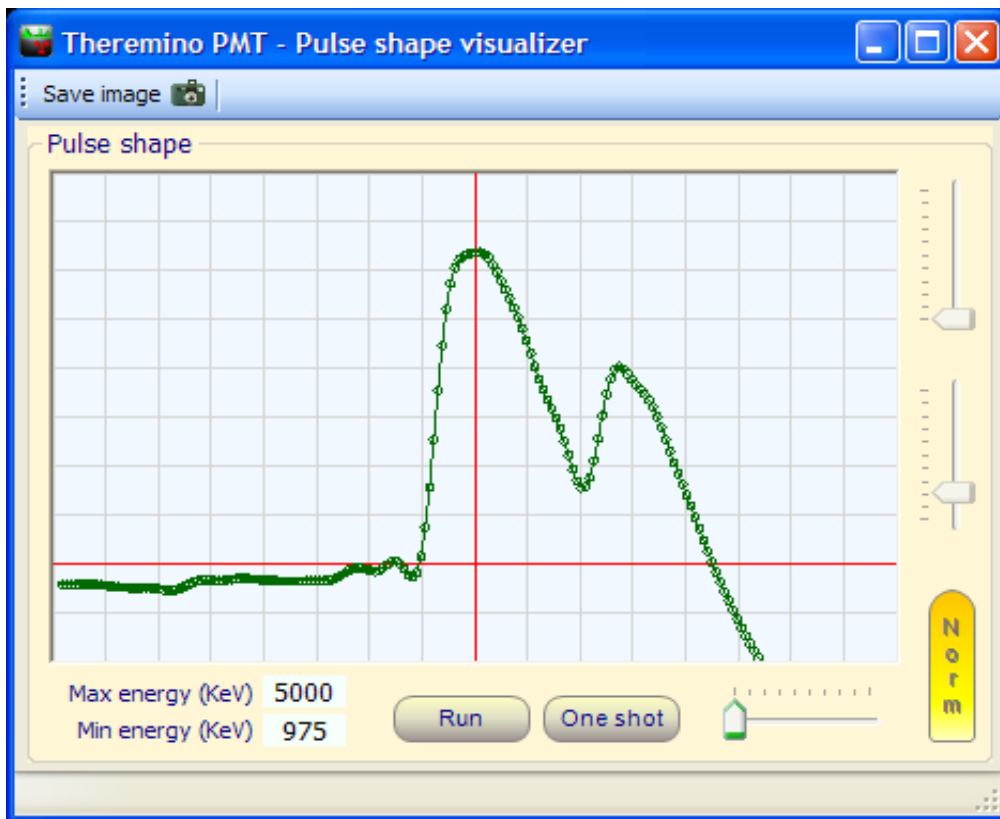
Compensazione dell'offset della linea di zero con BaseLine Restoring.



Il ThereminoMCA individua la **posizione dello zero nel centro virtuale dell'impulso** (terzo pallino rosso) con la massima precisione possibile.

Vengono fatte due medie per individuare la miglior stima della posizione dei primi due pallini rossi e infine una estrapolazione pesata per la posizione del terzo pallino. Questo in hardware non si potrebbe fare e se anche lo si facesse non si avrebbe la precisione necessaria.

Appendice 13 - Individuazione dei Pile Up



Questi sono due impulsi impilati, il secondo si appoggia sulla discesa del primo.

Si può misurare l'altezza del primo impulso con precisione, ma il secondo impulso viene sollevato dalla discesa del primo e non è possibile individuare la posizione della sua linea di base.

Quindi il primo impulso è valido ma non potendo misurare il secondo lo si deve scartare.

