
UNIVERSITÀ DEGLI STUDI DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E NATURALI
CORSO DI LAUREA IN FISICA

**Esperimento AMS-02: il nuovo sistema di
acquisizione dati del telescopio per il test dei
contatori a scintillazione.**

Tesi di Laurea
di:
MICHELE SALVADORE

Relatore:
Chiar.mo Prof. FEDERICO
PALMONARI

Correlatori:
Dr. GIUSEPPE LEVI
Dr. FEDERICO CINDOLO

Anno Accademico 2001/2002— Sessione II

Indice

Introduzione	vii
1 Fisica dei raggi cosmici	1
1.1 Introduzione	1
1.2 Antimateria	2
1.2.1 Il modello standard delle particelle	2
1.2.2 Simmetrie	2
1.2.3 Simmetria materia-antimateria	3
1.2.4 Universo a bolle	4
1.2.5 Bariogenesi	4
1.3 Materia Oscura	8
1.3.1 Il modello cosmologico standard	8
1.3.2 Misure dei parametri cosmologici	10
1.3.3 Candidati a dark matter	15
1.4 Raggi cosmici	16
1.4.1 Rigidità magnetica	19
1.4.2 Cutoff geomagnetico	20
1.4.3 Propagazione ed origine dei raggi cosmici	20
1.4.4 Accelerazione dei raggi cosmici	23
1.5 Risultati dell'esperimento AMS-01	24
1.5.1 Misura \overline{He}/He	25
1.5.2 Spettro dei protoni	25
1.5.3 Spettro dei leptoni	26
1.6 Fisica per AMS-02	27
2 Lo spettrometro AMS-02	29
2.1 Funzionamento di uno spettrometro	29
2.1.1 Perdita di energia	29
2.1.2 Misura del momento	31

INDICE

2.1.3	Misura della velocità	32
2.1.4	Identificazione delle particelle	34
2.2	Il rivelatore nella prima fase	34
2.3	Struttura di AMS-02	35
2.3.1	Il Magnete Superconduttore	36
2.3.2	Il Tracker	37
2.3.3	Il Time Of Flight	38
2.3.4	Il Ring Imaging Cherenkov	39
2.3.5	Il Transition Radiation Detector	40
2.3.6	Il Calorimetro Elettromagnetico	40
2.3.7	Le Anticoincidenze	42
2.4	Il TOF di AMS-02	42
2.4.1	I contatori del TOF	43
2.4.2	Trigger	45
3	L'hardware del telescopio per raggi cosmici	47
3.1	Il telescopio per il test dei contatori coi raggi cosmici	48
3.1.1	I tubi LST	48
3.1.2	Tracciamento delle particelle: le strips	49
3.1.3	Il punto di lavoro dei tubi	50
3.2	L'elettronica per l'acquisizione dati	51
3.3	Fase 1: CAMAC	53
3.3.1	Ricostruzione dell'evento	58
3.3.2	Lunghezza di attenuazione	64
3.3.3	Fotostatistica	64
3.3.4	Correzione di Time Walk	66
3.3.5	Velocità di propagazione della luce nello scintillatore	67
3.3.6	Risoluzione sul tempo di volo	68
3.3.7	Risoluzione intrinseca	72
3.4	Fase 2: VME	75
3.4.1	Confronto con la fase 1	78
4	Il software di acquisizione dati	81
4.1	Struttura generale	81
4.1.1	Utilizzo delle classi	83
4.1.2	Moduli e interfacce: classi di lettura	83
4.1.3	I processi di acquisizione e monitoring	91
4.2	GUI e slow control	95
4.3	Analisi dell'efficienza	96
4.4	Utilizzo del software	98

5	Il DAQ di AMS	103
5.1	Sistema di acquisizione dati di AMS-02	103
5.1.1	Resistenza alla radiazione	103
5.1.2	Rindondanza	105
5.1.3	Schema generale	105
5.2	Definizione del sistema di acquisizione del TOF	108
5.2.1	Scheda di sviluppo S9007	109
5.2.2	SFET	110
5.2.3	SFEC	111
5.2.4	SDR	111
5.2.5	USCM	112
5.3	Integrazione SDR-SFE[X]	113
5.3.1	Prospettive future	115
	 Conclusioni	 117
A	Alcuni estratti del codice del DAS	121
B	Studio dell'efficienza del DAS	129
C	Il codice della SDR	131
	 Bibliografia	 135
	 Ringraziamenti	 137

Introduzione

Questo scritto riguarda il lavoro da me svolto presso l'Istituto Nazionale di Fisica Nucleare (sezione di Bologna) a partire dall'ottobre 2001 nell'ambito della costruzione del TOF (*Time Of Flight*) dello spettrometro per raggi cosmici AMS (*Alpha Magnetic Spectrometer*).

Il TOF, strumento dedicato alla misura del tempo di volo, è costituito da una serie di contatori a scintillatore plastico le cui caratteristiche devono essere accuratamente studiate. A questo scopo è stato installato a Bologna un apparato che consente il tracciamento delle particelle prodotte nelle interazioni tra raggi cosmici e atmosfera (principalmente muoni al minimo di ionizzazione) che arrivano al suolo. L'analisi della risposta dei contatori al passaggio di queste particelle permette di ottenere in breve tempo i dati necessari alla valutazione delle loro caratteristiche.

Il compito assegnatomi è stato la realizzazione del sistema di acquisizione dati di questo apparato (telescopio per raggi cosmici), ed in particolare la gestione del passaggio dall'elettronica usata nella fase di costruzione di AMS-01 (un prototipo dello spettrometro) all'elettronica di nuovo tipo usata attualmente.

Il lavoro si compone sostanzialmente di due parti eseguite parallelamente: l'assemblaggio dell'*hardware* e la scrittura del *software*.

La prima parte ha visto una fase iniziale in cui è stato realizzato un sistema di acquisizione basato su elettronica *standard* che ha permesso di ottenere dati relativi al primo prototipo del contatore. In una seconda fase l'*hardware* è stato riassembleato con elettronica moderna interfacciata con un *bus* di tipo VME, ed i dati acquisiti sullo stesso contatore usato in precedenza hanno reso possibile il confronto tra le due configurazioni diverse.

Durante l'estate 2002 ho avuto l'occasione di passare tre mesi al CERN di Ginevra come *Summer Student*; nel corso di questo periodo ho lavorato nell'ambito dell'esperimento HARP, proseguendo nel tempo libero il lavoro in AMS collaborando con alcuni ricercatori che seguono lo sviluppo del sistema di acquisizione dell'apparato. Questa tesi raccoglie in cinque capitoli una descrizione del lavoro che ho svolto e delle nozioni che ho appreso nel corso di questo anno di collaborazione con AMS.

Nel **primo capitolo** viene introdotta la fisica da cui l'esperimento trae ispirazione; in particolare si introduce il problema dell'antimateria e vengono accennate alcune prove sperimentali relative alla presenza di materia oscura. Nella parte finale si dà un'introduzione alla fisica dei raggi cosmici (caratteristiche, propagazione e accelerazione), e si riportano i principali risultati ottenuti da AMS-01.

Nel **secondo capitolo** si descrive in dettaglio la versione definitiva dello strumento (AMS-02), analizzando i singoli *subdetectors* e focalizzando l'attenzione sul TOF nella parte finale.

Il **terzo capitolo** dà una descrizione completa del lavoro relativo alla costruzione del sistema di acquisizione del telescopio per raggi cosmici e descrive l'analisi dei dati relativi al contatore prototipo.

Il **quarto capitolo** tratta lo sviluppo del *software* di acquisizione dati (DAS), che ha trovato come prime applicazioni la gestione del telescopio per raggi cosmici e del sistema di acquisizione dati del *test beam* svolto al CERN (fascio SPS) per la caratterizzazione in ampiezza d'impulso di due prototipi dei contatori del TOF.

Nel **quinto capitolo** viene introdotto il sistema di acquisizione di AMS e vengono riportati i primi sviluppi relativi all'elettronica di acquisizione dati del TOF; in particolare si accenna al previsto utilizzo del *software* descritto nel capitolo quattro con i sistemi di *test* e di sviluppo dell'elettronica di AMS.

Capitolo 1

Fisica dei raggi cosmici

1.1 Introduzione

AMS, acronimo di *Alpha Magnetic Spectrometer*, è uno spettrometro magnetico progettato per operare sulla stazione spaziale internazionale *Alpha* (ISS). La fase centrale della missione, che inizierà nel 2005 e si protrarrà per tre anni, vedrà AMS orbitare ad una quota di circa 400 km ancorato alla struttura portante della ISS.

La costruzione di AMS vede coinvolti gli istituti di fisica delle alte energie e le agenzie spaziali di molti paesi, tra cui Cina, Finlandia, Francia, Germania, Italia, Portogallo, Regno Unito, Romania, Russia, Spagna, Stati Uniti, Svizzera e Taiwan.

I motivi che hanno stimolato la realizzazione di un apparato di questo tipo sono legati sia alla fisica delle particelle che all'astrofisica che alla cosmologia; in particolare a due aspetti osservativi che non hanno ancora trovato una chiara spiegazione teorica:

- l'apparente asimmetria tra la quantità di materia e quella di antimateria presente nell'universo;
- l'evidenza dell'abbondante presenza di una o più forme di materia non ancora identificate.

L'obiettivo principale di AMS è l'acquisizione di una notevole mole di dati relativi ai raggi cosmici che arrivano nella parte superiore dell'atmosfera terrestre, al fine di porre limiti precisi sulla presenza di antimateria di origine cosmologica e di verificare le previsioni dei modelli che candidano particelle supersimmetriche al ruolo di materia oscura.

Una ricaduta di primaria importanza dell'esperimento sarà la misura accurata delle abbondanze assolute e degli spettri energetici dei raggi cosmici in prossimità delle terra.

1.2 Antimateria

1.2.1 Il modello standard delle particelle

Il *Modello Standard delle particelle* è la struttura matematica che attualmente viene usata per la descrizione formale della fisica di tre delle quattro interazioni fondamentali conosciute: interazione *elettromagnetica*, *forte* e *debole*.

Di questo modello fanno parte una serie di campi di “materia” (spinoriali), alcuni campi bosonici (vettoriali) che consentono la mediazione delle forze fondamentali ed un campo scalare tramite il quale viene assegnata la massa ai campi che ne sono dotati. I primi descrivono particelle a spin 1/2 suddivise in leptoni e barioni raggruppati in 3 famiglie, mentre secondi rappresentano particelle a spin 1.

In questo modello le interazioni elettromagnetica e debole sono in realtà due manifestazioni diverse della stessa forza, originata dall'imporre l'invarianza locale della simmetria $SU(2) \times U(1)$ di cui gode la Lagrangiana libera del sistema (simmetria di *gauge*).

Seguendo la strada indicata dall'unificazione elettrodebole (modello di Weinberg-Salam) si vorrebbe giungere ad una descrizione comune di tutte le interazioni note (compresa quella gravitazionale) in modelli usualmente chiamati GUT (*Grand Unification Theory*).

Un ruolo fondamentale in questo processo di unificazione sembra sia giocato dalle simmetrie del modello, dalle quali derivano sia le interazioni tra i campi (attraverso il principio di *gauge*) che una serie di quantità conservate (teorema di Nöther).

1.2.2 Simmetrie

Si dice che una trasformazione \hat{A} è una simmetria quando l'Hamiltoniana è invariante sotto di essa

$$\hat{H}' = \hat{A}^+ \hat{H} \hat{A} = \hat{H} \quad (1.1)$$

e in tal caso, se la trasformazione è unitaria ($\hat{H}^+ = \hat{H}^{-1}$), essa commuta con \hat{H} ($[\hat{H}, \hat{A}] = 0$). Esistono due tipi di trasformazioni distinte: quelle *continue* possono essere viste come una successione di trasformazioni infinitesime (come ad esempio traslazioni e rotazioni), mentre quelle *discrete* si esauriscono in un singolo passaggio. Queste ultime sono 3:

- C (*Charge conjugation*), cambia segno alla carica elettrica di una particella: $C(a^+) = a^-$;
- P (*Parity*), inverte le coordinate spaziali: $P(a(\vec{x}, t)) = a(-\vec{x}, t)$;
- T (*Time reversal*), inverte la coordinata temporale: $T(a(\vec{x}, t)) = a(\vec{x}, -t)$.

Il *teorema CPT* afferma che l'applicazione combinata di C , P e T è una simmetria esatta.

Nel modello standard le interazioni elettromagnetica e forte conservano separatamente tutte e tre le trasformazioni discrete, le loro combinazioni (CP , CT e PT) ed il sapore debole. L'interazione debole viceversa non conserva quest'ultimo, essendo gli autostati di massa diversi dagli autostati di sapore debole. Il fenomeno è descritto dalla matrice Cabibbo-Kobayashi-Maskawa (CKM)[1][2] che formalizza il mixing che avviene tra gli autostati di massa per ottenere gli autostati di sapore. Per i quark *down-type* d_i si ha

$$d_i^s = V_{ij} d_j^m \quad (1.2)$$

In modelli con una o due famiglie V_{ij} è reale visto che è possibile riassorbire una fase complessa nella ridefinizione del campo u_i . Se le invece le famiglie sono almeno 3 non è più possibile cancellare tutte le fasi complesse della matrice ridefinendo u_i , di conseguenza si avrà in generale $V_{ij} \neq V_{ij}^*$.

Essendo

$$\mathcal{L}_{CKM} = V_{ij} \bar{u}_i \gamma^\mu (1 - \gamma_5) d_j W_\mu^+ + V_{ij}^* \bar{d}_i \gamma^\mu (1 - \gamma_5) u_j W_\mu \quad (1.3)$$

applicando in sequenza gli operatori P e C si ha

$$CP(\mathcal{L}_{CKM}) = V_{ij} \bar{d}_i \gamma^\mu (1 - \gamma_5) u_j W_\mu + V_{ij}^* \bar{u}_i \gamma^\mu (1 - \gamma_5) d_j W_\mu^+ \quad (1.4)$$

Quindi se $V_{ij} \neq V_{ij}^*$ si ha CP violation:

$$\mathcal{L}_{CKM} \neq CP(\mathcal{L}_{CKM}) \quad (1.5)$$

A causa dell'interazione debole quindi, oltre al sapore non viene conservata nemmeno la simmetria CP , e di conseguenza (vista la validità del teorema CPT) nemmeno T .

La prima osservazione sperimentale di CP violation, avvenuta nel '64 studiando i decadimenti nei sistemi di kaoni neutri[3], trova attualmente riscontro nelle recenti misure di asimmetria nei sistemi $B\bar{B}$ [4]. Come si vedrà la violazione di CP gioca un ruolo fondamentale nella discriminazione dei modelli che descrivono la composizione barionica dell'universo.

1.2.3 Simmetria materia-antimateria

Nel '29 Dirac postulò l'equazione relativistica che governa la dinamica di un elettrone libero; essa prevedeva l'esistenza di stati ad energia negativa, interpretati come stati ad energia positiva di una nuova particella dotata delle stesse caratteristiche dell'elettrone ma con carica elettrica opposta. Tre anni dopo, la scoperta dell'antielettrone nei raggi cosmici ad opera di Blacket e Occhialini confermò l'interpretazione

data da Dirac.

Nel modello standard ogni particella esiste nei due stati di materia ed antimateria, ai quali sono associati rispettivamente numero barionico 1 e -1 per i barioni, e numeri leptonici 1 e -1 per le tre famiglie leptoniche.

L'apparente simmetria tra i due diversi stati è all'origine di quello che potrebbe sembrare un paradosso; infatti nella parte di universo a noi accessibile si osserva un totale esubero di materia rispetto all'antimateria, quando non si hanno a disposizione ragionevoli argomenti per assumere una distribuzione iniziale così asimmetrica. È più ragionevole pensare che il numero barionico totale all'origine dell'universo (*Big-Bang*) fosse zero, e che solo in seguito un qualche processo non ancora identificato consentì la sua evoluzione fino allo stato attuale.

Gli scenari che si prospettano in questo caso sono sostanzialmente due: il primo presenta la conservazione rigorosa del numero barionico totale dell'universo, il quale sarebbe formato da enormi bolle di materia o antimateria separate tra loro da ampi spazi vuoti. Il secondo scenario prevede che durante l'evoluzione dell'universo sia stato prodotto un esubero di materia rispetto all'antimateria (bariogenesi).

1.2.4 Universo a bolle

Nell'ipotesi che l'universo sia formato da bolle separate di materia e antimateria, è possibile fissare il limite inferiore del raggio della bolla di materia in cui si trova la terra osservando la quantità di antimateria che giunge sino a noi attraverso la propagazione nel mezzo interstellare.

Da misure effettuate sui raggi cosmici da esperimenti su pallone, appare chiaro che la via lattea è globalmente formata di materia, a parte un piccola quantità di antiparticelle prodotte negli urti energetici tra raggi cosmici, come ad esempio gli antiprotoni prodotti nei processi $p + p \rightarrow 3p + \bar{p}$.

Più precisamente, è possibile porre un limite di circa $20Mpc$ ($\simeq 10^8$ anni-luce) al raggio della bolla in cui si troverebbe la terra; questo limite corrisponde all'incirca alle dimensioni dell'ammasso locale di galassie.

Anche se notevoli sforzi sono stati spesi alla ricerca di modelli cosmologici che fossero in grado di spiegare strutture su larga scala compatibili con l'ipotesi di universo a bolle, attualmente non sono ancora stati ottenuti buoni risultati.

1.2.5 Bariogenesi

Un altro tipo di scenario nasce dall'osservazione che il numero di barioni di cui è composto l'universo è molto piccolo se confrontato al numero di fotoni; sia infatti n_B la densità numerica di barioni osservata (dovuta a stelle, pianeti, etc.), ed n_γ quella dei fotoni costituenti il fondo cosmico a microonde (CBR, Cosmic Background

Radiation). Dal confronto tra queste due quantità si osserva che $n_B \sim 0$ pur essendo $n_B \neq 0$; infatti

$$\frac{n_B}{n_\gamma} \simeq 10^{-9} \div 10^{-10} \quad (1.6)$$

considerando un numero iniziale di barioni e antibarioni confrontabile con quello di fotoni, si ha

$$\frac{n_B - n_{\bar{B}}}{n_B + n_{\bar{B}}} \simeq 10^{-9} \div 10^{-10} \quad (1.7)$$

il significato di questa formula è che soltanto un barione su circa 10^{10} non è stato vittima dell'annichilazione con il corrispondente antibarione.

L'effetto che ha prodotto tale asimmetria sembra quindi essere molto debole; suo compito è quello di portare l'ipotetica simmetria iniziale ($n_B - n_{\bar{B}} \equiv \Delta n_{B\bar{B}} = 0$) al valore osservato attualmente (1.7).

Nel '67 Sakharov[5] espresse per primo le tre condizioni necessarie per consentire un qualsiasi processo di bariogenesi. Esse sono:

1. violazione della conservazione del numero barionico B;
2. violazione delle simmetrie C e CP ;
3. uscita dall'equilibrio termodinamico.

L'importanza della prima condizione è chiara: in assenza di processi in cui $\Delta n_{B\bar{B}} \neq 0$ non è possibile generare alcuna asimmetria tra barioni e antibarioni.

Se C e CP fossero simmetrie esatte, si dimostrerebbe che per ogni transizione $N \rightarrow f$ con violazione di B sarebbe presente il processo complementare $\bar{N} \rightarrow \bar{f}$ con pari frequenza, che compenserebbe il cambiamento nel numero barionico; è quindi fondamentale avere

$$\Gamma(N \xrightarrow{\Delta n_{B\bar{B}} \neq 0} f) \neq \Gamma(\bar{N} \xrightarrow{\Delta n_{B\bar{B}} \neq 0} \bar{f}) \quad (1.8)$$

In caso di equilibrio termodinamico la simmetria CPT si riduce a quella CP , e come si è visto quest'ultima deve essere evitata per consentire bariogenesi. È fondamentale quindi che le violazioni di B e di CP avvengano fuori dall'equilibrio, viceversa la produzione di materia barionica non sarebbe possibile: le tre condizioni devono valere *simultaneamente*.

Con questo criterio a disposizione è possibile iniziare la ricerca di scenari in cui le tre condizioni di Sakharov siano soddisfatte. Ognuno dei modelli attualmente noti presenta però alcuni problemi che saranno evidenziati in quanto segue.

Bariogenesi a GUT

Qualsiasi modello di grande unificazione soddisfa la condizione di violazione del numero barionico. Barioni e leptoni sono infatti inseriti all'interno della stessa rappresentazione del gruppo di gauge, dal quale scaturiscono interazioni che violano B o L (numero leptonic). Tali modelli sono inoltre sufficientemente complessi da fornire diverse sorgenti di *CP violation*.

Inoltre alle energie corrispondenti a GUT ($\sim 10^{19} GeV$), l'universo si trovava in una fase di espansione molto rapida che provocò la fuoriuscita dall'equilibrio. Tutte e tre le condizioni necessarie erano quindi soddisfatte.

Il numero barionico generato a questa scala energetica sarebbe però facilmente diluito durante la successiva evoluzione dell'universo, ed è per questo motivo che attualmente appare improbabile che la bariogenesi che ha dato origine all'attuale asimmetria sia avvenuta a queste energie.

Bariogenesi alla scala elettrodebole

Due caratteristiche fondamentali rendono attraente l'ipotesi di produzione barionica alla scala elettrodebole ($\sim 10^2 GeV$, pari a circa $10^{-6} s$ dopo il Big-Bang): la prima è che in seguito ad essa nessuna diluizione dell'asimmetria generata sarebbe possibile, mentre la seconda è che la fisica di queste energie è accessibile alla tecnologia attualmente disponibile. Appare quindi plausibile una verifica diretta di questo tipo di modelli.

Sfortunatamente anche la bariogenesi a questa scala presenta punti di difficile spiegazione, anche se le tre condizioni di Sakharov sono soddisfatte.

Diversamente dai modelli ad energie di GUT, a questa scala B appare una simmetria esatta; in realtà è noto dal '76[6] che la tecnica dell'espansione perturbativa, solitamente utilizzata nei calcoli relativi al modello standard, non descrive tutta la dinamica racchiusa nella teoria stessa; in particolare effetti non perturbativi possono dare origine alla violazione di $B + L$ mantenendo esatta la simmetria $B - L$.