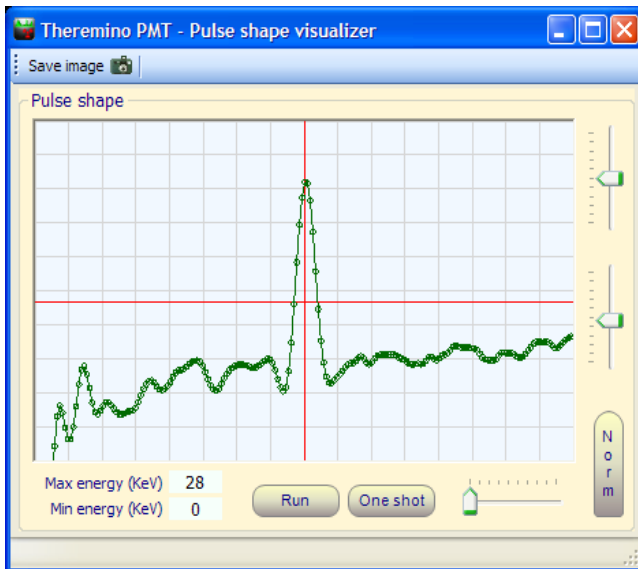
Tutti i PileUp devono essere individuati in modo affidabile ma quando la situazione si fa intricata, per la presenza di rumore o per l'affollamento di molti impulsi vicini, può essere difficile scoprirli.

Con la limitata intelligenza possibile in hardware il numero di PileUp non individuati può essere significativo e ogni PileUp "scappato" genera un impulso misurato male, e ogni impulso misurato male allarga le righe (se l'errore è piccolo) oppure aumenta il tappeto di rumore (se l'errore è grande)

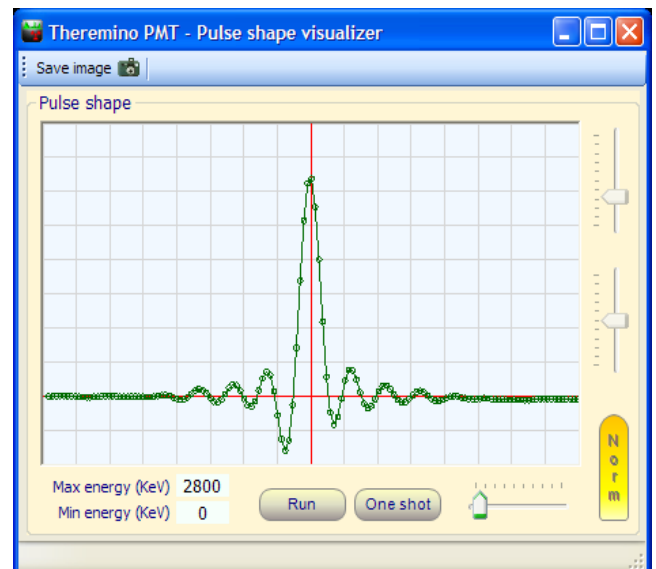
Appendice 14 - Impulsi troppo disturbati

Individuazione degli impulsi troppo "disturbati" (da rumore, ringing, pileup o linea di zero instabile)

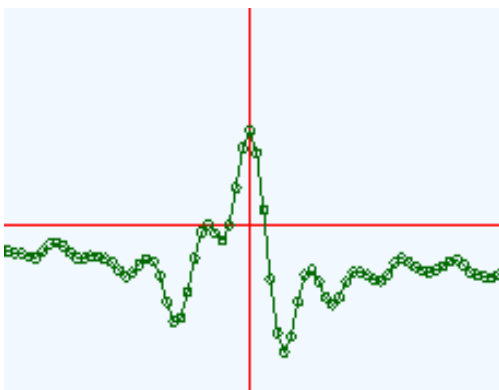
Per avere il minimo FWHM tutti gli impulsi della stessa altezza devono finire nel medesimo bin ma ogni tipo di errore può farli finire in un bin diverso, dalla imprecisa misura della ampiezza alla imprecisa valutazione dello zero.