Nella teoria di gauge dell'interazione debole (QFD) compare un termine che accoppia i campi di materia con i campi di gauge W^μ attraverso una corrente assiale $j_\mu^5 = \bar{\Psi}\gamma_\mu\gamma^5\Psi$. Questa corrente presenta quella che viene solitamente indicata come *anomalia*[7][8][9], avendo divergenza non rigorosamente nulla:

$$\partial^\mu j_\mu^5 = \frac{Ng^2}{16\pi^2} F_{\mu\nu}^a \tilde{F}_a^{\mu\nu} \quad (1.9)$$

N è un numero intero, g è il gauge coupling di $SU(2)_L$ e $F_{\mu\nu}^a$ sono i tensori di campo elettrodebole

$$F_{\mu\nu}^a = \partial_\mu A_\nu^a - \partial_\nu A_\mu^a + g\epsilon^{abc} A_\mu^b A_\nu^c \quad (1.10)$$

mentre $\tilde{F}_{\mu\nu}^a$ sono i loro duali:

$$\tilde{F}_{\mu\nu}^a = \frac{1}{2}\epsilon_{\mu\nu\rho\sigma}F^{a\rho\sigma} \quad (1.11)$$

Introducendo l'*indice di Pontryagin*[9] definito da

$$q = \frac{g^2}{32\pi^2} \int F_{\mu\nu}^a \tilde{F}_a^{\mu\nu} d^4x \quad (1.12)$$

si dimostra che esistono soluzioni (che prendono il nome di istantoni) nelle quali quando t va da $-\infty$ a $+\infty$ lo stato di vuoto iniziale evolve in uno stato di vuoto finale diverso dal primo, e all'evoluzione è associata una variazione unitaria di q .

In pratica $Q = 2Nq$ rappresenta un indice *topologico* associato allo stato di vuoto, e la transizione tra due *ground states* differenti provoca la variazione di questo indice. Essendo però Q pari all'integrale del lato destro della 1.9, deriva che a questo tipo di transizione è associata una variazione della carica legata alla corrente assiale j_μ^5 che comporta la presenza di processi che violano la conservazione del numero barionico. La violazione di B può dunque essere originata nella transizione tra differenti stati di vuoto, transizione che può avvenire, ad esempio, tramite l'attraversamento per effetto tunnel della barriera di potenziale che separa *ground states* diversi.

La probabilità del processo a basse energie è soppressa esponenzialmente dal fattore $\exp(-16\pi^2/g^2) \sim \mathcal{O}(10^{-160})$ che rende l'effetto trascurabile a questi regimi.

Ad alte energie (quando i bosoni deboli W^\pm, Z^0 sono *massless*) la transizione con $\Delta n_{B\bar{B}} \neq 0$ è consentita ma l'effetto globale è, anziché la produzione di carica bariónica, la diluizione di una eventuale asimmetria creata in precedenza.

L'unico punto utile rimasto per la produzione del $\Delta n_{B\bar{B}}$ osservato attualmente è la transizione che separa le due fasi di $SU(2)_L \times U(1)_Y$. Le condizioni necessarie per la bariogenesi si avrebbero soltanto nel caso di transizione di fase del primo ordine, ma quest'ultima (richiedendo una massa per il bosone di Higgs minore di 70GeV) è esclusa dai limiti imposti dalle recenti misure fornite dagli acceleratori.

Anche l'ipotesi di bariogenesi alla scala elettrodebole, pur con le sue allettanti caratteristiche, appare quindi infruttuosa.

Bariogenesi attraverso leptogenesi

Un'altra possibilità è che la produzione di barioni derivi come effetto secondario da una asimmetria nel numero leptónico generata ad energie estremamente elevate; in questo scenario, solitamente la leptogenesi iniziale è catalizzata da un pesante neutrino di Majorana *right-handed*, il cui inserimento è stato inizialmente proposto in modelli non direttamente legati alla bariogenesi (oscillazioni dei neutrini).

Essendo questo neutrino l'immagine attraverso *CPT* di se stesso, la violazione

del numero leptonico segue naturalmente, essendo aperti almeno i due canali di decadimento

$$\begin{aligned}\nu_M &\rightarrow \bar{l}H \\ \nu_M &\rightarrow \bar{l}\bar{H}\end{aligned}$$

Inoltre *CP violation* sarebbe originata da una matrice analoga alla *CKM* per il *mixing* dei neutrini. Essendo poi questi decadimenti molto lenti, avverrebbero fuori dall'equilibrio termodinamico.

L'asimmetria nel numero leptonico così prodotta sarebbe poi trasferita nel numero barionico attraverso transizioni fra stati di vuoto diversi appena sopra la scala elettrodebole, conservando la simmetria esatta $B - L$; il numero barionico ottenuto non avrebbe successivamente alcuna possibilità di diluizione.

1.3 Materia Oscura

Attualmente appare certa la presenza nell'universo di forme di materia esotica non ancora identificate. I motivi che portano a ipotizzarne l'esistenza necessitano una breve introduzione al modello utilizzato per descrivere l'evoluzione su larga scala dell'universo.

1.3.1 Il modello cosmologico standard

Non ci sono ragioni fondate per ipotizzare la presenza di punti privilegiati nell'universo; è dunque plausibile assumere che esso sia *omogeneo*. Dall'osservazione diretta del cielo sembra inoltre che esso abbia le stesse proprietà indipendentemente dalla direzione in cui si guarda (*isotropia*).

La geometria più generale che descrive uno spazio omogeneo ed isotropo è quella definita dalla metrica di *Friedmann-Robertson-Walker* [10], dove la distanza tra due punti infinitamente vicini è data da

$$dl^2 = a^2(t) \left[\frac{dr^2}{1 - kr^2} + r^2(d\Theta^2 + \sin^2\Theta d\Phi^2) \right] \quad (1.13)$$

$a(t)$ prende il nome di fattore di scala, mentre r , Θ e Φ sono dette coordinate comoventi. k è il fattore (costante) che determina il tipo di geometria di cui si sta trattando, e più precisamente si hanno tre casi possibili:

- $k = 0$: lo spazio è piatto (euclideo);
- $k > 0$: lo spazio è sferico (chiuso);
- $k < 0$: lo spazio è iperbolico (aperto).

Negli ultimi due casi si definisce il *raggio di curvatura* $R_c(t) = a(t)/\sqrt{|k|}$.

La curvatura è direttamente legata alla densità di energia dell'universo $\rho(t)$ attraverso l'equazione di Friedmann

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi\mathcal{G}}{3c^2}\rho - \frac{kc^2}{a^2} - \frac{\Lambda}{3} \quad (1.14)$$

la quale esprime la dinamica del tasso di espansione del fattore di scala, solitamente indicato con H

$$H(t) = \frac{\dot{a}(t)}{a(t)} \quad (1.15)$$

in funzione di tre termini che dipendono rispettivamente dalla densità di energia, dalla geometria (k , e $a(t)$) e da una costante (chiamata solitamente *costante cosmologica*) che descrive il contributo dato dall'energia del vuoto (che nel modello standard può essere non nullo).

La quantità $H(t)$ calcolata al tempo attuale t_0 prende il nome di *parametro di Hubble*, e viene parametrizzato in funzione della quantità adimensionale h (parametro di Hubble ridotto) come segue:

$$H_0 = 100h \text{ km s}^{-1} \text{ Mpc}^{-1}, \quad 0.5 < h < 0.8 \quad (1.16)$$

Per capire il significato fisico di H_0 basta calcolare il *redshift* z di una galassia vicina ($L \sim dl$); in questa approssimazione si ottiene

$$z \equiv \frac{v}{c} = H_0 \frac{dl}{c} \quad (1.17)$$

da cui si deriva proprio la relazione postulata da Hubble che lega proporzionalmente la distanza di una galassia e la sua velocità di deriva

$$v = H_0 dl \quad (1.18)$$

Grazie alla 1.14 è possibile, misurando i parametri cosmologici in essa contenuti al tempo attuale t_0 , ricostruire l'evoluzione passata e futura dell'universo fino a raggiungere i limiti di validità dell'equazione stessa (si ipotizza infatti che la 1.14 perda di validità a energie dell'ordine di $E_{Planck} \sim 10^{19} \text{ GeV}$, dove dovrebbe essere sostituita da una teoria quantistica della gravità).

Valutando quindi la 1.14 per $t = t_0$ e dividendo ambo i lati per H_0^2 si ottiene la seguente identità:

$$1 = \Omega_M - \Omega_k + \Omega_\Lambda \quad (1.19)$$

dove si sono definiti

$$\begin{aligned} \Omega_M &= \frac{8\pi\mathcal{G}}{3H_0^2 c^2} \rho_0 \\ \Omega_k &= \frac{kc^2}{H_0^2 a_0^2} \\ \Omega_\Lambda &= \frac{\Lambda}{3H_0^2} \end{aligned} \quad (1.20)$$

Nel caso di universo piatto la quantità $\Omega_0 \equiv \Omega_M + \Omega_\Lambda$ risulterebbe uguale a 1 ($\Omega_k = 0$).

È chiaro quindi che il tipo di geometria dell'universo dipende dalla quantità di energia (materia, radiazione, energia del vuoto, etc.) che esso contiene; si definisce così la *densità critica* $\rho_c(t)$ come il valore che deve assumere la densità totale (somma dei diversi contributi dovuti a materia, radiazione, vuoto, etc.) per avere un universo piatto. Si ha

$$\rho_c(t) = \frac{3c^2 H(t)^2}{8\pi\mathcal{G}} \quad (1.21)$$

da cui si vede che le quantità definite in 1.20 possono essere espresse dal rapporto tra la densità delle varie componenti e la densità critica:

$$\Omega_x = \frac{\rho_x(t_0)}{\rho_c(t_0)} \quad (1.22)$$

La misura di questi parametri è lo scopo principale della cosmologia moderna.

1.3.2 Misure dei parametri cosmologici

Esistono diversi modi più o meno sofisticati che permettono di stimare in maniera indipendente i diversi parametri che appaiono nella 1.19. In quanto segue ne verranno esaminati alcuni.

Cosmic Microwave Background Radiation (CMB)

Il primo che verrà esaminato è basato sulla misura dell'apertura angolare delle perturbazioni osservate nella radiazione cosmica di fondo a microonde. La CMB è l'immagine diretta dell'universo come era all'epoca in cui avvenne la *ricombinazione* tra nuclei ed elettroni; essi erano liberi nel plasma di cui era costituito l'universo prima che la sua temperatura scendesse al di sotto di $\sim 2500K$ ($\sim 10^5 yr$ dopo il Big-Bang). A questa temperatura nuclei ed elettroni cominciarono a unirsi e a formare atomi neutri, praticamente trasparenti alla radiazione con la quale fino a quel momento erano stati in equilibrio termico; la radiazione si disaccoppiò quindi dalla materia ed iniziò a propagarsi liberamente nell'universo, vincolata soltanto dal *redshift*, che la raffreddò fino al punto da raggiungere i $2.725 \pm 0.001K$ [12] che misuriamo attualmente.

Osservando questa radiazione (COBE, BOOMERANG, DASI) è possibile farsi un'idea delle condizioni che vigevano al momento della ricombinazione; in particolare si osservano anisotropie (dovute a onde di pressione presenti nel plasma primordiale) la cui ampiezza media è dell'ordine di $\delta T/\bar{T} \sim 10^{-5}$.

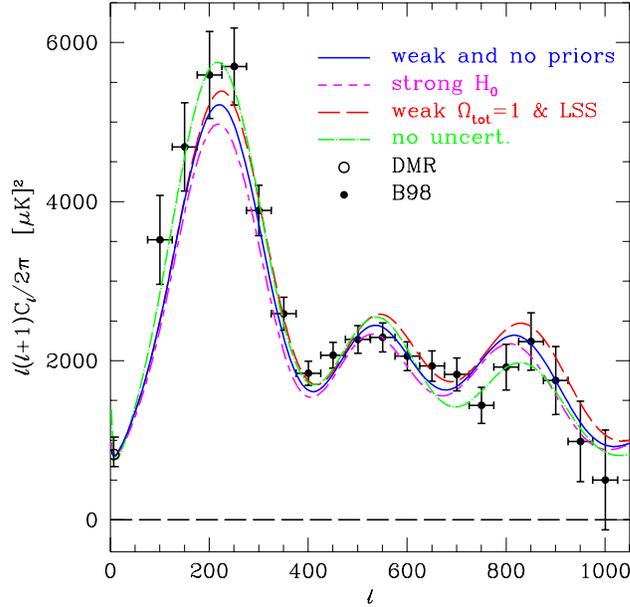


Figura 1.1: Spettro di potenza angolare del CMB misurato a 150GHz da Boomerang[11]. Dalle posizioni dei picchi si ricava l'apertura angolare osservata delle anisotropie sulla superficie di ultimo *scattering*, da cui deriva la stima di $\Omega_0 = 1 \pm 0.1$.

L'apertura angolare di un oggetto che osserviamo dalla terra dipende, oltre che dalle dimensioni dell'oggetto stesso e dalla sua distanza da noi, anche dalla curvatura dello spazio che ci separa da esso; misurando quest'ultima possiamo quindi ottenere una stima del parametro $\Omega_k (= \Omega_M + \Omega_\Lambda - 1)$.

Le analisi effettuate sulle mappe del CMB recentemente fornite da BOOMERANG[11] e DASI[13] (figura 1.1) rivelano che viviamo in un universo sostanzialmente piatto:

$$\Omega_0 \equiv \Omega_M + \Omega_\Lambda = 1 \pm 0.1 \quad (1.23)$$

Inoltre esso è dominato in larga parte (circa $2/3$) dal termine cosmologico Ω_Λ , e soltanto il restante $1/3$ è dovuto alla materia.

Per esaminare più in dettaglio quest'ultimo termine (Ω_M) è comodo separarlo nei diversi contributi dovuti a radiazione (Ω_R), materia barionica (Ω_B), ed una ipotetica forma di materia non barionica (Ω_{NB}). La densità barionica è legata all'ampiezza

relativa delle anisotropie presenti nel CMB ($\delta T/\bar{T}$), da cui deriva la seguente stima per Ω_B :

$$\Omega_B h^2 = 0.02 \pm 0.01 \quad (1.24)$$

Il contributo dei barioni sembra quindi trascurabile rispetto alle altre forme di materia ipotizzate.

Nucleosintesi primordiale

Una misura di Ω_B indipendente da quest'ultima può essere fatta utilizzando i modelli attuali di nucleosintesi primordiale, avvenuta quando la temperatura dell'universo era compresa tra $10^{11} K$ ($\sim 10 MeV$) e $10^9 K$ ($\sim 0.1 MeV$). Essi hanno ormai raggiunto un elevato grado di accuratezza, essendo in grado di prevedere precisamente le abbondanze di tutti gli elementi leggeri prodotti, e presentano come unico parametro libero proprio Ω_B . Il valore ottenuto è compatibile con il precedente:

$$\Omega_B h^2 = 0.020 \pm 0.002 \quad (1.25)$$

Dato che attualmente l'universo attraversa una fase di dominazione della materia sulla radiazione ($\rho_{matter} \gg \rho_{radiation}$), il contributo di Ω_R è sicuramente trascurabile rispetto ad Ω_B ; siamo portati quindi a concludere che la dinamica dell'universo sia determinata in larga misura (circa il 98%) da forme di materia ed energia non ancora chiaramente identificate: Ω_{NB} e Ω_Λ .

Curve di rotazione delle galassie

Un'altro forte indizio (storicamente il primo) che porta a ipotizzare la presenza di materia oscura deriva dall'osservazione delle curve di rotazione delle galassie[15][16]. La loro dinamica è determinata dalla distribuzione di massa $\rho_m(r)$, e misurando il *redshift* in diversi punti è possibile valutare la distribuzione di velocità della materia luminosa al loro interno ($v_{rs}(r)$). È anche possibile misurare la distribuzione della *luminosità* $I(r)$ della materia di cui una galassia è composta.

A questo punto, sembrerebbe ragionevole assumere che la distribuzione della materia luminosa (data da $I(r)$) descriva abbastanza fedelmente la distribuzione di massa della galassia:

$$\rho_{lum}(r) = MI(r) \quad (1.26)$$

Risolvendo l'equazione per il potenziale gravitazionale $\Phi_{lum}(r)$

$$\Delta \Phi_{lum}(r) = 4\pi \mathcal{G} \rho_{lum}(r) \quad (1.27)$$

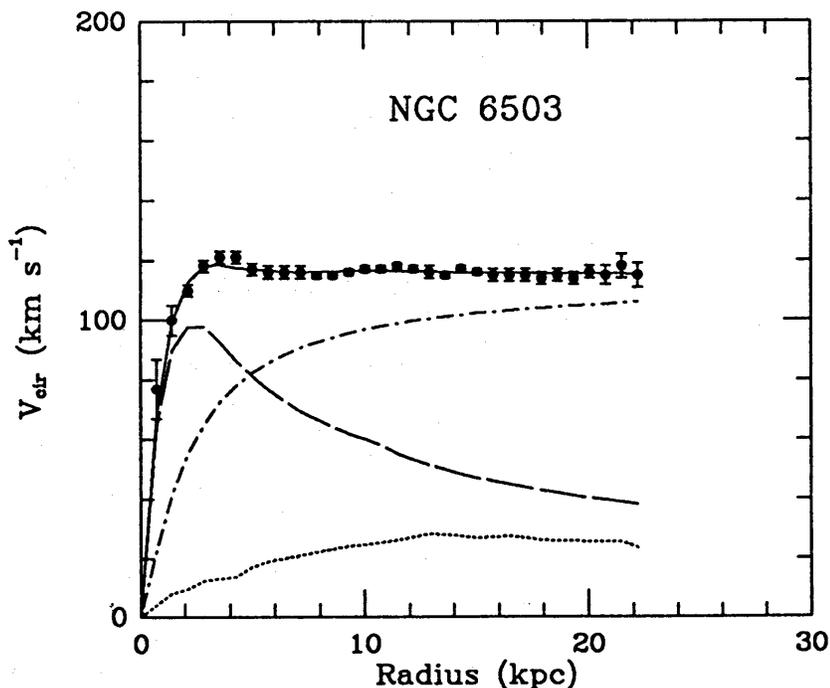


Figura 1.2: Curva di rotazione della galassia NGC6503[14]. La linea tratteggiata indica la curva attesa considerando soltanto la materia del disco, mentre quella tratto-punto deriva dall'alone di materia oscura.

si ottiene la corrispondente distribuzione di velocità $v_{lum}(r)$

$$v_{lum}^2(r) = r \frac{\partial}{\partial r} \Phi_{lum}(r) \quad (1.28)$$

che, calibrata adeguatamente la costante moltiplicativa M , dovrebbe coincidere con $v_{rs}(r)$; in realtà ciò non avviene e $v_{rs}(r)$ risulta sistematicamente maggiore di $v_{lum}(r)$ al crescere di r , come se in realtà la massa distribuita nell'alone galattico fosse molta di più di quella che emette luce ($\rho_m(r) > \rho_{lum}(r)$).

In figura 1.2 è possibile confrontare la curva di rotazione della galassia NGC6503 misurata dalle osservazioni radio del gas di idrogeno nel disco con la curva attesa. Per spiegare il suo andamento è necessario ipotizzare la presenza di un alone di materia oscura che contribuisce alla curva di rotazione come indicato dalla linea tratto-punto.

Dal confronto tra la densità di massa necessaria al spiegare le curve di rotazione

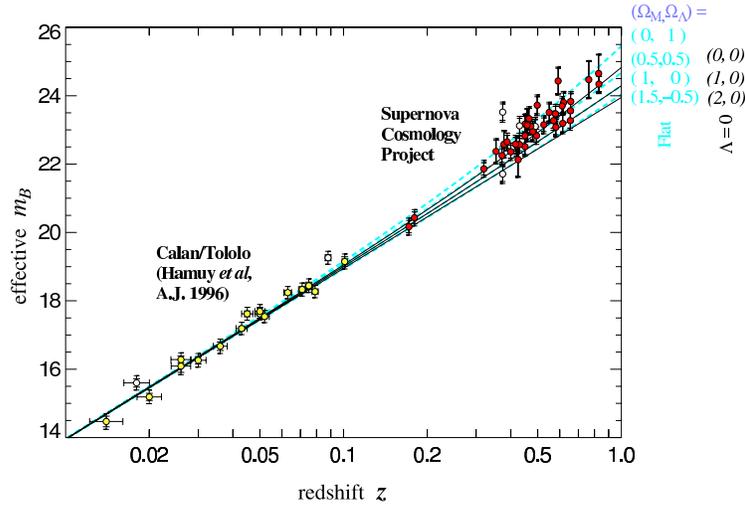


Figura 1.3: Diagramma di Hubble (magnitudine apparente-redshift) per 42 supernovae ad alto *redshift* e 18 a basso *redshift*, entrambe di tipo Ia. Le linee continue sono le curve attese nel caso di curvatura piatta e costante cosmologica nulla ($\Omega_\Lambda = 0$) per diversi valori di Ω_M , mentre quelle tratteggiate corrispondono sempre a curvatura piatta ma $\Omega_\Lambda \neq 0$ [17].

($\rho_m(r)$) e la distribuzione misurata $\rho_{lum}(r)$ si arriva a stimare il contributo che la materia luminosa dà a Ω_0 :

$$\Omega_{lum} < 0.01 \quad (1.29)$$

Sembra quindi che anche parte della materia barionica non emetta luce ($\Omega_B > \Omega_{lum}$).

Supernovae lontane ($z \sim 1$) Ia

Per le supernovae di tipo Ia esiste una relazione semplice che lega la magnitudine assoluta al tasso di decadimento della luminosità. Misurandone quindi la curva di luce è possibile risalire alla luminosità assoluta, e confrontando quest'ultima con la luminosità apparente, si può stimare la distanza dell'oggetto. Per le supernovae

abbastanza vicine a noi ($z < 0.2$) l'andamento della luminosità apparente (una volta corretta per la luminosità assoluta stimata) in funzione del *redshift* conferma pienamente la legge di Hubble 1.18.