Impulsi con la linea di zero instabile



Impulso affetto da forte "ringing"

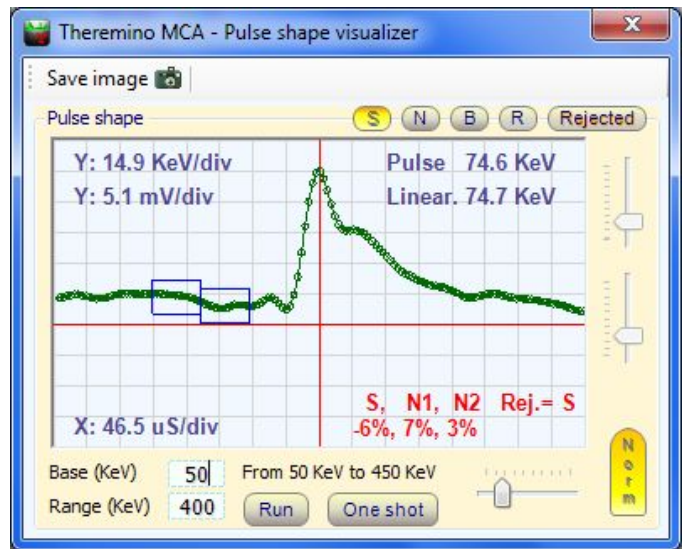
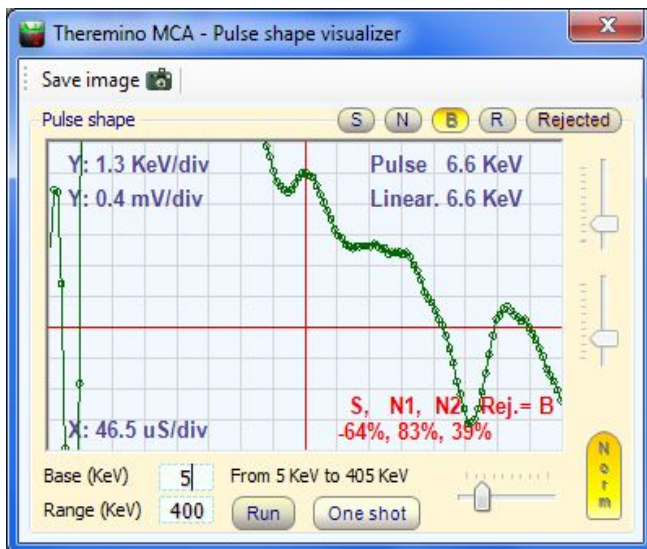
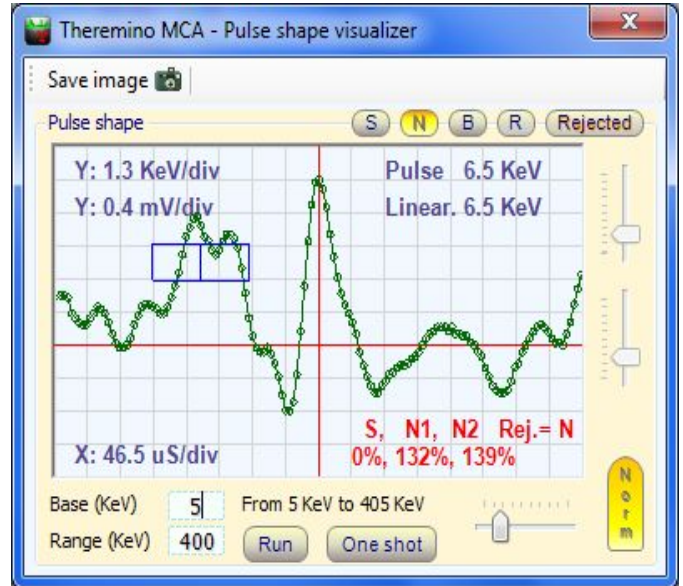
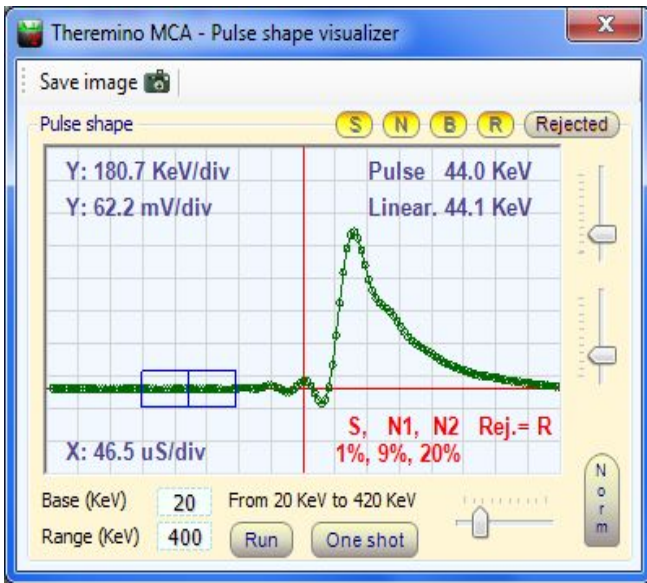


Impulso molto disturbato

Alcuni impulsi, per svariati motivi, possono essere disturbati al punto da rendere impossibile misurare la loro altezza con precisione. Questi impulsi devono essere tutti riconosciuti e scartati.

Ogni impulso misurato male allarga le righe (se l'errore è piccolo) oppure aumenta il tappeto di rumore (se l'errore è grande)

Appendice 15 - Funzioni di controllo



Una strumentazione "di servizio" in grado di individuare gli errori e di mostrarne la frequenza e il tipo è essenziale per ridurre lo FWHM.

Anche conoscere la forma degli impulsi e visualizzare i casi "difficili" aiuta molto a effettuare un buon "tuning" dei parametri e a individuare eventuali difetti dell'hardware.

Ci sono una gran quantità di eventi strani che si verificano in pratica e che la teoria dimentica di considerare.

Le applicazioni che non dispongono di questi strumenti fanno tutto al buio e credono di lavorare bene ma non è così e i loro spettri lo dimostrano.

Appendice 16 - Compensazione della risoluzione

Il nostro algoritmo di compensazione è una forma particolare di Deconvoluzione digitale.

DEFINIZIONE: La deconvoluzione è un processo basato su algoritmi utilizzati per invertire gli effetti di convoluzione sui dati registrati. Il concetto di deconvoluzione è ampiamente utilizzato nelle tecniche di elaborazione del segnale e di elaborazione delle immagini. Queste tecniche sono a loro volta ampiamente utilizzate in molte discipline scientifiche e ingegneristiche, nelle quali la deconvoluzione trova molte applicazioni. Tratto da: en.wikipedia.org/wiki/Deconvolution

L'algoritmo di ThereminoMCA

Secondo la letteratura corrente il nostro è un "Algoritmo di deconvoluzione tramite derivazione"

Prima di scrivere questo documento non sapevamo nulla sulla deconvoluzione e ora, dopo aver letto la letteratura sull'argomento, il nostro algoritmo appare semplice e ingenuo. Comunque i suoi risultati sono confrontabili con algoritmi più eleganti e matematicamente corretti, come quello del documento "Deconvolution of the Raman Spectrum of Amorphous Carbon" visibile nella prossima pagina.

Il nostro algoritmo effettua due scansioni dello spettro, la prima dal basso in alto e la seconda dall'alto in basso (questo per poter correggere in modo indipendente il lato destro e sinistro delle righe) Durante le passate il valore di ogni "BIN" viene modificato sommandogli la differenza (derivazione) tra il suo valore e il valore di un BIN che lo precede con distanza di un certo numero di BIN (regolabile con il parametro "Size (bins)") e moltiplicato per una frazione del parametro "Center (%)" I due ulteriori parametri "Left(%)" e "Right(%)" servono per regolare indipendentemente la quantità di correzione sui lati destro e sinistro delle righe. Altri due parametri interni (prefissati e non a disposizione dell'utente) regolano la quantità di correzione per le pendenze negative rispetto a quelle positive.