A distanze maggiori di $z \sim 0.2$ l'approssimazione fatta nella 1.17 non è più valida, e per le supernovae con $0.2 < z < 1$ (punti rossi nella figura 1.3) diventano importanti gli effetti introdotti dalla relatività generale; dal loro studio è quindi possibile arrivare ad una stima dei parametri cosmologici. Lo studio fatto recentemente su 42 SNIa lontane[17] esclude al 99% di confidenza il caso $\Omega_\Lambda = 0$ e da una stima di Ω_M nel caso di universo piatto ($\Omega_M + \Omega_\Lambda = 1$) pari a

$$\Omega_M = 0.28_{-0.08}^{+0.09} (1\sigma)_{-0.04}^{+0.05} (sys.) \quad (1.30)$$

1.3.3 Candidati a dark matter

Appare sempre più convincente l'ipotesi che esistano forme di materia ed energia "oscura" distribuite nel cosmo, e soltanto una piccola parte di essa sembra costituita da materia barionica non visibile (pianeti giganti, buchi neri, etc.).

La materia oscura barionica non è comunque sufficiente a spiegare le curve di rotazione delle galassie, e di conseguenza anche la parte non barionica deve agglomerarsi in una sorta di *alone galattico*.

Per essere un buon candidato, un'ipotetica particella di *materia oscura* non barionica dovrebbe avere particolari proprietà. Sicuramente, a parte l'interazione gravitazionale (già essa molto debole), dovrebbe interagire debolmente con la materia ordinaria; viceversa sarebbe già stata osservata in laboratorio.

Un'altra caratteristica fondamentale è la sua velocità. Se la materia oscura fosse costituita da particelle relativistiche (*Hot Dark Matter*) non si potrebbero formare aloni perchè la velocità media di queste particelle sarebbe maggiore della velocità di fuga dalle galassie, ed essa si distribuirebbe uniformemente nel cosmo. Un candidato "caldo" è quindi escluso.

Rimane il candidato freddo (*Cold Dark Matter*): particelle massive non relativistiche che interagiscono molto debolmente (o per nulla) con la materia ordinaria. Particelle di questo tipo sono quelle che derivano dalle estensioni *supersimmetriche* del modello standard (teorie di SUSY), nelle quali leptoni e barioni non sono più mondi separati ma appartengono agli stessi multipletti di particelle, e si rendono quindi possibili transizioni leptone-barione. Queste teorie prevedono l'esistenza di nuove particelle che rappresentano i *partners* supersimmetrici delle particelle ordinarie, dotate di un numero quantico conservato R. La più leggera di queste particelle (che solitamente viene indicata con il nome di *neutralino* χ) risulterebbe quindi stabile, ed è attualmente il candidato più gettonato per la materia oscura.

Uno dei modi possibili per verificare la presenza del neutralino è l'osservazione dei

prodotti della sua annichilazione con l'antineutralino:

$$\begin{aligned}\bar{\chi} + \chi &\rightarrow \bar{p} + \chi \\ \bar{\chi} + \chi &\rightarrow e^+ + \chi \\ \bar{\chi} + \chi &\rightarrow 2\gamma\end{aligned}$$

mentre nel caso di un'ipotetica violazione dell'R-parità sarebbero possibili anche i decadimenti

$$\chi, \bar{\chi} \rightarrow \gamma\nu \quad (1.31)$$

Un eccesso del flusso di antiprotoni o positroni nei raggi cosmici, o la rivelazione di γ caratteristici, sarebbero una forte indicazione dell'esistenza di questi processi.

1.4 Raggi cosmici

La parte superiore dell'atmosfera terrestre è continuamente bombardata da un intenso flusso di particelle, la cui energia spazia in un ampio intervallo. Esse hanno origine il larga misura al di fuori del sistema solare, e ci portano quindi importanti informazioni di carattere astrofisico complementari a quelle fornite ad esempio dall'astronomia osservativa. Lo studio dei raggi cosmici è quindi di importanza cruciale, sia come mezzo di verifica e confronto dei risultati forniti da altri rami dell'astrofisica (e non solo), che come sonda per fisica altrimenti inaccessibile.

I raggi cosmici si classificano solitamente in *primari* e *secondari*; dei primari fanno parte, oltre agli elettroni, le particelle generate dalla nucleosintesi (primordiale e stellare) e in seguito accelerate dalle sorgenti astrofisiche; i secondari sono composti dalle particelle prodotte nelle interazioni tra i primari ed il mezzo interstellare. Elettroni, protoni, elio, carbonio, ossigeno, ferro ed altri nuclei, essendo prodotti finali della nucleosintesi stellare, fanno parte dei raggi cosmici primari, mentre elementi come litio, berillio e boro sono secondari.

All'incirca il 79% dei raggi cosmici è costituito da protoni e, della parte restante, il 70% è composto da nuclei di elio.

In figura 1.4 è riportato lo spettro complessivo di tutte le componenti presenti nei raggi cosmici al di sopra dell'atmosfera per energie comprese tra 10^8 e 10^{21} eV. Se si indica con $I(E)$ il flusso differenziale di particelle incidenti

$$I(E) = \frac{N_{particles}}{m^2 s sr GeV nucleon} \quad (1.32)$$

questi spettri seguono abbastanza fedelmente una legge di potenza del tipo

$$I(E_k) \equiv \frac{d\phi}{dE_k} = K E_k^{-\gamma}, \quad K \approx 1.8 \times 10^4 \frac{nucleons}{m^2 s sr GeV^{1-\gamma}} \quad (1.33)$$

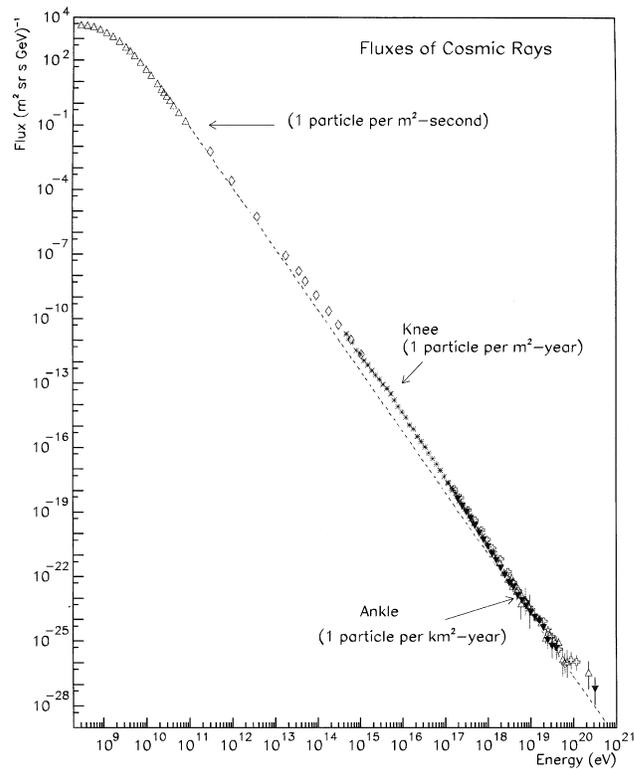


Figura 1.4: Spettro differenziale dei raggi cosmici (tutte le componenti) fuori dall'atmosfera[18]. le frecce indicano il *ginocchio* ($E \sim 10^{16} \text{eV}$) e la *caviglia* ($E \sim 10^{18} \text{eV}$).

per energie comprese tra pochi GeV fino ad oltre 100TeV se si prende γ (indice spettrale differenziale) nell'intervallo 2.5 – 2.7, mentre per energie più basse si nota una forte attenuazione.

Lo spettro di elettroni e positroni è più ripido rispetto a quello dei nuclei, ed inoltre la frazione di positroni decresce dal ~ 0.2 a ~ 0.05 da energie di pochi GeV alle più alte osservate.

Il *rate* di antiprotoni rispetto ai protoni è di circa 2×10^{-4} a $10 - 20 \text{GeV}$ e dipende fortemente, come del resto anche la frazione di positroni, dalla posizione nel ciclo solare.

Attualmente gli spettri di positroni e antiprotoni sono spiegabili con la produzione secondaria dovuta all'interazione dei primari con il mezzo interstellare, e non c'è

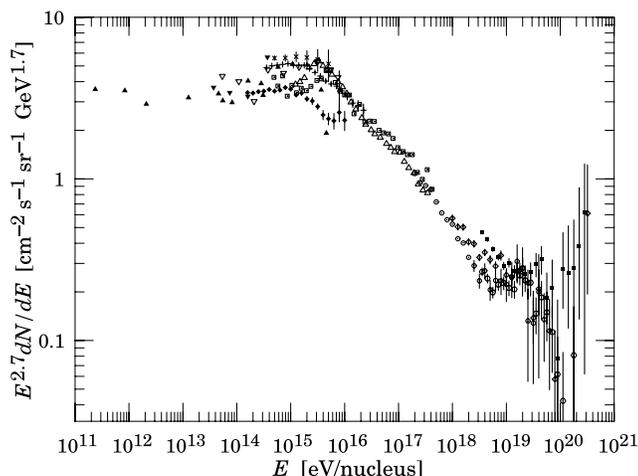


Figura 1.5: Spettro differenziale moltiplicato per $E^{2.7}$ [19]. Sono evidenti i due punti in cui avviene il salto dell'indice spettrale γ .

quindi evidenza di una ipotetica componente primaria di queste particelle derivante da agglomerati macroscopici di antimateria.

I processi di accelerazione che danno origine allo spettro osservato non sono ancora ben noti, anche se i candidati più probabili sono le onde d'urto prodotte nelle esplosioni di supernovae secondo un meccanismo introdotto inizialmente da Fermi (meccanismo di Fermi).

L'attenuazione del flusso che si nota in figura 1.4, per particelle incidenti con energie minori di 1 GeV , è il frutto dell'interazione tra il *vento solare* ed il campo magnetico terrestre, che produce una sorta di "schermatura" per le componenti meno energetiche dei raggi cosmici.

Il vento solare è composto da particelle emesse radialmente dal sole (per lo più protoni), con velocità tale per cui il cammino libero medio ($\lambda \simeq 3 \times 10^{13} m$) è molto maggiore della distanza terra-sole ($d \simeq 1.5 \times 10^{11} m = 1AU$); può quindi essere trattato come un plasma caratterizzato da una conducibilità elettrica infinita. L'effetto che un mezzo di questo tipo ha su un campo magnetico è quello di "congelarlo" al proprio interno, e di conseguenza il vento solare trasporta il campo magnetico del sole in prossimità della terra.

A questo livello avviene l'interazione con il campo magnetico terrestre, causandone una notevole deformazione; il campo risultante ostacola l'arrivo della componente meno energetica ($E < 1\text{GeV}$) dei cosmici sulla terra.

Nel grafico di figura 1.4 sono indicati due punti corrispondenti alle energie $10^{15} - 10^{16} \text{ eV}$ (*ginocchio*) e 10^{19} eV (*caviglia*) a cui corrispondono repentine variazioni dell'indice spettrale γ ; l'ipotesi corrente è che nell'intervallo tra il primo ed il secondo punto alcune (ma non tutte) delle sorgenti acceleratrici dei raggi cosmici raggiungano la massima energia erogabile. In questo intervallo il flusso subisce un'attenuazione dovuta alla progressiva mancanza di queste sorgenti, fino ad assestarsi sulla curva corrispondente ad un nuovo indice spettrale in corrispondenza della caviglia. I due punti sono messi in evidenza in figura 1.5 dove è riportato il grafico del flusso moltiplicato per $E^{2.7}$.

1.4.1 Rigidità magnetica

Siccome i raggi cosmici viaggiano per gran parte del loro tempo attraverso campi magnetici, per trattare la loro dinamica è comodo utilizzare la quantità nota come *rigidità magnetica*, definita da

$$R = \frac{pc}{|z|e} \quad (1.34)$$

dove p è il modulo dell'impulso della particella, $|z|e$ il modulo della sua carica elettrica e c la velocità della luce nel vuoto; R si misura solitamente in gigavolts GV . Il moto di una particella carica in un campo magnetico uniforme \mathbf{B} è descritto da una spirale in cui la distanza r della particella dall'asse della spirale è

$$r = \left(\frac{\gamma m_0 v}{ze} \right) \frac{\sin\theta}{B}$$

indicando con γ il fattore $(1 + v^2/c^2)^{-1/2}$, m_0 la massa a riposo della particella, v e B i moduli rispettivamente di velocità e campo magnetico e θ l'angolo tra la velocità e l'asse della traiettoria.

Siccome $\gamma m_0 v$ non è altro che il modulo dell'impulso si ha

$$r = R \frac{\sin\theta}{Bc}$$

Da questa formula appare chiara l'importanza del parametro R : particelle diverse ma con uguale rigidità si muovono nello stesso modo in un campo magnetico costante. La 1.33 può essere riparametrizzata in funzione della rigidità anzichè dell'energia cinetica E_K , ottenendo

$$\Phi(R) = \Phi_0 R^{-\gamma} \quad (1.35)$$

1.4.2 Cutoff geomagnetico

Il campo magnetico terrestre (dipolare in prima approssimazione) agisce come schermo per i raggi cosmici la cui rigidità non è sufficientemente elevata per penetrare nelle zone più interne. Si può quindi definire la *rigidità di cutoff* $R_c(r)$ come la rigidità minima che deve avere una particella per penetrare fino a distanza r dal centro del dipolo. Nel caso di campo esattamente dipolare si può calcolare R_c analiticamente (Strömer 1930):

$$R_{cutoff} = \frac{M \cos^4 \lambda}{r^2 (1 + \sqrt{1 - \cos \alpha \cos^3 \lambda})^2} \quad (1.36)$$

con M = momento di dipolo, λ = latitudine geomagnetica, r = distanza radiale e α = angolo d'incidenza della particella rispetto all'ovest. Per campi non ideali (come quello terrestre) è necessario ricorrere all'analisi numerica per ottenere una mappa verosimile dell'andamento della rigidità di *cutoff*.

1.4.3 Propagazione ed origine dei raggi cosmici

Diversi esperimenti basati sull'osservazione di raggi γ hanno messo in evidenza il fatto che esiste una forte componente di γ di decadimento del π^0 distribuita su tutta la regione del piano galattico in modo pressapoco uniforme. L'origine di questi π^0 è da imputarsi alle interazioni tra nuclei di energie dell'ordine dei 100 MeV ; è quindi possibile risalire dalle osservazioni di questi γ ad una stima della distribuzione della componente barionica nel mezzo interstellare.

Misurando la composizione chimica dei raggi cosmici incidenti nella parte superiore dell'atmosfera terrestre, vengono alla luce alcune sostanziali differenze rispetto alla composizione chimica della materia che compone il sistema solare. In particolare ci sono tre componenti dei raggi cosmici che sono molto più abbondanti di quello che ci si aspetterebbe:

- gli elementi leggeri Litio, Berillio e Boro;
- i nuclei di 3He rispetto a quelli di 4He ;
- gli elementi appena più leggeri del ferro.

Il processo fondamentale tramite il quale si spiegano queste caratteristiche peculiari va sotto il nome di *spallazione*: avviene sostanzialmente che un nucleo pesante interagisce inelasticamente con un nucleo di H o di He , frammentandosi e liberando così un certo numero di nuclei più leggeri.

Per trattare in maniera formale il problema è necessario introdurre l'*equazione di trasferimento* per i nuclei leggeri[20], che esprime l'andamento della densità numerica nel tempo di ogni componente in funzione dei meccanismi che la influenzano:

- diffusione (coefficiente di diffusione D);
- perdita di energia (dE/dx);
- sorgenti (descritte dal tasso di immissione $Q_i(E, t)$);
- fuga dal volume di confinamento (espressa dal termine N_i/τ_i , con τ_i tempo di confinamento);
- spallazione (dipendente dalla probabilità che la distruzione di un nucleo j in una collisione anelastica sia accompagnata dalla produzione di un nucleo della specie i (P_{ji}), e dal tempo di confinamento τ_j).

L'equazione si esprime quindi così:

$$\frac{\partial N_i}{\partial t} = D\nabla^2 N_i + \frac{\partial}{\partial E} \left[-\frac{dE}{dx} N_i \right] + Q_i - \frac{N_i}{\tau_i} + \sum_{j>i} \frac{P_{ji}}{\tau_j} N_j \quad (1.37)$$

In prima approssimazione si possono trascurare i termini di diffusione e perdita di energia; inoltre, siccome i nuclei a cui siamo interessati hanno un'abbondanza cosmica decisamente bassa, si può tranquillamente porre per essi $Q_i = 0$. È infine comodo esprimere la 1.37 in termini dello spessore di massa $\xi = \rho x = \rho vt$ (con v velocità della particella e ρ densità del gas).

Ciò che si ottiene è il seguente sistema di equazioni:

$$\frac{dN_i(\xi)}{d\xi} = -\frac{N_i(\xi)}{\xi_i} + \sum_{j>i} \frac{P_{ji}}{\xi_j} N_j(\xi) \quad (1.38)$$

dove ora ξ_i e ξ_j rappresentano i rispettivi cammini liberi medi espressi in $kg\ m^{-2}$. Un'ulteriore semplificazione deriva dall'osservazione che i nuclei leggeri a cui siamo interessati (Li, Be e B, che saranno indicati con L) sono originati per spallazione sostanzialmente dalle tre specie più abbondanti: carbonio, azoto e ossigeno (riferiti con il pedice M). Considerando che $N_L(0) = 0$, deriva il seguente sistema di due equazioni per le specie L ed M:

$$\frac{dN_M(\xi)}{d\xi} = -\frac{N_M(\xi)}{\xi_M} \quad (1.39)$$

$$\frac{dN_L(\xi)}{d\xi} = -\frac{N_L(\xi)}{\xi_L} + \frac{P_{ML}}{\xi_M} N_M(\xi) \quad (1.40)$$

Dall'integrazione della 1.39 deriva che l'abbondanza della specie M decade esponenzialmente

$$N_M(\xi) = N_M(0)e^{-\xi/\xi_M}$$

mentre dalla 1.40 deriva

$$\frac{N_L(\xi)}{N_M(\xi)} = \frac{P_{ML}\xi_L}{\xi_L - \xi_M} \left[e^{\left(\frac{\xi}{\xi_M} - \frac{\xi}{\xi_L}\right)} - 1 \right] \quad (1.41)$$

Quest'ultima equazione dipende sostanzialmente da parametri noti. Una stima di P_{ML} si può ottenere da una media pesata delle probabilità di frammentazione di C, N ed O (ottenendo $P_{ML} = 0.28$), mentre i cammini liberi medi sono dati da $\xi_M = 60 \text{ kg m}^{-2}$ e $\xi_L = 84 \text{ kg m}^{-2}$. Dalle misure della composizione chimica dei raggi cosmici (misura la cui accuratezza sarà notevolmente aumentata da AMS-02) deriva anche una stima del rapporto tra le abbondanze dei nuclei L e nuclei M ($N(\xi)_L/N(\xi)_M = 0.25$).

Sostituendo questi valori nella 1.41 si trova che lo spessore di massa che i raggi cosmici devono aver attraversato per dare origine alle abbondanze relative osservate in prossimità della terra è $\xi = 48 \text{ kg m}^{-2}$. Questo valore è sorprendentemente simile ai cammini liberi medi degli elementi di tipo M, indicando che il modello, seppur notevolmente semplificato, è coerente con le osservazioni attuali.

Con un calcolo analogo è possibile studiare la produzione di ${}^3\text{He}$ per spallazione di ${}^4\text{He}$ presente nel gas interstellare, ottenendo che per esso lo spessore di massa attraversato deve essere $\xi = 50 \text{ kg m}^{-2}$, in ottimo accordo con il valore precedente. Le cose si complicano studiando la spallazione del ferro; in questo caso per riprodurre i dati osservati è necessaria una trattazione più accurata, utilizzando piuttosto che semplici cammini liberi medi, distribuzioni di questi che tengano conto della diffusione, della perdita di energia e di tutti i fattori presenti nella 1.37.

Uno di questi fattori è legato al *tempo di confinamento*. La necessità che i raggi cosmici siano infatti confinanti per un certo tempo all'interno di un certo volume appare chiara considerando che se così non fosse, essi non potrebbero attraversare uno spessore di massa sufficiente a dare origine alle abbondanze chimiche effettivamente osservate.

Assumendo una densità numerica media per la materia interstellare di 10^6 m^{-3} e velocità relativistiche ($\beta \simeq 1$) si arriva a stimare il tempo di permanenza utile (nel senso che durante questo tempo avviene spallazione) necessario alla produzione delle abbondanze osservate:

$$\tau_e \simeq 3 \times 10^6 \text{ yr}$$

Considerando le dimensioni della via lattea ($1 - 10 \text{ kpc}$), se i raggi cosmici si muovessero liberamente il loro tempo di permanenza utile sarebbe compreso tra 3×10^3 e 3×10^4 anni, largamente insufficiente a dare origine alle abbondanze osservate.

Il modello tipicamente usato per descrivere il confinamento va sotto il nome di modello *leaky box*. Esso descrive particelle che diffondono liberamente all'interno di un certo volume e che vengono riflesse dalle pareti. C'è però una certa probabilità che,

incidendo sulla parete stessa, la particella esca dal volume di confinamento.

Una questione ancora aperta riguarda l'estensione del volume entro il quale i raggi cosmici sono confinati; alcuni modelli lo considerano pari all'incirca a quello della galassia (un disco di circa $10kpc$ di diametro spesso $300 - 500pc$), mentre altri ipotizzano la presenza di un alone che si estende oltre il volume occupato dalla galassia stessa.

1.4.4 Accelerazione dei raggi cosmici

L'equazione di trasporto applicata ai raggi cosmici mantiene intatto l'andamento del loro spettro energetico nel caso in cui le sorgenti diano origine ad uno spettro di potenza (eq. 1.33).

Un meccanismo di accelerazione che produce uno spettro di questo tipo è quello introdotto da Fermi nel '49, e va sotto il nome di *meccanismo di Fermi*.