Ringraziamo molto l'utente "bepobalote" del radioactivityforum che non si è fidato di spiegazioni approssimative e ha fortemente richiesto questo documento. Grazie a lui abbiamo trovato le versioni "serie" del nostro algoritmo (principalmente quella ricorsiva) e **ci sono buone possibilità che applicandole si possa "risicare" un'altra frazione di FWHM.**

La deconvoluzione ricorsiva

Una delle tecniche di deconvoluzione più note (chiamata "ricorsiva") implica diverse fasi: calcolo della trasformata di Fourier inversa dello spettro, moltiplicazione per una funzione Gaussiana o Lorentziana e infine trasformata di Fourier del risultato. Il processo viene ripetuto iterativamente nel cosiddetto tuning della forma di banda.

I parametri variabili di questo algoritmo sono:

- La proporzione della forma di banda: Gaussiana o Lorentziana;
- La larghezza della banda. Se troppo piccola, lo spettro mostra piccole variazioni di intensità; se troppo grande, si producono bande negative lateralmente a quelle deconvolute;
- Il grado di intervento. Se troppo piccolo, lo spettro risultante non differisce molto dall'originale; se troppo grande, si producono false bande e anche il rumore tende ad aumentare.

Un esempio di deconvoluzione ricorsiva

Un perfetto esempio di deconvoluzione ricorsiva si trova nel documento: "Deconvolution of the Raman Spectrum of Amorphous Carbon": www.physics.byu.edu/faculty/allred/cv/publications%5CDeconvolution%20of%20the%20Raman%20Spectrum%20of%20Amorphous%20Carbon.pdf

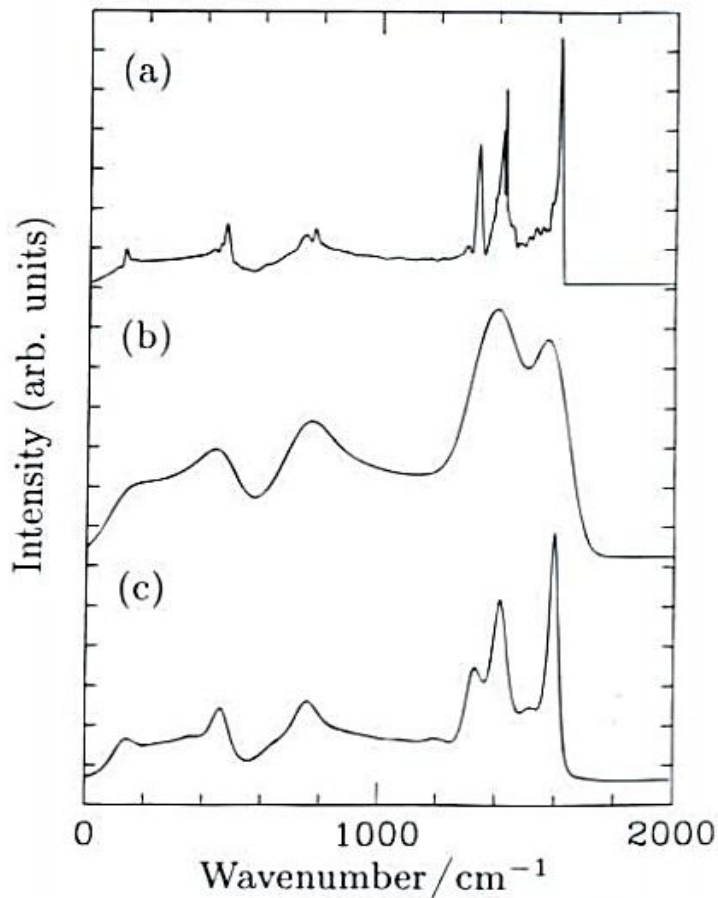


Figure 1. (a) The known spectrum taken as σ . (b) The convolution of the spectrum in (a) with a Gaussian function with a width factor of 50 cm^{-1} . (c) The deconvolution of (b) using a Gaussian-shaped spread function with a width factor of 50 cm^{-1} . The iteration number is 200.