La prima versione originariamente introdotta descrive il processo di accelerazione come una sequenza di urti delle particelle contro "specchi magnetici" associati ad irregolarità nei campi magnetici galattici, che si muovono con moto casuale nello spazio con velocità V . Ad ogni urto con uno di questi fronti la particella acquista energia, e nel caso in cui la permanenza nella zona di accelerazione sia limitata nel tempo (τ_{fuga} finito) lo spettro energetico prodotto assume proprio l'andamento di potenza osservato.

Un problema fondamentale di questo processo è la sua lentezza; infatti il guadagno energetico medio in ogni urto è pari a

$$\left\langle \frac{\Delta E}{E} \right\rangle = \frac{8}{3} \left(\frac{V}{c} \right)^2 \quad (1.42)$$

che è del secondo ordine in V/c (meccanismo di Fermi del secondo ordine). Siccome $V/c \leq 10^{-4}$ e il numero di collisioni di questo tipo è stimato in una all'anno, l'incremento energetico che ne deriva risulta irrisorio rispetto alle energie effettivamente osservate.

Soltanto nel caso di collisioni esclusivamente frontali il guadagno energetico risulta del primo ordine (meccanismo di Fermi del primo ordine). Una situazione in cui entra in gioco questo meccanismo è il passaggio di una particella attraverso il fronte di una onda d'urto. Onde d'urto utili a questo scopo sono ad esempio quelle prodotte nelle esplosioni di supernovae, in cui la velocità della materia espulsa (che può arrivare a $10^4 km s^{-1}$) è decisamente maggiore della velocità del suono nel mezzo (dell'ordine di $10km s^{-1}$).

Un esame dettagliato della situazione[21][22] mostra che la particella guadagna energia sia quando attraversa il fronte da davanti che quando lo attraversa da dietro, in modo totalmente simmetrico. Il guadagno in energia per attraversamento del fronte

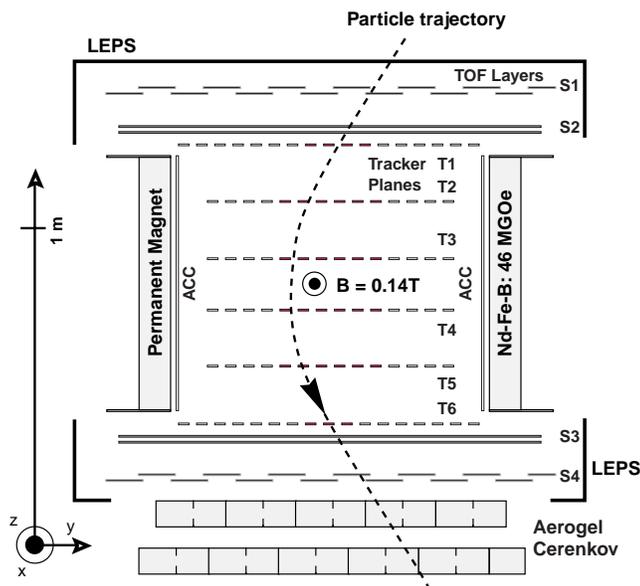


Figura 1.6: Vista schematica di AMS-01 dove si notano il magnete cilindrico permanente, i 6 piani di microstrips al silicio del tracker, i 4 piani del TOF, il cerenkov ad aerogel e i contatori per le anticoincidenze.

risulta

$$\left\langle \frac{\Delta E}{E} \right\rangle = \frac{2V}{3c} \quad (1.43)$$

Il processo è abbastanza lento anche in questo caso; il limite superiore dell'energia erogabile in questo tipo di processi è stato stimato in circa $10^{14} \text{ GeV nucl}^{-1}$ [23], ed è quindi inadeguato per spiegare l'origine dei numerosi raggi cosmici osservati ad energie più elevate.

1.5 Risultati dell'esperimento AMS-01

Nel giugno del 1998 si è svolta una missione nella quale un prototipo dell'apparato, AMS-01, ha compiuto un volo di prova della durata di dieci giorni a bordo dello *shuttle Discovery* (missione NASA STS-91). Il successo della missione si è concretizzato nella conferma dell'affidabilità delle soluzioni tecnologiche adottate, e soprattutto nella raccolta di 100 milioni di eventi che hanno permesso un notevole affinamento dei risultati precedenti riguardanti la fisica sondata da AMS.

In figura 1.6 si può osservare la struttura del rivelatore in questa prima fase, in cui

si notano il magnete permanente (Nd-Fe-B, potere anallizzante $BL^2 = 0.14Tm^2$), il tracker (6 piani di tracciatori al Silicio, $\sigma_y = 20\mu m$, $\sigma_x = 33\mu m$), il TOF (4 piani di scintillatori plastici, $\sigma = 120ps$), il contatore Cerenkov ad aerogel (indice di rifrazione $n = 1.035$) e i contatori di anticoincidenze (contatori plastici sulla superficie interna del magnete).

1.5.1 Misura \overline{He}/He

Dei possibili candidati nessuno ha superato i tagli per la selezione di nuclei di antielio, metre 2.86×10^6 particelle sono state identificate come elio con rigidità $R \in [1, 140]GV$. In questo intervallo il limite sul rapporto \overline{He}/He è stato fissato in 1.1×10^{-6} con un livello di confidenza del 95%[24].

Si è giunti a questo valore facendo l'ipotesi che lo spettro di \overline{He} segua lo stesso andamento dello spettro di He , assunzione comunemente fatta nella maggior parte degli esperimenti sull'antimateria.

Gli eventi selezionati sono stati utilizzati per fittare la legge di potenza 1.35 per rigidità da $20GV$ a $200GV$ [25], ottenendo le seguenti stime per l'indice spettrale γ e la costante ϕ_0 :

$$\gamma = 2.740 \pm 0.026 \quad (1.44)$$

$$\phi_0 = 2.52 \pm 0.22 \pm 0.14(\gamma) \text{ } GV^{2.74}/(m^2 s sr MeV) \quad (1.45)$$

1.5.2 Spettro dei protoni

Per la misura dello spettro dei protoni sono stati usati i dati presi durante due periodi: nel primo il detector era diretto verso lo zenit (misura dei protoni discendenti) mentre nel secondo era diretto verso il nadir (protoni ascendenti). In figura 1.7 si osservano gli spettri relativi a dieci intervalli di latitudine geomagnetica.

Si vede chiaramente la diminuzione della rigidità di cutoff con l'avvicinarsi alle zone polari. Al di sotto di questo valore è presente un altro spettro dovuto a particelle di bassa energia intrappolate nel campo geomagnetico. Chiaramente questa caratteristica non si osserva nello spettro dei protoni ascendenti, essendo questi ultimi soltanto di origine atmosferica.

I dati misurati sono stati fittati con la 1.35 per $10 < R < 200 \text{ } GV$, ottenendo per ϕ_0 e γ i valori[26]:

$$\gamma = 2.78 \pm 0.028 \quad (1.46)$$

$$\phi_0 = 17.1 \pm 1.5 \pm 1.5(\gamma) \text{ } GV^{2.78}/(m^2 s sr MeV) \quad (1.47)$$

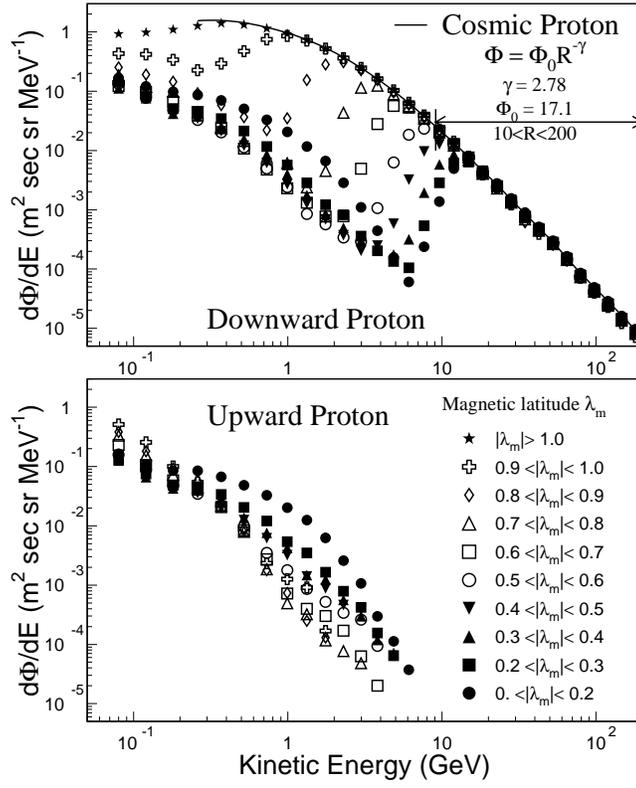


Figura 1.7: Spettri misurati da AMS-01 dei protoni discendenti (*Downward Protons*), e ascendenti (*Upward Proton*) per dieci intervalli della latitudine geomagnetica. Nel primo caso è evidente l'effetto del cutoff, e la presenza sotto ad esso di un altro spettro.

1.5.3 Spettro dei leptoni

Il procedimento usato per la misura dello spettro dei leptoni (e^+ ed e^-) ricalca quello precedente, anche se l'accettanza del rivelatore è limitata a 25° rispetto all'asse z perchè per il riconoscimento è necessario il passaggio della particella attraverso il Cherenkov. I range di energia effettivi sono $[0.2, 40] \text{GeV}$ per elettroni e $[0.2, 3] \text{GeV}$ per positroni.

In figura 1.8 si possono osservare gli spettri di e^+ ed e^- per diversi intervalli di latitudine magnetica, sia per particelle discendenti (*downward*) che per particelle ascendenti (*upward*).

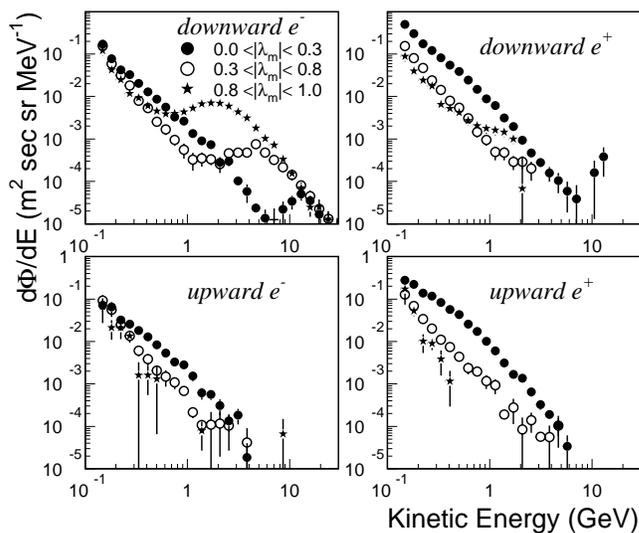


Figura 1.8: Spettri misurati da AMS-01 degli elettroni e dei positroni sia discendenti (*Downward* e^- , e^+) che ascendenti (*Upward* e^- , e^+) per diversi intervalli di latitudine geomagnetica.

1.6 Fisica per AMS-02

Il rivelatore AMS nella sua seconda fase (AMS-02) sarà integrato con nuovi apparati che ne incrementeranno notevolmente le prestazioni. Inoltre è di primaria importanza il fatto che il periodo di presa dati si protrarrà per ben tre anni. Questo insieme di fattori permetterà al rivelatore di migliorare ogni misura fatta precedentemente nel campo dei raggi cosmici a livello dell'alta atmosfera.

La statistica che raccoglierà AMS-02 sarà superiore a quella di AMS-01 di un fattore 10^3 , mentre il limite superiore sul momento verrà incrementato di circa sei volte.

I punti chiave su cui AMS-02 tenterà di gettare un po' di luce si possono riassumere in:

1. Limite sulla presenza di antimateria di origine cosmica (rates \overline{He}/He e \overline{C}/C): verranno raccolti circa 10^9 nuclei di He di rigidità compresa tra 1 e 1400GV.
2. Rivelazione dei prodotti di processi implicanti la presenza di *dark matter* di tipo non barionico (spettri di \overline{p} , e^+ e γ): ci si aspetta una statistica di $4 \times 10^6 e^+$ e $10^6 \overline{p}$.
3. Misura accurata della composizione spettrale della varie componenti dei raggi cosmici (D , 3He , B , C , 9Be e ${}^{10}Be$).

L'importanza del primo punto dovrebbe essere ormai chiara visto quanto detto in precedenza; la presenza o meno di antimateria di origine cosmica è strettamente legata alla fisica delle interazioni fondamentali, essendo la violazione di CP e la conservazione (o meno) del numero barionico B due degli aspetti cruciali non ancora chiariti. Esiste un forte legame anche con la cosmologia, dovuto alla dipendenza dei modelli di (eventuale) bariogenesi dalla situazione termodinamica dell'universo durante le varie fasi della sua evoluzione.

Il legame con la cosmologia e l'astrofisica si rafforza nel secondo punto, nel quale è evidente che risultati significativi arriveranno soltanto come conseguenza della collaborazione e confronto tra esperimenti diversi e complementari, ognuno dei quali deve portare il contributo che il proprio ramo della fisica può dare alla misura dei parametri cosmologici, e nella verifica dei modelli ad essi legati.

In questa ottica AMS-02 si propone di utilizzare i raggi cosmici come sonda per la materia oscura non barionica, evidenziando possibili deviazioni degli spettri di \bar{p} , e^+ e γ da quelli attesi.

Il terzo punto non è certamente di minor importanza: la composizione chimica dei raggi cosmici che arrivano in prossimità della terra è direttamente legata ai meccanismi di origine, accelerazione e propagazione dei raggi cosmici stessi nel mezzo interstellare (o intergalattico), che a loro volta dipendono fortemente da fenomeni fisici particolari non ancora ben noti. Si pensi ad esempio che non ci sono ancora modelli astrofisici in grado di rendere conto dell'accelerazione di particelle alle energie che si hanno nella coda dello spettro rappresentato in figura 1.33.

Inoltre la misura accurata delle abbondanze chimiche delle particelle incidenti sull'alta atmosfera è necessaria per la corretta interpretazione dei dati forniti da alcuni esperimenti sull'oscillazione dei neutrini. Gli esperimenti di questo tipo che si basano sull'osservazione dei processi dovuti ai neutrini di origine extraterrestre (per esempio Superkamiokande, che studia i neutrini solari) utilizzano complesse simulazioni montecarlo che forniscono il flusso di neutrini prodotti nelle interazioni tra raggi cosmici e la parte superiore dell'atmosfera; va da sé che l'accuratezza della simulazione dipende in prima istanza dal grado di conoscenza che si ha dei cosmici incidenti e delle sezioni d'urto relative ai processi di produzione di neutrini nelle interazioni in gioco.

AMS-02 fornirà quindi un aiuto sostanziale a questo tipo di fisica, sopperendo alla prima richiesta raccogliendo una statistica relativa a tutti gli elementi fino all'ossigeno decisamente superiore a quanto fatto in precedenza dagli esperimenti su pallone aerostatico.

Capitolo 2

Lo spettrometro AMS-02

2.1 Funzionamento di uno spettrometro

Uno spettrometro di massa è uno strumento che cerca di identificare le particelle che lo attraversano misurando alcune loro proprietà peculiari, come perdita di energia (dE/dx), impulso (p) e velocità ($\beta = v/c$). Conoscendo le relazioni che legano queste quantità a quelle incognite (una particella è identificata quando si conoscono massa a riposo m_0 e carica elettrica Ze), è possibile passare dai valori misurati delle prime ad una stima delle ultime.

Per ottenere queste quantità è necessario ricorrere all'utilizzo di diversi apparati complementari, ognuno dei quali specializzato in una misura particolare. Di sovente accade inoltre che l'intervallo utile nel quale un apparato è in grado di effettuare correttamente una misura è limitato; diventa quindi necessario l'utilizzo di apparati diversi che producono misure della stessa quantità ma in zone diverse dello spettro di valori a cui si è interessati.

2.1.1 Perdita di energia

Quando una particella carica attraversa uno strato di materia, subisce una serie di urti anelastici con gli atomi che la compongono, provocandone l'eccitazione (collisioni *soft*) o la ionizzazione (collisioni *hard*). L'effetto cumulativo di queste ripetute collisioni si traduce per la particella incidente in una perdita continua di energia. La quantità di energia persa per unità di lunghezza dipende, oltre che dal materiale attraversato, dalle caratteristiche della particella incidente.[27]

Il calcolo quantistico dell'espressione che descrive la perdita di energia fu portato a

termine per la prima volta da *Bethe* e da *Bloch*, da cui la formula prende il nome:

$$-\frac{dE}{dx} = 2\pi N_a r_e^2 m_e c^2 \rho \frac{Z z^2}{A \beta^2} \left[\ln \left(\frac{2m_e \gamma^2 v^2 W_{max}}{I^2} \right) - 2\beta^2 - \delta - 2\frac{C}{Z} \right] \quad (2.1)$$

Nella 2.1 appaiono alcuni parametri ben noti (come ad esempio m_e , r_e , etc.) ed alcune quantità il cui significato è il seguente:

- I : potenziale di eccitazione medio;
- ρ : densità del materiale attraversato;
- δ : correzione di densità;
- C : correzione di shell;
- W_{max} : massima energia trasferita in una singola collisione.

la costante moltiplicativa iniziale assume il valore

$$2\pi N_a r_e^2 m_e c^2 = 0.1535 \text{ MeV cm}^2 / g$$

Nel caso di elettroni e positroni la formula va modificata. In primo luogo la loro massa è circa uguale a quella delle particelle su cui incidono (gli elettroni atomici), quindi la traiettoria subisce notevoli deviazioni; inoltre gli elettroni sono soggetti ad urti tra particelle identiche, e quindi indistinguibili. Per finire è necessario tener conto del *Bremsstrahlung*, il cui contributo diventa importante vista la piccola massa delle due particelle.

La formula corretta è in questo caso

$$\left(\frac{dE}{dx} \right)_{tot} = \left(\frac{dE}{dx} \right)_{coll} + \left(\frac{dE}{dx} \right)_{rad}$$

Il primo termine (che risulta molto simile alla 2.1) tiene conto della perdita di energia per collisione, mentre il secondo (che diventa dominante al di sopra di un'energia critica E_c in cui i contributi si uguagliano) descrive il contributo energetico ceduto a causa della radiazione di *Bremsstrahlung*:

$$-\left(\frac{dE}{dx} \right)_{rad} = N E_0 \phi_{rad}$$

dove N è il numero di atomi per cm^3 , E_0 è l'energia iniziale della particella e ϕ_0 è una quantità funzione soltanto del materiale attraversato.

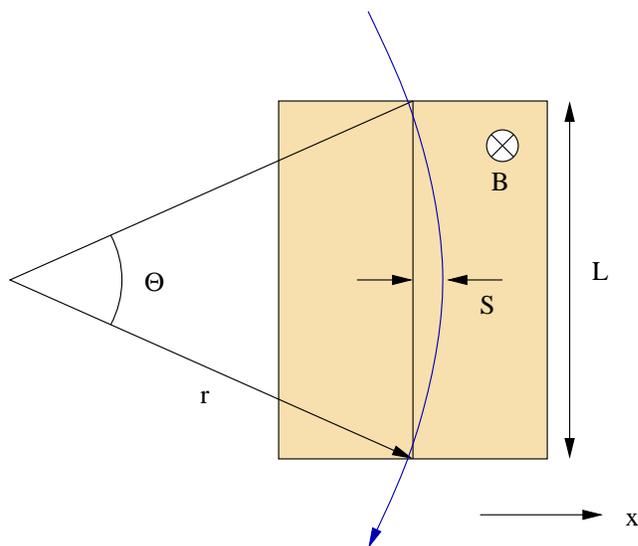


Figura 2.1: Moto di una particella carica in un campo magnetico costante.

2.1.2 Misura del momento

La tecnica solitamente usata per ricavare il momento di una particella si basa sulla misura della curvatura della sua traiettoria indotta da un campo magnetico noto creato proprio a questo scopo (*tracciamento*).

Nel moto di una particella in un campo magnetico costante (figura 2.1) il momento p si esprime in funzione della carica q , dell'intensità del campo B e del raggio di curvatura della traiettoria r come

$$p = qBr \quad (2.2)$$

Dato che $L = 2r \sin(\Theta/2)$, per piccoli angoli ($\Theta \sim 0$) vale

$$\Theta \simeq \frac{LqB}{p} \quad (2.3)$$

e sviluppando in Θ la sagitta $S = r(1 - \cos(\Theta/2))$ si ha

$$S \simeq r \frac{\Theta^2}{8} = \frac{qBL^2}{8p} \quad (2.4)$$

da cui si ottiene p in funzione di q , B , L , e S . La precisione che si ha sulla misura di p dipende da quella sulla sagitta; quest'ultima si ottiene dalla misura di tre valori

della posizione lungo x con precisione $\sigma(x)$:

$$S = x_2 - \frac{x_1 + x_3}{2} \quad \frac{\sigma(S)}{S} = \frac{\sqrt{\frac{3}{2}}\sigma(x)}{S} = \frac{8\sqrt{\frac{3}{2}}\sigma(x)p}{qBL^2} \quad (2.5)$$

Avendo N misure equidistanti $\sigma(S)/S$ diventa proporzionale all'inversa della radice di N [28].

Un altro effetto che bisogna mettere in conto nella valutazione della precisione su p è lo *scattering multiplo*. Nell'attraversare lo spettrometro la particella interagisce con i nuclei del materiale di cui è composto, subendo ripetute deviazioni dalla traiettoria governate dalla sezione d'urto di Rutherford. Il risultato netto è uno spostamento laterale rispetto al percorso originale e una deviazione angolare. L'effetto però non dipende dal momento, e diventa quindi importante solo a basse energie.

2.1.3 Misura della velocità

Esistono diversi sistemi che consentono di misurare la velocità di una particella, e si differenziano sostanzialmente per l'intervallo di energia in cui forniscono risultati. AMS-02 incorporerà tre apparati dedicati alla misura della velocità:

- *Time Of Flight (TOF)*, dedicato appunto alla misura del tempo di volo (oltre che al trigger);
- *Ring Imaging Cherenkov (RICH)*, dedicato alla raccolta del cono di luce Cherenkov, la cui apertura dipende dalla velocità;
- *Transition Radiation Detector (TRD)*, dedicato alla rivelazione della radiazione di transizione, la cui intensità dipende dalla velocità.

Tempo di volo

Il tempo di volo t è il tempo impiegato da una particella per attraversare la distanza tra due punti nei quali viene rivelato suo il passaggio, preferibilmente con rivelatori veloci; misurando t , si ottiene la velocità conoscendo la distanza L tra i due punti dalla relazione $t = L/\beta c$.

La differenza nel tempo di volo tra due particelle di ugual momento è pari a

$$\Delta t = \frac{L}{c} \left(\frac{1}{\beta_1} - \frac{1}{\beta_2} \right) = \frac{L}{c} \left(\sqrt{1 + m_1^2 c^2 / p^2} - \sqrt{1 + m_2^2 c^2 / p^2} \right) \simeq \frac{Lc}{2p^2} (m_1^2 - m_2^2) \quad (2.6)$$

quindi per distinguere due particelle di massa m_1 ed m_2 è necessario misurare il tempo di volo con una risoluzione migliore di Δt . Se ad esempio $L = 1 \text{ m}$ e la risoluzione è pari a 150 ps , la separazione tra elettroni e antiprotoni è possibile fino a impulsi di circa $3 \text{ GeV}/c$.

Radiazione Cherenkov

La radiazione Cherenkov è emessa da una particella la cui velocità supera quella della luce nel mezzo attraversato. È quindi un effetto che si produce soltanto al di sopra di una certa velocità di soglia $\beta_{thr} = 1/n$ (n è l'indice di rifrazione).

L'angolo tra il vettore d'onda della radiazione Cherenkov e la direzione di propagazione della particella soddisfa la relazione

$$\cos \Theta = \frac{1}{n\beta} \quad (2.7)$$

misurando quindi l'apertura del cono di luce prodotto si ottiene la velocità β .

L'angolo Θ va da 0, quando $\beta = \beta_{thr}$, a $\Theta_{max} = \arccos(1/n)$ quando $\beta = 1$; il numero di fotoni emessi per unità spazio percorso per unità di lunghezza d'onda è

$$\frac{dN}{dx d\lambda} = \frac{2\pi z^2 \alpha}{\lambda^2} \sin^2 \Theta \quad (2.8)$$

ed esprimendo λ come hc/E si vede che $dN/dE = costante$.

Negli apparati *RICH* la misura di Θ viene fatta determinando la conica risultante dall'intersezione tra il cono di luce prodotto dalla particella con un piano fotosensibile. Il cono è prodotto nell'attraversamento di uno strato di materiale scelto sostanzialmente in funzione dell'indice di rifrazione, in modo da ottimizzare l'angolo di apertura del cono prodotto e il numero di fotoni emessi.

Radiazione di transizione

Nell'attraversamento di un mezzo con indice di rifrazione discontinuo viene emessa una piccola dose di radiazione elettromagnetica; l'energia irradiata nell'attraversare la discontinuità è

$$W = \frac{1}{3} \alpha \hbar \omega_p \gamma \quad (2.9)$$

γ è l'usuale coefficiente relativistico, mentre ω_p (detta frequenza di plasma) è pari a

$$\omega_p = \sqrt{\frac{N_e e^2}{\epsilon_0 m_e}} \quad (2.10)$$

Siccome W è proporzionale a γ la rivelazione della radiazione di transizione è possibile soltanto per particelle ultrarelativistiche e solitamente è usata per l'identificazione di e^+e^- ad alta energia.

$\hbar\omega_p$ è dell'ordine di 20 eV per radiatori plastici, e il numero di elettroni emessi è molto basso:

$$N_{ph} \simeq \frac{W}{\hbar\omega} \propto \alpha \simeq \frac{1}{137} \quad (2.11)$$

da cui la necessità di costruire apparati formati da molti strati sottili di radiatore intervallati con materiale sensibile per ottenere un segnale rivelabile. L'energia tipica dei fotoni emessi per radiazione di transizione è

$$\hbar\omega \simeq \frac{1}{4}\hbar\omega_p\gamma \quad (2.12)$$

che risulta dell'ordine del keV ; il massimo dei raggi X prodotti viene inoltre emesso ad un angolo proporzionale a $1/\gamma$, quindi la radiazione rimane vicina alla traccia.

2.1.4 Identificazione delle particelle

Dalla 2.1 si vede che, scelto il materiale assorbitore, la perdita di energia è funzione della carica, della velocità e della massa a riposo della particella. Nell'espressione della rigidità data in 1.34 compaiono ancora una volta le stesse quantità z , β e m_0 . Lo spettrometro misura le quantità dE/dx , R e β da cui è possibile risalire alla massa ed alla carica della particella in esame invertendo le 1.34 e 2.1, identificando così la particella incognita.

2.2 Il rivelatore nella prima fase

In figura 2.2 si può osservare la struttura schematica del rivelatore nella fase di test (volo STS-91).

Esso consiste in uno spettrometro magnetico ad alta accettazione (circa $0.6m^2sr$ ridotta a $0.36m^2sr$ a causa della mancanza di tempo per il completamento del *tracker* per il volo di prova). Una particella carica che attraversa il sistema “triggera” l'apparato tramite il TOF, che fornisce anche la misura della velocità della particella con una risoluzione di circa $120ps$ su una distanza tra i due piani di $1.4m$, e una misura di dE/dx .

La misura dell'impulso viene fornita dal tracker, composto da 6 piani di paralleli costituiti da *wafers* di silicio spessi $300\mu m$ su cui sono depositate delle *microstrips* conduttrici su entrambe le facce, ma in direzioni ortogonali tra loro. In questo modo si ottengono misure con una accuratezza di $10\mu m$ sul piano di curvatura e di $30\mu m$ sul piano ad esso ortogonale.

Il campo magnetico è prodotto all'interno dello spettrometro da un magnete permanente costruito con una lega di $Nd - Fe - B$ che fornisce un rapporto *Intensità di campo/peso* decisamente favorevole. Si ottiene infatti un potere analizzante di $0.15Tm^2$ con un peso inferiore a $2t$.

Il sistema per le anticoincidenze consente di scartare eventi prodotti dall'interazione dei cosmici con il magnete o con le strutture di sostegno, ed è costituito da un

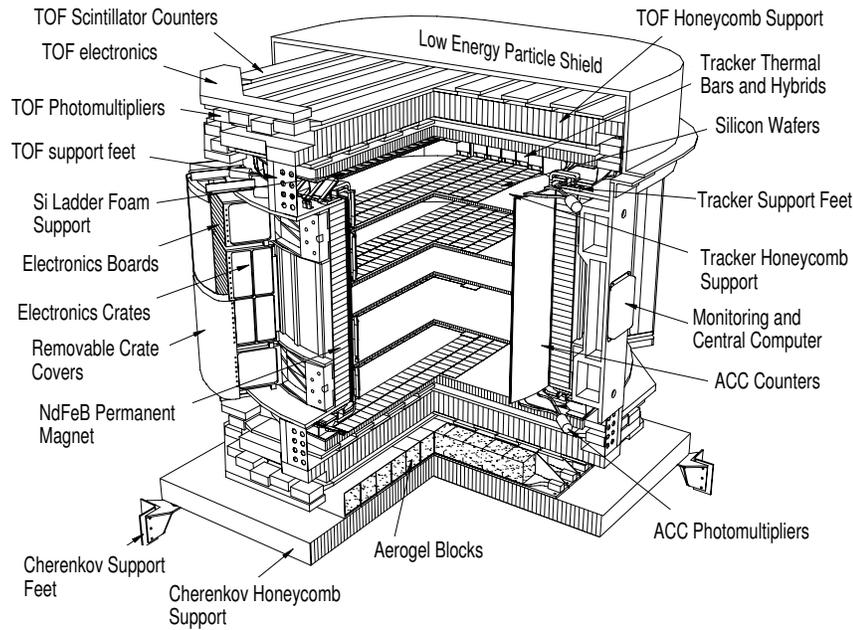


Figura 2.2: Struttura dell'apparato AMS01

cilindro di plastico scintillante dello spessore di 1cm che riveste internamente il magnete. Il segnale fornito dai fototubi incaricati della lettura di questi contatori viene utilizzato come veto per il *trigger*.

Allo scopo di estendere l'intervallo di rigidità in cui è possibile separare gli antiprotoni, AMS-01 è dotato anche di un contatore Cherenkov a soglia, costituito da due piani di celle di *aerogel* (indice di rifrazione $n - 1 = 7 \times 10^{-3}$).

2.3 Struttura di AMS-02

In seguito all'esito positivo del volo di prova, si è deciso di incrementare le prestazioni dell'apparato nella seconda fase della missione (figura 2.3); in particolare il magnete permanente sarà sostituito da un magnete a superconduttore, e un contatore RICH prenderà il posto del Cherenkov a soglia. Inoltre verranno aggiunti due *subdetectors*: un calorimetro elettromagnetico e un rivelatore a radiazione di transizione (TRD).

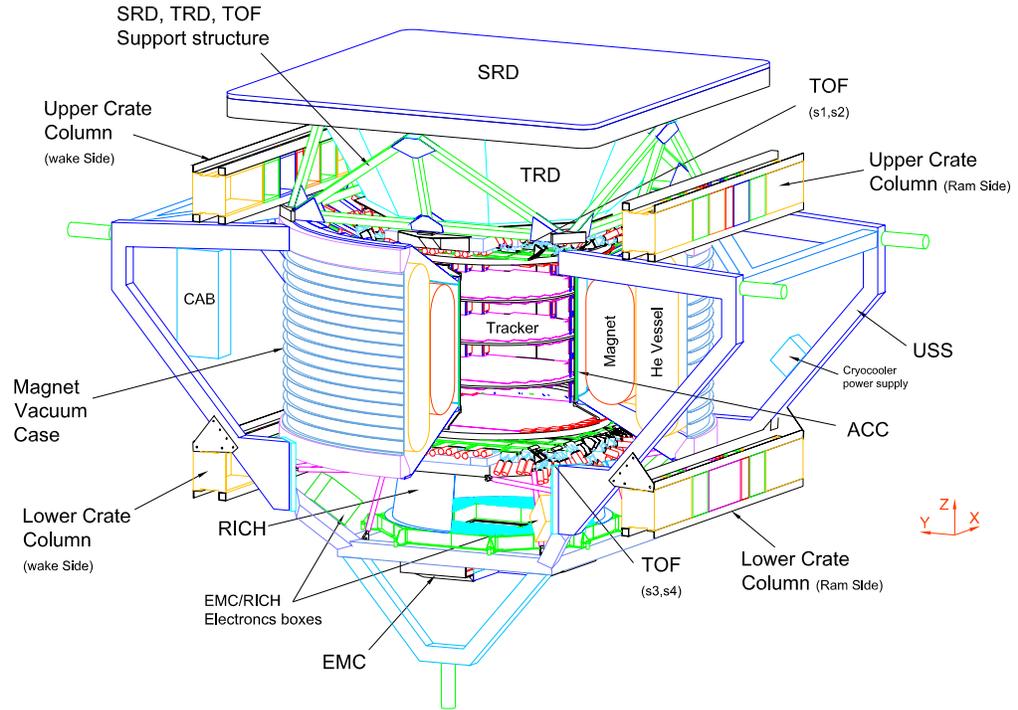


Figura 2.3: Struttura dell'apparato AMS-02.

2.3.1 Il Magnete Superconduttore

Il campo magnetico di AMS-02 sarà sei volte superiore a quello di AMS-01. Produrre un campo così intenso ($0.87T$) nello spazio non è affatto banale, in quanto un magnete permanente non è sufficiente.

La soluzione adottata per AMS-02 è di utilizzare un magnete a superconduttore (figura 2.4), mantenuto alla temperatura di lavoro di $1.8K$ per i tre anni della missione da un sistema di raffreddamento basato su una riserva di $2500l$ di elio superfluido.

Il magnete è formato da una serie di 14 anelli di superconduttore, dei quali 2 (di dimensioni maggiori che i restanti) sono utilizzati per generare l'intenso campo magnetico trasverso, mentre gli altri 12, oltre ad incrementare l'intensità del campo di dipolo, riducono il campo residuo fuori dal magnete e il momento torcente generato dall'interazione con il campo magnetico terrestre; il campo residuo stimato è di $15mT$ a una distanza di $2.3m$ da centro.

I magneti sono connessi in serie e lavoreranno ad una corrente nominale di $459A$. Essi sono costruiti con lo stesso tipo di conduttore: filamenti di $NbTi$ immersi in

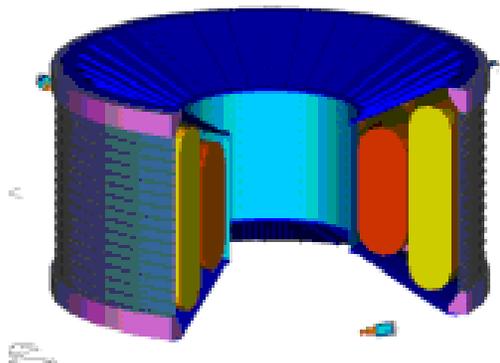


Figura 2.4: Vista del magnete. Diversamente da quello che la forma può far pensare, il campo dipolare è ortogonale all'asse del cilindro.

una matrice di rame altamente puro, il tutto impiantato all'interno di una barra di alluminio tramite un processo di coostrusione.

2.3.2 Il Tracker

Il Tracker (figura 2.5) si occuperà di tracciare le particelle all'interno del campo magnetico prodotto dal magnete superconduttore. Questo tipo di informazione consentirà di dedurre il momento della particella ed il segno della sua carica (si suppone infatti che il *verso* di percorrenza della traccia sia fornito dal TOF).

Per svolgere il suo compito, il Tracker sarà formato da 4 strati interni al magnete più due situati appena sopra e sotto di esso. Ogni strato sarà formato da diversi moduli composti da un numero variabile di sensori al silicio (il numero dipende dalla posizione della placca sul piano), dal cavo in kapton per la lettura delle *strips* e da due unità elettroniche per l'acquisizione seriale di entrambi i lati del modulo. La superficie attiva totale è di circa 7 m^2 , corrispondenti a 192 moduli suddivisi su 8 piani, di cui 6 interni al magnete e 2 posti ai suoi estremi.

I sensori nascono da *wafer* silicio di tipo n ($4 \times 7 \text{ cm}^2$), su cui vengono impiantate *strips* di tipo n^+ su un lato e di tipo p^+ sull'altro lato (in senso ortogonale alle precedenti). Una tensione positiva di $40 - 50 \text{ V}$ viene imposta alle *strips* di tipo n^+ rispetto a quelle di tipo p^+ , cosicché gli elettroni e le lacune prodotte dal passaggio di una particella ionizzante derivano verso le *strips*, e il segnale prodotto può venire letto (serialmente) dall'elettronica di *front end*.

Rispetto al Tracker di AMS-01 è stato sviluppato un nuovo sistema di raffreddamento che permette di estrarre il calore dissipato dall'apparato senza influire negativamente

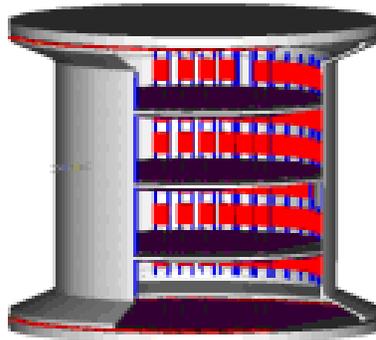


Figura 2.5: Vista del tracker al silicio.

te sulla temperatura del magnete superconduttore.

La risoluzione sarà pari a $30 \mu m$ nella direzione del campo magnetico e $10 \mu m$ in quella di curvatura, e dalla misura della carica depositata si otterrà il dE/dx con risoluzione del 6% nell'intervallo di rigidità $[3, 10] GV$. Il consumo in potenza sarà di circa $360 W$ per un numero totale di canali di 2×10^5 .

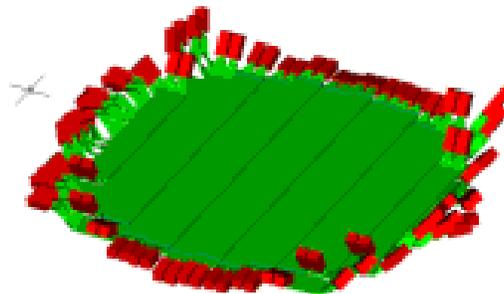


Figura 2.6: Rappresentazione grafica di un piano del TOF.

2.3.3 Il Time Of Flight

Il compito principale del TOF di AMS-02 sarà quello di fornire il *trigger* all'esperimento e di misurare il tempo di volo delle particelle. Esso sarà formato da quattro piani di materiale scintillante posti due sopra e due sotto al magnete. Ogni piano

(figura 2.6) è costituito da 12 contatori larghi 11 *cm* e spessi 1 *cm*, ed i piani adiacenti sono orientati in modo che i contatori siano ortogonali tra loro.

Il fototubi del TOF si troveranno a lavorare in condizioni più difficili che nel caso di AMS-01. Il magnete superconduttore, infatti, produce un campo più elevato del magnete permanente usato nella prima fase, e di conseguenza anche il campo residuo sui piani del TOF (e soprattutto sui suoi fototubi) risulta incrementato fino a $2 \div 3kG$.

Per ovviare almeno parzialmente il problema, la meccanica è stata ridisegnata e i fototubi *Hamamatsu* R5900 sono stati sostituiti dagli R5946. Questi ultimi sono del tipo *fine mesh*, configurazione che risulta meno sensibile di altre alla presenza di un campo magnetico al suo interno. Vista però l'intensità elevata del campo residuo è stato necessario inserire guide di luce alle estremità dei contatori che portassero i PM a lavorare ad angoli tra l'asse del tubo ed il campo minori di 40° .

In ogni caso, con questa configurazione non si arriverà ad ottenere le prestazioni che si avevano per AMS-01 ($\sigma_t \simeq 120ps$): la risoluzione attesa è di $150 \div 170ps$, che permetterà di separare elettroni da protoni fino ad un'energia di 1.5 *GeV*.

In aggiunta al tempo di volo, il TOF misurerà anche la carica rilasciata della particella, da cui si potrà ottenere una stima della perdita di energia.

Si parlerà più diffusamente del TOF nel paragrafo successivo.

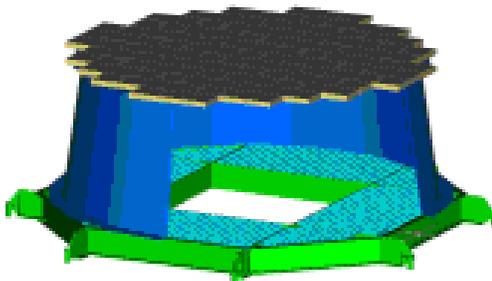


Figura 2.7: Visione grafica del RICH (*Ring Imaging CHerenkov*).

2.3.4 Il Ring Imaging Cherenkov

L'apparato dedicato alla rivelazione dei coni di luce Cherenkov (figura 2.7) è costituito da uno strato spesso 2 *cm* di *aerogel* (indice di rifrazione 1.05) che funge da radiatore, separato dal piano fotosensibile con un volume vuoto spesso 40 *cm* e circondato da specchi; in questo volume avviene l'espansione dei coni di luce.

L'acquisizione dei segnali luminosi è data in carico ad una serie di fototubi (gli stessi usati nel calorimetro) che consentono una granularità di lettura di $2 \times 2 \text{ cm}^2$.

Grazie al RICH sarà possibile identificare differenti isotopi fino al numero di massa $A = 25$ e distinguere tra diversi elementi chimici fino al ferro. La velocità sarà misurata con una risoluzione di $\Delta\beta/\beta \sim 2 \times 10^{-3}$ per impulsi nell'intervallo $[4, 12] \text{ GeV}/c$.

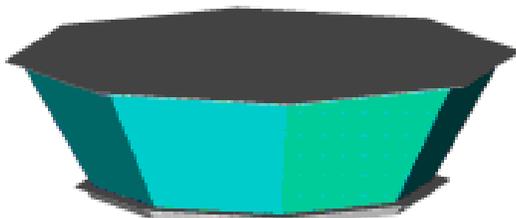


Figura 2.8: Vista del *Transition Radiation Detector* di AMS-02.

2.3.5 Il Transition Radiation Detector

Il TRD (figura 2.8) sarà addetto alla separazione e^-/\bar{p} e e^+/p fino a $3 \text{ TeV}/c$ di momento e contribuirà al tracciamento della traiettoria delle particelle. Esso è formato da 60 cm di materiale attivo suddivisi in 20 strati: ogni strato è composto da un foglio di materiale radiante (22 mm di fibra di polipropilene) e da una serie di tubi spessi 6 mm operanti in regime proporzionale riempiti con una miscela di Xe e CO_2 (5248 canali di lettura). I fili anodici (una lega di W e Au) hanno un diametro di $30 \mu\text{m}$.

L'emissione di radiazione di transizione avviene sulla superficie di separazione tra i fogli di radiatore ed il vuoto che separa questi ultimi dai tubi a gas.

Nell'intervallo $[10, 300] \text{ GeV}$ la separazione e^-/\bar{p} e e^+/p sarà al livello di 10^{-2} .

2.3.6 Il Calorimetro Elettromagnetico

Per consentire un'accurata misura del flusso di positroni, è fondamentale essere in grado di separarli dal fondo di protoni presente (per energie superiori a 10 GeV ci sono nei raggi cosmici circa $10^4 p$ per ogni e^+). Un modo per effettuare questa separazione è appunto l'utilizzo di un calorimetro elettromagnetico a campionamento (figura 2.9), che consente inoltre di misurare accuratamente l'energia di γ , e^+ ed e^-

fino a circa 300 GeV [29].

Il calorimetro di AMS-02 è composto da 9 piani, ognuno dei quali è formato da

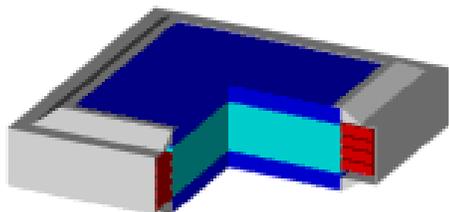


Figura 2.9: Il calorimetro elettromagnetico di AMS-02.