La Spettrometria Raman produce spettri con risoluzione molto scarsa (grafico b) che assomigliano in modo sorprendente a quelli della Spettrometria Gamma con cristalli Nai(Tl)

Il grafico (a) mostra lo spettro che si otterrebbe con un rivelatore perfetto.

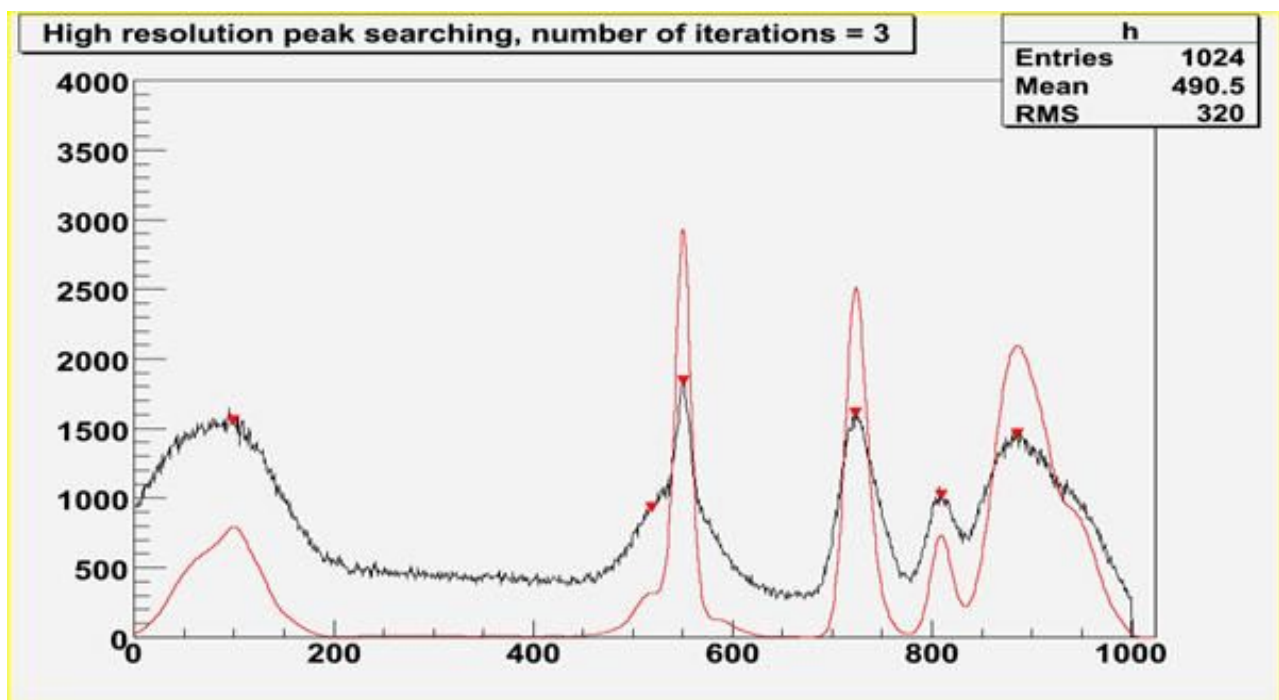
Il grafico (c) mostra l'incremento di risoluzione che si ottiene con la deconvoluzione.

Sono praticamente gli stessi risultati che otteniamo noi, talmente simili ai nostri spettri che potremmo confonderli facilmente. E anche la parziale ricostruzione dei dati originali e il miglioramento di FWHM appaiono molto simili a quelli ottenibili con il nostro algoritmo.