10 strati di piombo intervallati da fibre scintillanti; i piani adiacenti sono orientati in modo che le fibre risultino disposte ortogonalmente tra uno e l'altro. In questo modo si può studiare in maniera efficace lo sviluppo sia longitudinale che trasversale dello sciame.

La lettura dei segnali prodotti dal passaggio delle particelle nel materiale attivo è data in carico a 36 fototubi quadrati, ognuno dotato di 4 anodi indipendenti, che forniscono una granularità di lettura di $2 \times 2 cm^2$ ($2 cm \simeq 1$ lunghezza di radiazione).

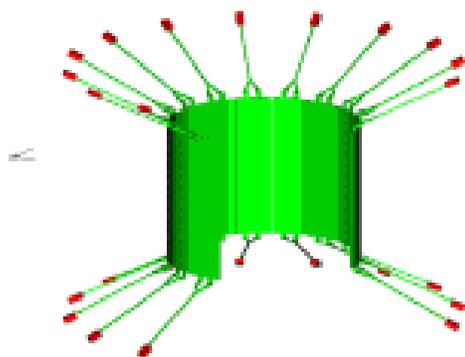


Figura 2.10: Il sistema di anticoincidenze di AMS-02.

2.3.7 Le Anticoincidenze

Il segnale prodotto dalle anticoincidenze è solitamente usato per vetare il *trigger*; corrisponde infatti ad eventi generati nell'interazione tra raggi cosmici e la materia di cui è composto lo spettrometro (principalmente il magnete e le strutture di sostegno). Gli 8 contatori di anticoincidenza sono formati da scintillatori spessi 1 *cm* piegati ad arco in modo da formare, affiancati, un cilindro che trova sede tra il tracker ed il magnete (figura 2.10). Ai due estremi di ogni scintillatore ci sono guide di luce a fibre ottiche che trasportano il segnale prodotto ad un fotomoltiplicatore; l'acquisizione dei segnali delle anticoincidenze è data in carico all'elettronica di acquisizione del TOF.

2.4 Il TOF di AMS-02

L'importanza del sistema TOF è superiore a quella degli altri *subdetectors* in quanto esso fornisce, oltre alle misure del tempo di volo e della perdita di energia, il *fast trigger* al sistema; ad esso è inoltre affidato il riconoscimento della direzione di volo delle particelle (discendenti o ascendenti).

I quattro piani di cui esso è costituito conterranno 12 contatori ognuno. I contatori situati ai lati di ogni piano avranno una forma diversa dagli altri, essendo più larghi al centro in modo da ricoprire tutta la superficie presapoco circolare che determina l'accettazione dello spettrometro.

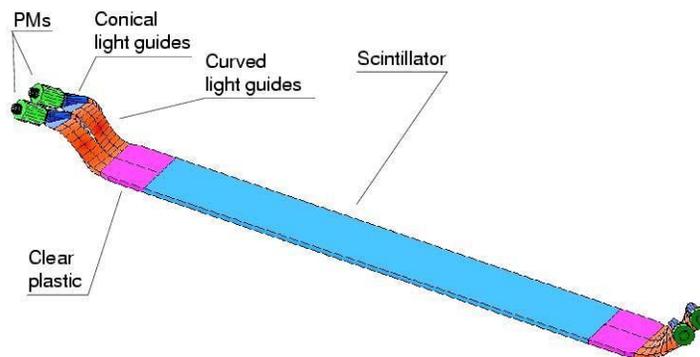


Figura 2.11: Contatore di AMS-02.

2.4.1 I contatori del TOF

L'elemento attivo del sistema TOF di AMS-02 è formato da una serie di contatori a scintillazione, ognuno dei quali è composto da una barra di scintillatore plastico terminata da due guide di luce che permettono la raccolta del segnale luminoso sui fototubi[27] (figura 2.11).

È stato scelto lo scintillatore plastico scintillante Bicron BC408, che viene sagomato in barre larghe 10 cm e spesse 1 cm, la cui lunghezza varia da 120 a 168 cm a seconda della posizione sul piano. Ai due estremi di ogni scintillatore è incollata una guida di luce in plexyglass che raccoglie la luce e la convoglia sul fotocatodo attraverso una pastiglia di silicone spessa 3 mm (Dow Corning 53-900).

A causa del campo magnetico residuo nelle posizioni in cui si troveranno ad operare i fototubi è stato necessario curvare alcune delle guide di luce in modo da minimizzare l'angolo formato tra l'asse del tubo e le linee di campo.

Su ogni lato del contatore ci sono due fototubi Hamamatsu R5946, di tipo *fine mesh*, che sono dotati di una catena di 16 dinodi a griglia molto vicini tra loro, il che li rende meno sensibili al campo magnetico residuo prodotto dal magnete superconduttore, stimato in $2 - 3kG$.

Il passaggio attraverso lo scintillatore di una particella carica si traduce nel deposito all'interno di esso di parte della sua energia cinetica sotto forma di luce. I fotoni così prodotti vengono convogliati (sia direttamente che per riflessione sulle pareti della barra) alle due estremità, dove, tramite l'accoppiamento ottico creato con la guida di luce e poi con il fototubo, incidono sul fotocatodo generando alcuni fotoelettroni. Questi ultimi subiscono l'effetto del campo elettrico applicato tra catodo e primo dinodo venendo accelerati; nell'urto con il dinodo avviene l'estrazione di altri elettroni, che vengono a loro volta accelerati verso il dinodo successivo. Il processo si ripete per il numero dei dinodi, finché il segnale arriva, notevolmente amplificato dal processo di valanga, all'anodo.

A questo livello il segnale prodotto dagli anodi dei due PM sullo stesso lato del contatore viene sommato e dopodiché arriva al sistema di acquisizione dove viene formato, processato, convertito ed, eventualmente, acquisito.

Bisogna tener conto di diversi effetti che deteriorano la qualità del segnale; i più importanti sono:

- attenuazione della luce nello scintillatore;
- perdita di luce nei contatti ottici e nelle guide di luce;
- bassa efficienza quantica dei fototubi (dell'ordine del 10 – 20%);
- presenza del campo magnetico all'interno dei fototubi.

Si può descrivere macroscopicamente l'effetto di assorbimento della luce nello scintillatore introducendo la quantità *lunghezza di attenuazione* λ :

$$I(x) = I_0 e^{x/\lambda} \tag{2.13}$$

Essendo $I(x)$ l'intensità della luce a distanza x dal punto in cui è stata generata, λ rappresenta la distanza alla quale l'intensità si riduce di un fattore e .

In generale $I(x)$ non dovrebbe dipendere dalla forma dello scintillatore ma soltanto dal materiale di cui esso è costituito; in realtà però di solito λ diminuisce all'aumentare della lunghezza di una barra scintillante. Questo fenomeno si spiega osservando che nel punto di produzione la luce viene emessa isotropicamente in tutte le direzioni, e quindi il segnale misurato lo si può pensare formato da due componenti:

- luce diretta, che non subisce riflessioni durante il tragitto;
- luce riflessa, che raggiunge una delle estremità della barra dopo aver subito ripetute riflessioni sulle sue pareti.

A parità di distanza x a cui si osserva il segnale, quest'ultima componente risulta maggiormente attenuata a causa del maggior percorso effettuato (a zig-zag piuttosto che rettilineo) e delle ripetute riflessioni che ha subito; come conseguenza si ha che il rapporto tra le intensità delle due componenti dipende dalla distanza x a cui il segnale viene prelevato, e ne risulta pertanto influenzata anche la lunghezza di attenuazione λ .

Ogni volta che il segnale luminoso attraversa la superficie che separa due mezzi con indice di rifrazione diverso, viene attenuato a causa della riflessione all'indietro che subisce parte del segnale e dalla modifica della direzione di propagazione nel nuovo mezzo.

L'efficienza quantica rappresenta l'efficienza di conversione dei fotoni incidenti sul fotocatodo in elettroni.

Il problema più gravoso a cui è necessario far fronte è sicuramente dovuto alla presenza del campo magnetico residuo prodotto dal magnete superconduttore. I fotoelettroni prodotti si trovano a percorrere traiettorie curve che, nel caso peggiore, li portano ad incidere sulle pareti del tubo piuttosto che sul dinodo. I fototubi R5946 scelti per AMS-02 sono meno sensibili a questo tipo di effetto essendo dotati, come si è detto, di dinodi a griglia molto vicini tra loro; l'ampia superficie di questi ultimi rende la moltiplicazione del segnale meno sensibile alla traiettoria degli elettroni e quindi all'effetto di curvatura prodotto dal campo magnetico in cui è immerso.

Lato negativo di questi fototubi è risultata la variabilità dei parametri (ad esempio il guadagno, che può variare di un ordine di grandezza), e la sensibilità a sollecitazioni meccaniche, causata probabilmente dall'esilità della struttura interna.

Per concludere, il parametro fondamentale che si richiede ai contatori del TOF è una buona risoluzione temporale ($150ps$), che viene notevolmente influenzata da tutte le componenti facenti parte della catena illustrata. Inoltre, il fatto che i contatori dovranno lavorare per un lungo periodo senza possibilità di intervento diretto impone una filosofia di costruzione decisamente rindondante; per questo ognuno di essi è dotato di due fototubi per lato alimentati da canali di alta tensione separati.

2.4.2 Trigger

Il sistema di *trigger* si basa sulla presenza nell'elettronica di acquisizione del TOF di tre discriminatori paralleli con soglie diverse. La soglia *low* (di circa $30 mV$) è usata per la misura di tempo, dato che il degrado introdotto dal *time walk* diminuisce abbassando la soglia di discriminazione (si vedano in proposito i capitoli 3 e 4).

La soglia intermedia (chiamata *high*), dell'ordine di $150 mV$, corrisponde al passaggio di una MIP (*Minimum Ionizing Particle*), ed il segnale che supera questa soglia viene usato per la produzione del *fast trigger*.

Esiste una terza soglia proporzionale alla precedente (*super high*) che corrisponde indicativamente al passaggio di una particella il cui numero atomico è maggiore di 2. Questa soglia è utilizzata dal trigger per mascherare il segnale delle anticoincidenze. Il sistema di *trigger* è organizzato su tre livelli di cui il *fast trigger* è chiaramente il primo:

1. *fast trigger*: corrisponde alla coincidenza dei segnali di almeno tre dei quattro piani che compongono il TOF; il tempo di produzione è dell'ordine di $150 - 200 ns$;
2. *second level trigger*: agisce ancora localmente, a livello del *front end* dei singoli *subdetector*. Questo segnale deve pervenire entro un *timeout* di circa $6 \mu s$, e corrisponde alla presenza combinata di:
 - *fast trigger*;
 - *matrice di coincidenza*, programmabile a seconda della logica di trigger desiderata;
 - assenza del segnale delle anticoincidenze.
3. *third level trigger*: viene prodotto una volta raccolti i dati dai vari *subdetectors*, quando un processore dedicato effettua una prima analisi dei dati scartando gli eventi che non soddisfano i seguenti requisiti:
 - presenza di segnale su *entrambi* i lati dei contatori del TOF colpiti;

- limite inferiore sul momento valutato in base alla distanza dei punti identificati dal tracker rispetto all'ipotesi di traiettoria rettilinea;
- passaggio della traiettoria del tracker attraverso i contatori del TOF colpiti.

Il rate atteso durante il periodo di presa dati è di circa 2 kHz , che crescerà nel corso dei tre anni della missione per via dell'avvicinamento alla fase di minimo dell'attività solare.

Capitolo 3

L'hardware del telescopio per raggi cosmici

Il lavoro da me svolto nell'ambito dell'esperimento AMS ha riguardato prevalentemente il sistema utilizzato per la caratterizzazione dei contatori costituenti il TOF. È infatti fondamentale conoscere precisamente i loro parametri di funzionamento, sia relativi alla tempistica che all'ampiezza d'impulso.

In particolare sono stati valutati i seguenti parametri:

- lunghezza di attenuazione;
- fotostatistica;
- correzioni di *time walk*;
- velocità di propagazione della luce nel contatore;
- risoluzione sul tempo di volo;
- risoluzione intrinseca.

Lunghezza di attenuazione e fotostatistica determinano l'ampiezza dei segnali prodotti dal contatore; il primo è determinato sostanzialmente dalla qualità dello scintillatore, mentre il secondo dipende dall'efficienza quantica dei fototubi.

Le correzioni per l'effetto chiamato *time walk* tengono conto del tempo di salita del segnale prodotto dai fototubi, e la velocità della luce all'interno del contatore dipende chiaramente dal materiale di cui lo scintillatore è costituito. Per una valutazione corretta della risoluzione è necessario tenere conto dell'effetto introdotto sulle misure da tutti questi parametri.

Per studiare il comportamento dei contatori si è installato a Bologna un apparato

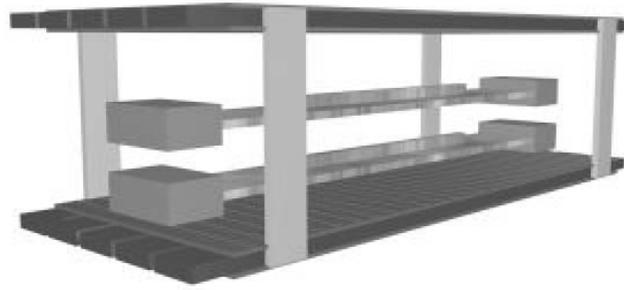


Figura 3.1: Rappresentazione grafica del telescopio per raggi cosmici utilizzato per il test dei contatori nel laboratorio di Bologna.

che consente il tracciamento dei raggi cosmici secondari (prodotti nell'atmosfera) che giungono al suolo, in modo da poter osservare la risposta del sistema sotto esame al passaggio di queste particelle ed essere in grado di fare previsioni corrette su quello che sarà il comportamento dei contatori durante i tre anni di missione sulla ISS.

3.1 Il telescopio per il test dei contatori coi raggi cosmici

Il telescopio (figura 3.1) è formato da due piani distanziati tra loro 30 cm , ognuno dei quali è composto da 5 tubi a streamer limitato (*Tubi di Iarocci*) larghi 9 cm e lunghi 140 cm . Ortogonalmente ai tubi è sistemata una serie 72 di *strips* larghe 1.8 cm , che consentono di individuare la posizione di passaggio delle particelle lungo l'asse del piano con una risoluzione sufficiente[30]. Tra i due piani vengono posti un contatore usato come riferimento (fornisce il t_0 al sistema) e i contatori da verificare. Nota la geometria dell'apparato e misurando i punti di passaggio di una particella sui due piano LST, è immediato ricavare il punto di impatto della particella stessa sui contatori.

Con questo sistema è possibile acquisire in poco tempo tutti i dati necessari alla caratterizzazione dei contatori.

3.1.1 I tubi LST

I tubi a streamer limitato (LST, Limited Streams Tubes) sono contatori a gas che lavorano nella zona intermedia tra il regime proporzionale e quello a scarica (Geiger); il segnale prodotto, infatti, è saturato (ovvero non mantiene memoria dell'energia rilasciata dalla particella ionizzante), ma il regime di scarica viene controllato uti-

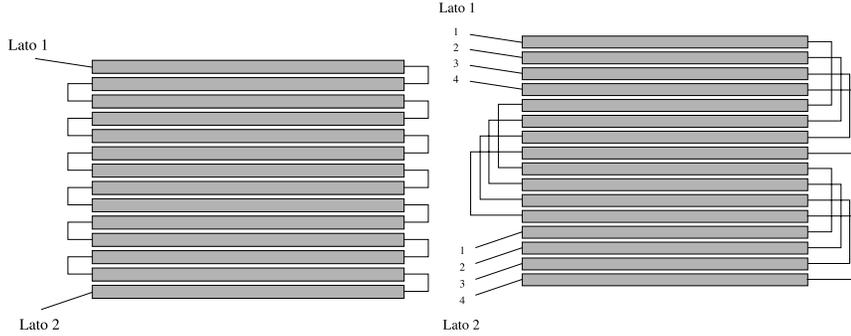


Figura 3.2: Connessione seriale per la lettura delle *strips* con una o quattro catene per piano. Quest'ultima è la configurazione scelta per il telescopio.

lizzando nella miscela di gas una componente addetta al *quenching*. Nel nostro caso sono isobutano e CO_2 a svolgere questo compito.

I tubi LST usati per il telescopio sono composti da 8 cellette di sezione $1 \times 1\text{cm}^2$, e ognuno di essi è dotato di un filo anodico del diametro di $100\mu\text{m}$ ricoperto in oro, mantenuto in posizione centrale da alcuni distanziatori plastici distribuiti lungo tutta la lunghezza. Le pareti (in plastica) sono ricoperte da uno strato di grafite di resistività minima pari a $50\text{k}\Omega/\text{m}^2$.

In corrispondenza del passaggio di una particella ionizzante, gli elettroni liberati dall'interazione di questa con le molecole del gas derivano verso l'anodo. Giunti in sua prossimità, l'aumento del campo elettrico (il cui andamento radiale va come $1/r$) fornisce sufficiente energia agli elettroni perchè questi provochino ionizzazione secondaria negli urti con le molecole del gas: il processo a valanga genera una scarica che viene estinta grazie ai gradi di libertà forniti dal *quencher* al sistema, i quali contribuiscono alla conversione dell'energia delle particelle ionizzanti in calore.

3.1.2 Tracciamento delle particelle: le strips

Il moto di queste cariche genera per induzione un segnale elettrico di polarità inversa sulla *strip* più vicina al punto d'impatto della particella sul piano; è questo segnale che fornisce l'informazione spaziale cercata.

Per acquisire il segnale le *strips* vengono lette serialmente (figura 3.2), e sfruttando i diversi ritardi di arrivo agli estremi della catena è possibile risalire alla *strip* colpita. Numerando le *strips* della catena da 1 a N (dove $N = 72$), si supponga che il passaggio di una particella generi un segnale sull' n esima. Sia δt il tempo che necessario al segnale ad attraversare una *strip*. Allora il segnale arriverà ad un estremo dopo un tempo $t_{sx} = n\delta t$, ed all'altro estremo dopo $t_{dx} = (N - n)\delta t$. La differenza dei

tempi di arrivo sarà per cui $t_{sx} - t_{dx} = (2n - N)\delta t$, proporzionale quindi ad n . Chiaramente non ci si aspetta un segnale estremamente pulito perchè non si hanno soltanto particelle incidenti al centro delle strips, ma su buona parte della loro larghezza. È per questo motivo che si è scelto di avere *strips* più lunghe della zona effettivamente coperta dai tubi LST, in modo da isolare sufficientemente i segnali provenienti da *strips* diverse.

Con una configurazione di questo tipo basterebbero soltanto quattro canali di TDC per l'acquisizione dei tempi di arrivo ai due estremi delle catene (due catene, una per piano), ma in realtà la configurazione può ancora essere migliorata; infatti un generico segnale prodotto da una MIP viene indotto solitamente non su una singola *strip*, ma, con intensità diverse, anche sulle adiacenti. Spesso la molteplicità risulta essere di 2 o 3.

Questo inconveniente peggiora l'affidabilità con cui è possibile identificare la *strip* colpita, ma una soluzione non troppo dispendiosa in termini di canali TDC è di creare non una ma più catene indipendenti su ogni piano.

Così facendo, si identificheranno le *strips* in cui viene generato il segnale e, data la molteplicità, sarà possibile interpolare la posizione di passaggio della particella. La figura 3.2 illustra la soluzione scelta per il telescopio, ovvero con quattro catene indipendenti per piano.

Adottando questo tipo di soluzione sono necessari in definitiva soltanto 16 canali TDC per il tracciamento della particella lungo l'asse dei tubi. Vista la debolezza dei segnali indotti sulle *strips* si è reso inoltre necessario l'inserimento nella catena di acquisizione di 16 amplificatori lineari ($\times 10$).

3.1.3 Il punto di lavoro dei tubi

Il comportamento dei tubi LST dipende fortemente dalla miscela di gas in esso contenuta. In particolare la quantità di *quencher* è un fattore critico perchè pilota la formazione dello *streamer*. La miscela usata nei piani del telescopio è composta da isobutano (10%), CO_2 (80%) e argon (10%). Questa miscela non è infiammabile (requisito fondamentale vista la mancanza dei sistemi di sicurezza necessari all'utilizzo di miscele infiammabili in laboratorio); essa contiene due componenti di *quencher* (isobutano e CO_2) e l'argon che fa sviluppare lo *streamer*.

Per fissare le tensioni di lavoro dei tubi LST, una prima valutazione è stata fatta in base all'andamento dei conteggi dei fili anodici in funzione dell'alta tensione, dopodiché è stato fatto uno *scanning* in tensione dei conteggi forniti dalle *strips* per rifinirne il valore. È infatti il loro segnale che permette di ricostruire la traccia, e risulta pertanto più importante che i segnali forniti dai fili anodici stessi. Le tensioni di lavoro sono state così corrette e fissate al valore di 4800V osservando i *plateau* ottenuti (vedi fig. 3.3).

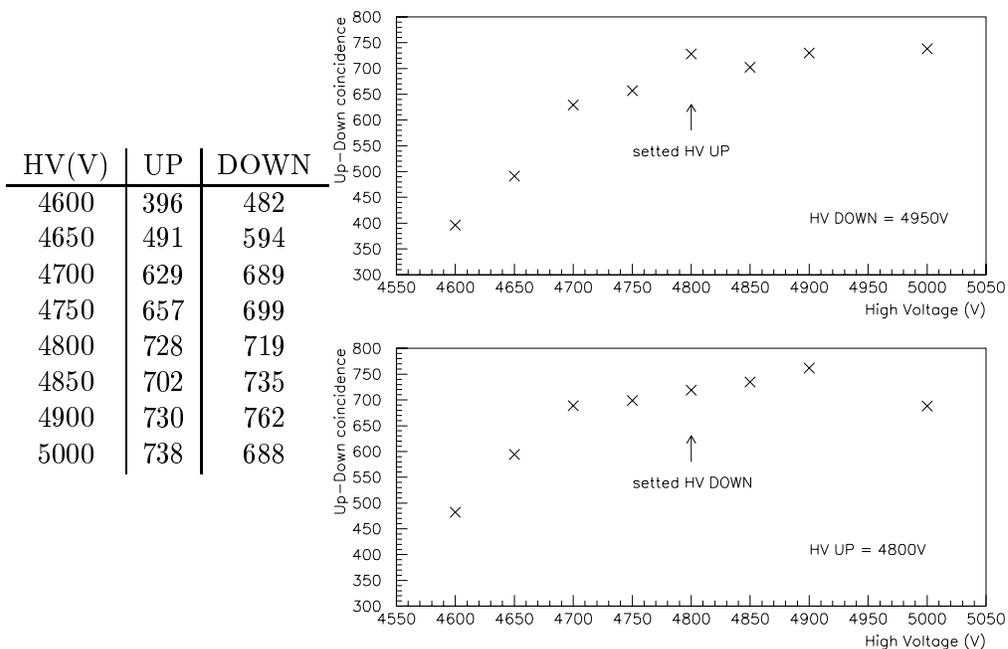


Figura 3.3: *Plateau* in tensione dei piani di *strips* *UP* (sopra) e *DOWN* (sotto). In entrambi i casi la tensione è fissata a 4800V.

Un procedimento analogo è stato seguito per fissare le tensioni di soglia dei discriminatori (i segnali delle *strips* vengono discriminati dopo essere stati amplificati di un fattore 10): si valuta in questo caso l'andamento delle coincidenze tra piani UP e DOWN, in quanto si richiede che almeno una *strip* per piano dia un segnale per poter ricostruire la traccia. Inizialmente la soglia è stata fissata in comune per tutti i 16 canali; inoltre si è deciso di mantenerla in ogni caso uguale per i due estremi della stessa *strip*. In figura 3.4 si osserva il plateau ottenuto.

Una volta fissato il valore iniziale a $30mV$ si è aggiustata l'equalizzazione tra le *strips* valutando i conteggi delle singole catene. I valori definitivi sono riportati in tabella 3.5.

3.2 L'elettronica per l'acquisizione dati

Per l'acquisizione dei dati forniti dal telescopio è necessaria un'interfaccia elettronica che adatti i segnali analogici prodotti dai rivelatori alla lettura da parte del PC.

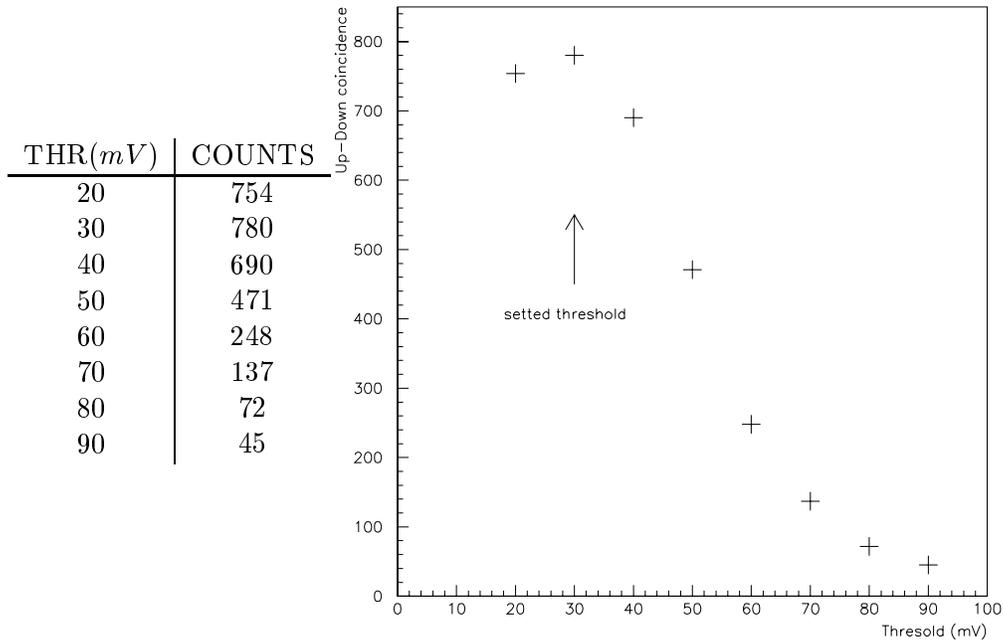


Figura 3.4: Piedistalli nelle soglie dei piani di *strips*. Il valore iniziale fissato a $30mV$ è stato in seguito rifinito *strip* per *strip* per migliorarne l'equalizzazione.

	Strip 1	Strip 2	Strip 3	Strip 4
UP	30	30	20	25
DOWN	20	30	25	30

Figura 3.5: soglie definitive delle 8 catene di *strips* (in mV).

Sul mercato è disponibile una moltitudine di schede elettroniche progettate per i più svariati compiti: ADC, TDC, discriminatori, Time Units, sono soltanto alcuni esempi. Questo tipo di schede necessita l'interfacciamento con un *bus*, dispositivo a cui è affidato il compito di alimentare, organizzare, e in alcuni casi interfacciare con altri *buses* le schede elettroniche necessarie al DAQ.

Diversi standard si sono alternati nel corso degli anni sia nel campo industriale che nella ricerca; ampia diffusione hanno avuto il NIM, il CAMAC e, attualmente, il VME.

Il bus NIM è il più semplice dei tre, in quanto si occupa solamente di fornire le tensioni di alimentazione ai moduli senza gestire la comunicazione dati; risulta comunque di ampia utilità tuttora, soprattutto quando sono necessari moduli di *formazione* di segnali (discriminatori, amplificatori, fanin-fanout, etc.).

Il Bus CAMAC, viceversa, si occupa fundamentalmente della trasmissione dati tra le schede in esso alloggiate e sistemi esterni. Negli anni passati la maggior parte degli esperimenti di fisica delle particelle (e non solo) si è basata su di esso. Lo sviluppo tecnologico lo ha ormai reso obsoleto, ma la grande diffusione che ha avuto rende tuttora disponibile una quantità notevole di elettronica basata su di esso. Inoltre l'affidabilità dei sistemi fondati su CAMAC è confermata da anni di intenso utilizzo. Ecco perchè ancora oggi risulta abbastanza diffuso.

Il nuovo standard che si sta affermando come sostituto del CAMAC è il VME. Questo tipo di Bus fornisce prestazioni notevoli ed una maggiore versatilità, a fronte di un incremento della complessità di gestione.

In fase di preparazione di AMS-01 tutti i sistemi di acquisizione usati per la realizzazione del TOF erano basati sul Bus CAMAC; si è deciso in seguito, visto l'inevitabile invecchiamento dell'elettronica disponibile, di cominciare per AMS-02 la migrazione a sistemi basati su VME.

Parte del lavoro sperimentale da me svolto ha riguardato proprio la scrittura del *software* di acquisizione per questo tipo di sistemi, parallelamente alla realizzazione del sistema di acquisizione vero e proprio.

Si è deciso di effettuare il salto tra i due standard in questo modo: durante una prima fase si è approntato un DAQ basato su CAMAC, ed una prima analisi dei dati ha fornito una indicazione delle caratteristiche del contatore di riferimento e del prototipo dei nuovi contatori per avere una base d'appoggio sulla valutazione dei dati forniti nella seconda fase. In questa seconda fase si è realizzato il sistema finale basato su VME, confrontando i risultati ottenuti con quelli della fase precedente.

3.3 Fase 1: CAMAC

Fissato il punto di lavoro dei piani LST si è passati a realizzare la prima versione del DAQ. Si può suddividere la struttura del sistema in diversi blocchi funzionali, ognuno dei quali svolge un compito diverso (fig. 3.6).

Il primo blocco che verrà analizzato è quello che fornisce la temporizzazione e sincronizzazione del sistema: il *trigger*. Osservando la figura 3.7 si nota che già in un sistema abbastanza semplice come il telescopio si sono resi necessari due livelli di *trigger* hardware. Il motivo è legato sostanzialmente alle due condizioni seguenti imposte al sistema di acquisizione:

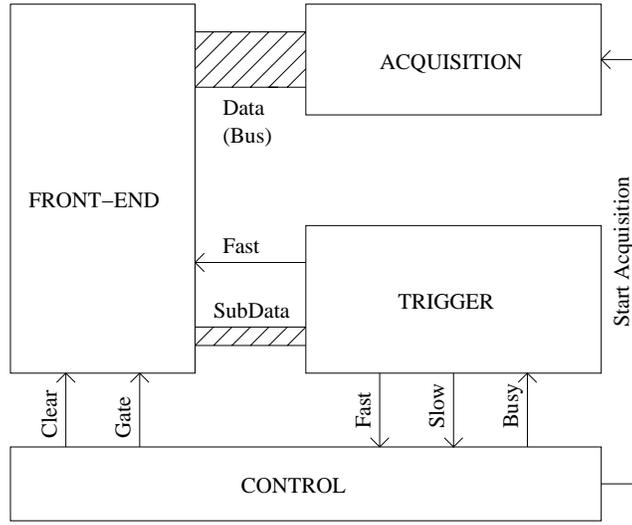


Figura 3.6: Schema a blocchi del sistema di acquisizione.

- un evento è considerato buono soltanto se è possibile ricostruirne la traccia, e quindi se c'è segnale su almeno una catena di *strip* per piano.
- per ottenere una buona risoluzione sui tempi misurati dai TDC è necessario che tutti i segnali che esso riceve siano veloci, ben formati e stabili.

La prima condizione richiederebbe che il *trigger* (che sarà utilizzato per dare lo start comune ai TDC) tenga conto anche dei segnali delle *strips*; ma i segnali delle *strips* sono molto lenti, sia a causa della lentezza intrinseca dei rivelatori a gas (dovuta al tempo di deriva degli elettroni verso l'anodo) che del lungo percorso che i segnali devono attraversare lungo la catena prima di giungere all'elettronica. Il segnale di *trigger* dovrebbe quindi essere ritardato un tempo sufficiente ad attendere il loro arrivo, ed inoltre la sua tempistica dipenderebbe fortemente dalla posizione di incidenza delle particelle.

In questa situazione si sarebbe quindi in contrasto con la condizione 2, essendo il *trigger* il segnale che gestisce la temporizzazione dei TDC.

La soluzione adottata è quella di utilizzare due livelli hardware di *trigger*. Il primo (veloce e stabile, prodotto soltanto dal contatore di riferimento) è utilizzato per pilotare i TDC: entrambi i lati del contatore di riferimento vengono discriminati ad una soglia di $30mV$ e posti in coincidenza formando il *Fast Trigger*. Uno di essi (quello sul lato sinistro del telescopio) è ritardato di $16ns$ in modo da renderlo sempre l'ultimo a entrare nella coincidenza. Con tale accorgimento tutti i tempi

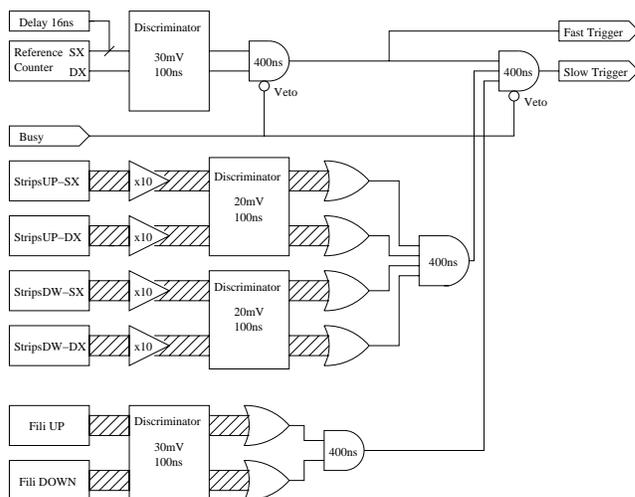


Figura 3.7: Schema funzionale del *trigger*.

acquisiti sono sempre riferiti allo stesso canale.

La digitalizzazione dell'evento inizia in corrispondenza della formazione del *Fast Trigger*. Dopodichè, il *trigger* di secondo livello (*Slow Trigger*) informa il sistema sulla qualità dell'evento che si sta acquisendo. Nel caso sia buono l'acquisizione procede normalmente, viceversa l'evento viene rigettato e il sistema si prepara ad una nuova acquisizione.

La bontà dell'evento dipende dalla risposta dei piani di *strips*: se almeno una *strip* per piano ha prodotto un segnale allora sarà possibile ricostruire la traccia della particella. In tal caso l'evento va acquisito, viceversa va scartato.

Per produrre questo secondo livello di *trigger* i segnali delle *strips* (amplificati e discriminati) vengono posti in una porta OR per ogni piano. Dopodichè è sufficiente mettere in coincidenza le uscite di queste due porte logiche per ottenere il segnale desiderato.

Il segnale di *busy* veta il *trigger* (fast e slow) quando l'elettronica è occupata in una conversione, consentendole di finire il processo senza interruzioni.

Il *front-end* effettua la digitalizzazione vera e propria dei segnali. Il suo schema è quello di figura 3.8.

Le informazioni da acquisire sono di due tipi:

- tempi di arrivo dei segnali rispetto ad uno usato come riferimento;
- integrale temporale del segnale, proporzionale alla carica raccolta dal detector.

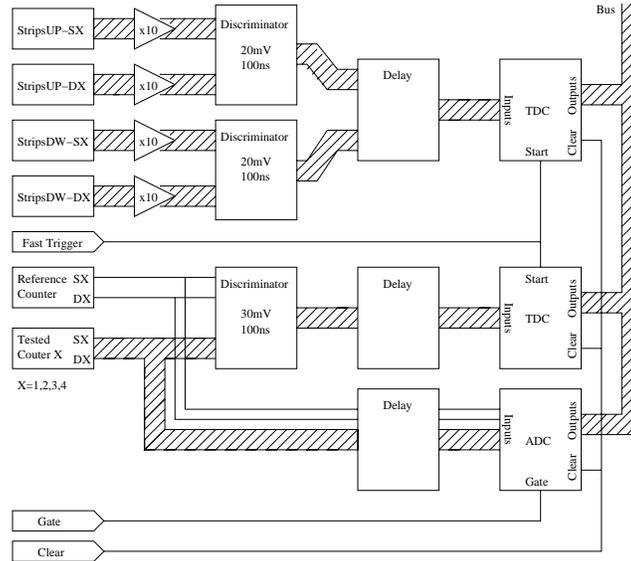


Figura 3.8: Schema funzionale del blocco di acquisizione.

Per le misure di tempo si utilizzano moduli TDC (*Time to Digital Converter*) che necessitano di un segnale comune utilizzato come punto di partenza (START) ed un certo numero di segnali che corrispondono alla fine degli intervalli temporali da misurare. I valori forniti dai TDC sono numeri binari proporzionali alle differenze di tempo tra i segnali di STOP e lo START.

Per la misura della carica si ricorre a moduli ADC (*Analog to Digital Converter*), che vengono pilotati da un segnale di *gate*. Il *gate* si trova solitamente a livello logico 0; quando esso commuta ad 1 inizia l'acquisizione della carica presente sui canali di ingresso, che termina soltanto quando il *gate* torna al suo stato iniziale. Schematicamente si può pensare che il segnale venga terminato su un condensatore, il quale si carica inducendo ai propri capi una tensione proporzionale alla carica immagazzinata ($V = CQ$). Il segnale acquisito viene poi convertito in un segnale digitale proporzionale alla sua ampiezza, e quindi alla carica raccolta.

Dallo schema si nota che il *fast trigger* viene utilizzato come start comune dei TDC, ed i segnali digitalizzati dai discriminatori vengono ritardati in modo tale da sfruttare al meglio la scala temporale disponibile sui moduli. Un discorso simile vale per gli ADC: vengono pilotati dal *gate* comune (sufficientemente ampio da tener conto delle fluttuazioni temporali dei segnali), e i segnali analogici sono ritardati in modo da temporizzare correttamente l'acquisizione.

I segnali prodotti dai moduli di conversione vengono memorizzati in *buffer* tempora-

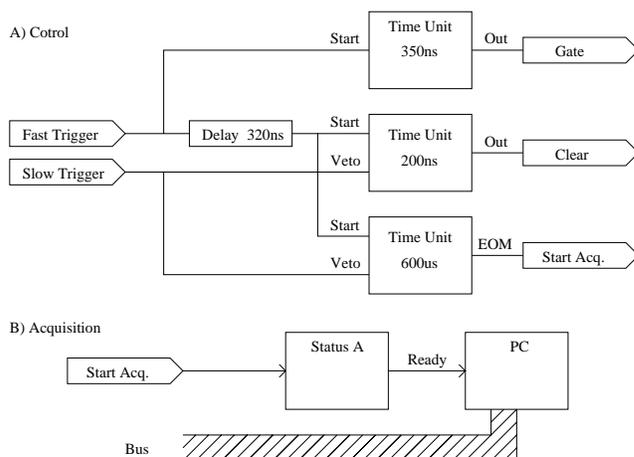


Figura 3.9: Schema funzionale dei blocchi di controllo (a) e acquisizione (b).

nei in attesa che il sistema di acquisizione li recuperi per lo stoccaggio nella memoria di massa.

Il controllo e la temporizzazione dei diversi blocchi funzionali necessitano la generazione di alcuni segnali dedicati: *Gate*, *Clear*, e *Start Acquisition*.

Lo schema che si occupa della loro gestione è quello riportato in figura 3.9. Il *gate* è formato utilizzando una *time unit* pilotata dal *fast trigger*; quindi gli ADC iniziano l'acquisizione ogni volta che una particella incide sul contatore di riferimento.

Se la traccia non può essere ricostruita perchè non si ha segnale sui piani di *strips* l'evento va eliminato; di questo se ne occupa il *clear*, anch'esso generato da una *time unit* sempre pilotata dal *fast trigger*. Nel suo caso però lo start è ritardato di $320ns$, attesa necessaria a consentire ai segnali delle *strips* di raggiungere l'elettronica e di formarsi adeguatamente.

Nel caso in cui questi segnali arrivino, parte lo *slow trigger* che veta la *time unit* del *clear* che così non viene generato. Viceversa, esso pulisce tutti i moduli di *front end* e riporta il DAQ allo stato iniziale.

In modo analogo viene generato il segnale *Start Acquisition*: il suo *End Of Marker* avvisa lo Status A (con un ritardo sufficiente a terminare la digitalizzazione) che un evento buono è pronto per essere acquisito.

Il PC continua periodicamente ad interrogare lo status A (*polling*). Quando l'evento è pronto lo status A dà il via libera al PC che procede all'acquisizione dei dati ed al loro stoccaggio.

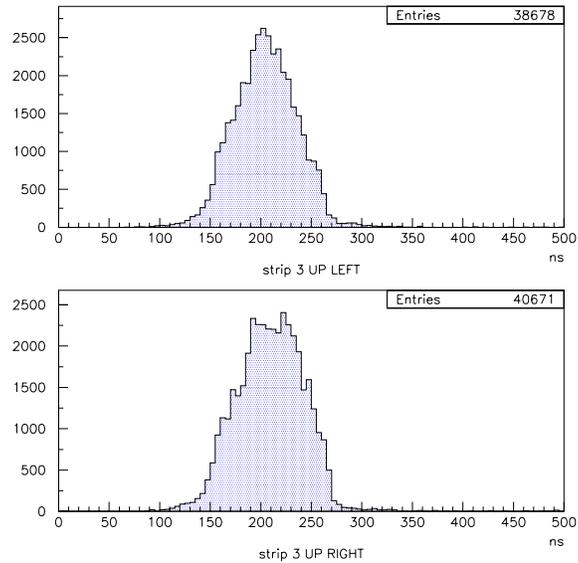


Figura 3.10: Spettri di TDC dei due lati (sinistro e destro) della catena di *strips* numero tre del piano superiore.

3.3.1 Ricostruzione dell'evento

Durante l'assemblaggio del sistema di acquisizione è cominciata gradualmente la fase di acquisizione dati con il primo prototipo dei contatori per verificare passo dopo passo la qualità del lavoro svolto. Come riferimento è stato usato uno dei contatori del TOF di AMS-01, costituito da una barra di scintillatore plastico Bicron BC408 spessa 1 *cm*, larga 11 *cm* e lunga 136 *cm* accoppiata a tre fototubi Hamamatsu R5900U su ogni lato tramite le apposite guide di luce e le pastiglie di silicone (Dow Corning 93-500).

Ricostruzione della traccia

Per poter utilizzare i dati acquisiti è necessario prima di tutto ricostruire la traccia percorsa dalla particella. Nel paragrafo 3.1.2 si è visto come le *strips* vengano lette serialmente per l'identificazione della posizione di incidenza della particella; in figura 3.10 è possibile osservare due spettri di tempo tipici forniti dai due lati di una delle catene di *strips*.

Gli spettri in sè non forniscono molte informazioni, mentre è la semidifferenza tra

i due lati ad essere importante: essa dipende, infatti, dalla *strip* della catena che è stata colpita.

Essendo la lunghezza delle *strips* maggiore della dimensione trasversale del piano (o meglio, della sua parte attiva), soltanto la parte centrale risponderà con un segnale elettrico al passaggio di una particella. Il tragitto percorso dal segnale stesso per raggiungere l'elettronica è invece comprensivo anche delle parti più estreme delle *strips*. Conseguenza fondamentale di questo accorgimento è una sorta di quantizzazione negli spettri, in cui l'intervallo di tempo tra i segnali provenienti da due *strips* adiacenti sono separati all'incirca dal tempo necessario al segnale ad attraversare un *strip*.

Questo tipo di quantizzazione non è apprezzabile negli spettri singoli, ma diventa evidente sulle differenze di tempo tra i due estremi. In questo caso sono presenti dei picchi notevolmente pronunciati in corrispondenza di particelle passate centralmente, separati da "zone morte" relative alle parti periferiche (non attive) dei piani. Un esempio di questo tipo di spettro è riportato in figura 3.11. Si notano chiaramente i 18 picchi che corrispondono alle 18 *strips* che compongono la catena; nell'istogramma giallo sono riportati tutti gli eventi senza selezioni particolari (sono soltanto stati eliminati gli *overflow*), mentre in blu si hanno i *single shoot events*, ovvero soltanto gli eventi in cui soltanto una catena per piano ha prodotto un segnale. In quest'ultimo plot al prezzo di una riduzione di statistica al 27.8% otteniamo una migliore separazione tra i picchi che consente di stabilire in modo più accurato i parametri utilizzati per ricostruire la traccia.