Un esempio su dati della Spettrometria Gamma

Sembrava impossibile che proprio nessuno ci avesse pensato e alla fine, cerca cerca, abbiamo trovato anche qualcuno che applica la deconvoluzione alla Spettrometria Gamma.

E chi potevano essere se non il CERN di Ginevra, i maghi della raccolta dati? E chi ha fatto il "lavoro sporco"? Uno slovacco di nome Miroslav Morhac: **M.Morhac et al.: Efficient one- and two-dimensional Gold deconvolution and its application to gamma-ray spectra decomposition. Nuclear Instruments and Methods in Physics Research A 401 (1997) 385-408.**



Per maggiori informazioni andare qui: <http://root.cern.ch/root/html/doc/TSpectrum.html>

E' tutto open source e ci sono gli algoritmi pronti, basta copiarli. Ma non è detto che alla fine trasferiremo queste classi nel nostro software, per vari motivi:

- Sono algoritmi pesanti, lenti e ricorsivi, intesi per essere applicati una volta sola e non dieci volte al secondo come facciamo noi.
- La regolazione dei parametri appare macchinosa e meno intuitiva della nostra.
- Non sembra che i miglioramenti ottenibili possano giustificare la perdita di interattività che provocherebbero con la loro lentezza.

Comunque li proveremo.

Limiti degli algoritmi di deconvoluzione

Se i dati fossero deteriorati con una funzione, che potremmo chiamare $F(x)$, della quale potessimo conoscere, sia la completa espressione matematica che i suoi coefficienti, basterebbe applicare la funzione inversa per ottenere esattamente lo spettro originale.

Il problema di conoscere i coefficienti è risolvibile trovando sperimentalmente i valori che danno i migliori risultati e la funzione è generalmente una gaussiana con leggere deformazioni causate da alcuni meccanismi minori.

Purtroppo però esistono motivi che impediscono una conoscenza completa sia della funzione che dei suoi parametri.

Quando un raggio gamma si disintegra nel cristallo genera un gran numero di fotoni. A seconda del punto del cristallo in cui avviene questa conversione il numero di fotoni generato è leggermente diverso (a causa della non perfetta omogeneità del materiale) E sempre a seconda della posizione il numero di riflessioni che i fotoni devono compiere contro le pareti prima di raggiungere il fotomoltiplicatore cambia. Per cui si genera una incertezza nella ampiezza dell'impulso generato.

Si dovrebbe dunque applicare una funzione diversa per ogni impulso, con parametri leggermente diversi e si dovrebbe fare questo sulla base di conoscenze che non possiamo avere.

Fortunatamente la somma di tutti gli errori si traduce in una funzione grossolanamente gaussiana con la sovrapposizione di alcune variazioni casuali. L'andamento gaussiano può essere corretto ma le variazioni casuali no. Pertanto la ricostruzione sarà solo parziale.

Se si esagera con la quantità di correzione si generano artefatti. E aumentando ulteriormente la correzione gli artefatti aumentano fino a che la curva risultante diventa peggiore di quella che si intendeva correggere.

Obiezioni

Qualcuno pensa che le tecniche di deconvoluzione non possano aggiungere nulla ai dati grezzi e dice "ciò che si è sottratto è ormai perso e non si può ricreare". Ma dato che si conoscono alcune caratteristiche della funzione che ha degradato i dati, basta "ricreare" la parti mancanti "aggiungendo" le conoscenze che abbiamo per ottenere un effettivo e reale miglioramento della qualità dei dati.

Conclusioni

Esiste la possibilità di migliorare, almeno in parte, i dati dello spettro.

La correzione non è solo cosmetica o apparente ma aumenta realmente la risoluzione della sonda.

Esiste un determinato set di parametri che garantisce la massima qualità della correzione e questi parametri possono essere trovati sperimentalmente.