Preso quindi un *run* di dati così selezionato usato poi come riferimento, ognuno dei 18 picchi delle 8 catene di *strips* è stato fittato con una funzione gaussiana in modo da determinarne il valore medio e la deviazione standard (vedi fig. 3.12). Noti questi parametri è possibile assegnare ogni evento alla *strip* "più probabile", ovvero quella che ha la probabilità maggiore di essere quella colpita dalla particella. Questo metodo è noto come *metodo di Maximum Likelihood*.

Sia ΔT_i la semidifferenza tra i tempi di arrivo ai due lati della catena *i*-esima

$$\Delta T_i = \frac{t_i(sx) - t_i(dx)}{2}$$

e con μ_{ij} e σ_{ij} si indichino la media e la deviazione standard della gaussiana ottenuta dal fit del picco *j* della catena *i*. La probabilità per un evento che ha fornito un certo ΔT_i di essere passato per la *strip j*-esima è data da:

$$P_i(j) = \frac{1}{\sigma_{ij}\sqrt{2\pi}} e^{-\frac{(\Delta T_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

dunque calcolando per ogni ΔT_j il corrispondente valore $P_i(j)$, la *strip* che più probabilmente è stata colpita è quella che corrisponde al suo massimo. In realtà

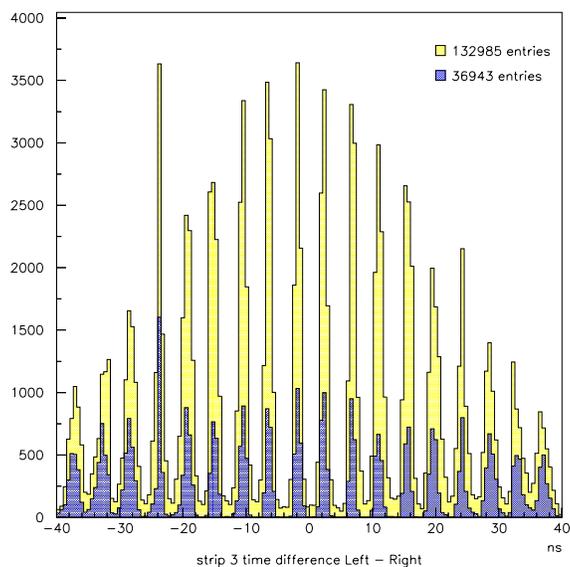


Figura 3.11: Semidifferenze di tempo sinistro - destro relative alla catena numero 3 del piano superiore. In giallo sono riportati tutti gli eventi acquisiti, mentre in blu ci sono soltanto gli eventi “single shoot”, ovvero quelli che hanno prodotto segnale su una sola catena per ogni piano.

non è necessario calcolare per ogni evento il corrispondente $P_i(j)$ per ognuna della 18 *strips*, infatti è noto a priori che le uniche due candidate sono quelle tra le cui medie è compreso ΔT_j . Quindi, identificate queste due (diciamo k e $k + 1$), basta confrontare tra loro i valori $P_k(j)$ e $P_{k+1}(j)$.

Una volta identificate le *strips* colpite si può ottenere ad una stima della posizione d'incidenza con semplici considerazioni geometriche: prendendo come origine delle coordinate il centro del piano e identificando l'asse x con l'asse del piano, vale la formula

$$x = [72(N_{strip} - 1) + 18(N_{catena} - 1) - 639] \text{ cm}$$

dove N_{strip} è il numero della *strip* colpita e N_{catena} è il numero della catena che ha generato il segnale.

È necessario però tenere conto che c'è una certa percentuale di eventi per cui più di una delle catene di ogni piano producono un segnale soprasoglia; in tal caso si dice che la molteplicità dell'evento è diversa da 1 (o più precisamente è uguale al numero di catene soprasoglia). In questo caso c'è una possibile ambiguità nella ricostruzione

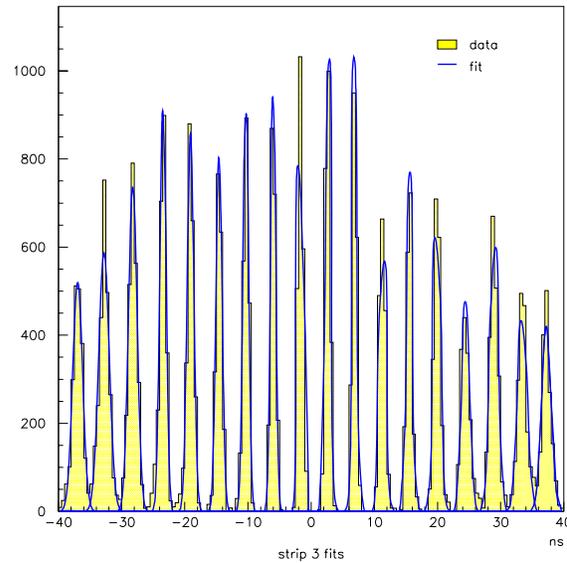


Figura 3.12: Fits dei picchi relativi alla semidifferenza della terza catena di *strips* del piano superiore.

della traccia, perchè non è chiaro quale delle *strips* colpite utilizzare.

La scelta effettuata è stata quella di scartare gli eventi a molteplicità 3 e 4, mentre per quelli a molteplicità 2 si verifica che le *strip* colpite siano adiacenti, ed in tal caso la posizione assegnata è la media delle due.

La risoluzione spaziale di questo sistema di tracciamento è chiaramente vincolata dalla dimensione trasversa delle *strips*, che risulta pari a 18mm .

Tempi e carica rilasciata

Note le posizioni d'impatto sui due piani attivi si può facilmente ricavare la posizione di passaggio della particella sui contatori posti all'interno del telescopio, e quindi cominciare la valutazione dei segnali prodotti dai contatori stessi.

In figura 3.13 sono rappresentati ad esempio gli spettri di tempo relativi al contatore di riferimento e al contatore prototipo. Come spiegato in precedenza, il lato sinistro del *reference counter* è stato ritardato rispetto al destro in modo che la temporizzazione del *trigger* sia sempre stabilita dal primo; nel grafico relativo osserviamo infatti che il tempo di arrivo di questo canale rispetto al *trigger* è praticamente costante, e

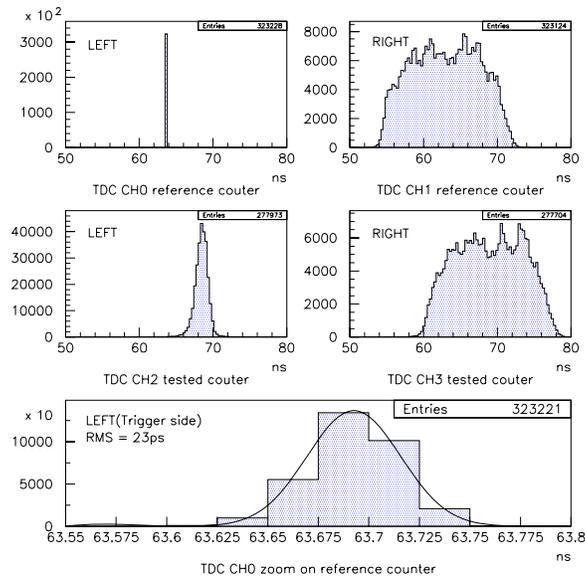


Figura 3.13: Spettri di tempo forniti dal contatore di riferimento e dal contatore prototipo. Il lato sinistro del contatore di riferimento è ritardato di $16ns$ rispetto all'altro lato in modo che sia sempre l'ultimo ad arrivare e quindi a stabilire la partenza del *trigger*. Il fit gaussiano su di esso fornisce una stima del rumore elettronico ($23ps$).

tutti gli altri tempi risultano riferiti ad esso.

Fittando questo spettro di tempo con una gaussiana si ottiene una stima del rumore elettronico introdotto dai moduli usati per l'acquisizione. Il valore ottenuto di $23ps$ risulta decisamente buono, e non influenzerà significativamente la risoluzione temporale finale.

Osservando gli altri grafici si nota che la larghezza degli spettri dei canali sul lato destro dei contatori risulta, come ci si aspetta, all'incirca doppia del tempo necessario al segnale luminoso prodotto nello scintillatore ad attraversarlo in tutta la sua lunghezza ($v \simeq 16 \text{ cm/ns}$), essendo in realtà la *differenza* di tempo di arrivo tra lato destro e sinistro, e non il tempo di arrivo all'estremità rispetto all'istante di passaggio della particella nel contatore.

Per quanto riguarda gli spettri di carica la prima operazione fatta è stata l'equilizzazione dei fototubi: l'alta tensione va regolata in base al guadagno del fototubo in modo tale che i segnali prodotti sugli ADC da particelle di uguale energia sia

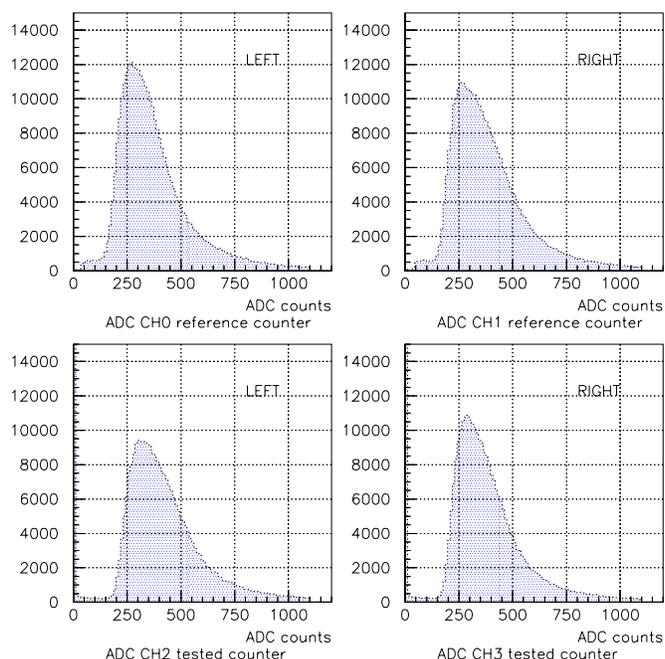


Figura 3.14: Spettri di ADC dei PM equalizzati in tensione una volta eliminato il piedistallo. Quest'ultimo si nota incorrispondenza del bin 0 dei due *plots* relativi al contatore prototipo che, non essendo nel *trigger*, acquisisce eventi nei quali non è stato attraversato a particelle.

pressochè uguale per ogni canale. Questa operazione si è svolta prendendo alcuni *runs* di dati con un solo fototubo acceso su ogni lato, selezionando soltanto una zona centrale di 10 cm del contatore tramite uno scintillatore aggiunto a questo scopo, e variando la tensione di alimentazione da $-100V$ a $+100V$ rispetto alla tensione nominale (alla quale il guadagno della catena di diodi è di 10^6).

La fase successiva consiste nell'eliminazione dei piedistalli dagli spettri. Gli ADC, infatti, restituiscono conteggi non identicamente nulli anche in assenza di segnali al loro ingresso perchè in corrispondenza del *gate* viene iniettata una piccola corrente (costante) che produce un *offset* proporzionale al *gate* stesso; così facendo la misura viene fatta nella zona in cui la risposta dell'ADC è più lineare.

Questo effetto si traduce in uno spettro simmetrico molto piccato su un valore (detto piedistallo) all'incirca proporzionale all'ampiezza del segnale di *gate* che pilota

il modulo. La carica effettivamente rilasciata nel rivelatore sarà quindi data (una volta calcolate le relative costanti di calibrazione) dalla differenza tra il valore misurato dall'ADC e il piedistallo. Alcuni *runs* sono stati quindi dedicati alla misura dei piedistalli, ottenuti acquisendo dati lasciando i contatori spenti; eliminati poi gli eventi fuori scala si ottengono spettri del tipo riportato in figura 3.14. Con questi dati a disposizione è possibile valutare le caratteristiche fisiche del contatore prototipo.

3.3.2 Lunghezza di attenuazione

Per ottenere una stima della lunghezza di attenuazione è sufficiente interpolare con una funzione esponenziale il grafico della carica collezionata dagli ADC in funzione della posizione. Un esempio di ciò che si ottiene è riportato in figura 3.15.

3.3.3 Fotostatistica

Dalle misure delle ampiezze dei segnali è possibile stimare il numero di fotoelettroni prodotti dal passaggio di una MIP al centro del contatore. Per il calcolo, siano definite le seguenti quantità:

- N_{phe} : numero totale di fotoelettroni prodotti in corrispondenza del passaggio di una MIP al centro del contatore;
- N_{phe}^i : numero di fotoelettroni sul lato i , con $i = L, R$ per i lati sinistro e destro;
- I_{ADC}^i : misura dell'ampiezza del segnale del lato i , fornita dall'ADC.
- g : guadagno dei fototubi (che si suppongono ben equalizzati);
- ϵ : efficienza del processo di moltiplicazione dei fotoelettroni (si suppone che sia uguale per entrambi i lati).

È chiaro che $I_{ADC}^i = g \epsilon N_{phe}^i$. Si definisca allora la quantità R come il rapporto tra la differenza e la somma dei segnali sui due lati del contatore:

$$R = \frac{I_{ADC}^L - I_{ADC}^R}{I_{ADC}^L + I_{ADC}^R} = \frac{N_{phe}^L - N_{phe}^R}{N_{phe}^L + N_{phe}^R} \quad (3.1)$$

Le quantità N_{phe}^i sono correlate dal fatto che $N_{phe} = N_{phe}^L + N_{phe}^R$, quindi

$$R = \frac{N_{phe}^L - N_{phe}^R}{N_{phe}} \quad (3.2)$$

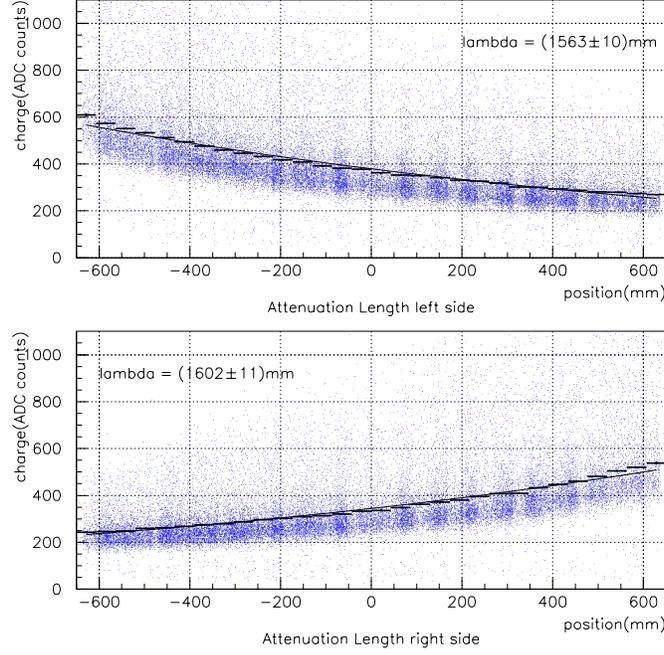


Figura 3.15: Misura della lunghezza di attenuazione degli scintillatori. Sono calcolate le stime di λ per entrambi i lati, e con i valori ottenuti se ne fa la media che nell'esempio riportato risulta pari a 1580 ± 30 mm.

e siccome $N_{phe}^R = N_{phe} - N_{phe}^L$, si ha

$$R = \frac{2N_{phe}^L}{N_{phe}} - 1 \quad (3.3)$$

ma dato che N_{phe}^L è distribuita secondo la binomiale, la deviazione standard su R è data da

$$\sigma_R = \frac{2}{N_{phe}} \sqrt{N_{phe} p (1-p)} = \frac{1}{\sqrt{N_{phe}}} \quad p = \frac{1}{2} \quad (3.4)$$

da cui si ottiene il numero di fotoelettroni totale:

$$N_{phe} = \frac{1}{\sigma_R^2} \quad (3.5)$$

In figura 3.16 è riportato lo spettro di R con la stima ottenuta per N_{phe} .

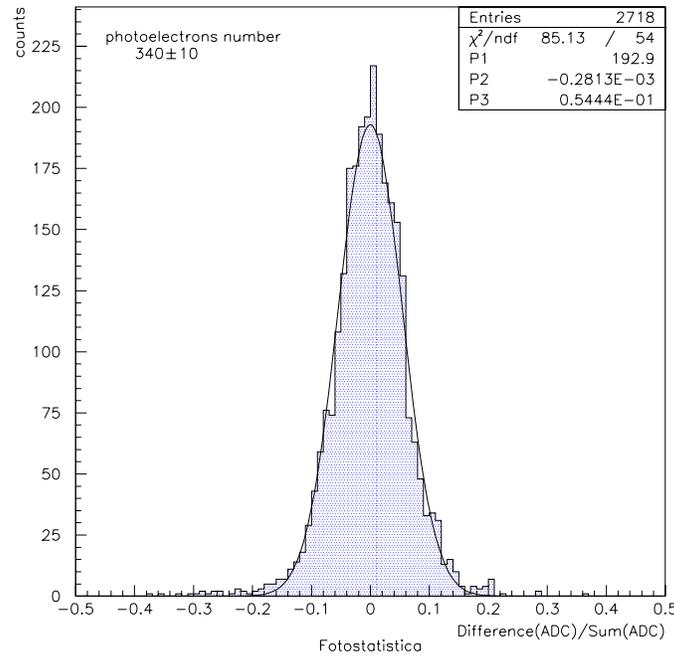


Figura 3.16: Stima del numero di fotoelettroni per una MIP al centro del contatore. La quantità graficata è il rapporto tra la semidifferenza e la semisomma delle ampiezze dei segnali ai due lati del contatore.

3.3.4 Correzione di Time Walk

Prima di utilizzare i tempi misurati dai TDC vanno corretti per l'effetto chiamato *time walk*: il fronte di salita (o meglio, di discesa visto che il segnale è negativo) dei segnali prodotti dai fototubi è molto ripido ma non ha una pendenza infinita; questo comporta che i discriminatori "a soglia" introducano nel segnale discriminato un ritardo che dipende dall'ampiezza del segnale di anodo.

Esiste un metodo empirico per compensare almeno in parte questo difetto che consiste nel sottrarre ai tempi misurati dai TDC (modificati dai ritardi prodotti dai discriminatori a soglia) un quantità del tipo

$$\Delta t = a \frac{1}{\sqrt{Q_{adc}}}$$

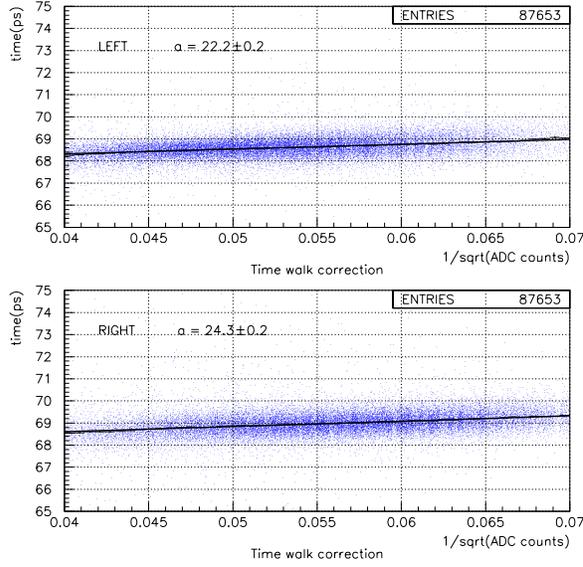


Figura 3.17: Calcolo dei parametri a per la correzione di *time walk*.

finalizzata a compensare le deviazioni dovute al *time walk*.

Le costanti a (una per ogni canale di TDC) si ricavano ad esempio fittando il plot del tempo t_i , $i = L, R$ in funzione di $1/\sqrt{Q_i}$, $i = L, R$ con una retta, come si può vedere nel grafico di figura 3.17

3.3.5 Velocità di propagazione della luce nello scintillatore

Siano ora t''^L e t''^R i tempi di misurati dai TDC ai due lati di un contatore corretti per il *time walk* (il motivo di questa notazione sarà chiaro una volta letto il paragrafo successivo). Detta L la lunghezza del contatore, si può esprimere t''^L in funzione della posizione di attraversamento della particella (x) e della velocità della luce nel contatore v_c :

$$t''^L = t_0 + (L/2 + x)/v_c \quad t''^R = t_0 + (L/2 - x)/v_c$$

Sottraendo la seconda equazione dalla prima ed invertendo quella ottenuta rispetto a v_c si ottiene:

$$v_c = \frac{2x}{t''^L - t''^R}$$

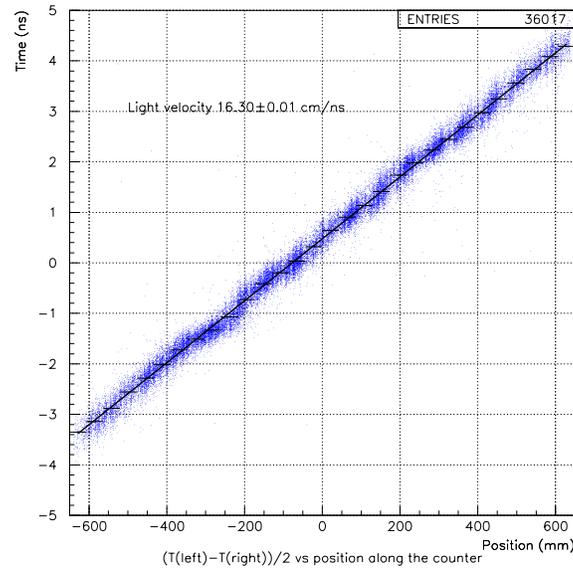


Figura 3.18: Andamento della semidifferenza dei tempi di arrivo ai due lati di un contatore in funzione della posizione. La velocità della luce è data dall'inverso della slope della retta.

Quindi si ottiene v_c dalla slope (facendone l'inverso) della retta che fitta l'andamento della semidifferenza di tempi rispetto alla posizione (figura 3.18).

3.3.6 Risoluzione sul tempo di volo

L'obiettivo del sistema TOF è quello di ottenere la misura del tempo di volo con la miglior risoluzione possibile. Per capire come viene fatto questo tipo di misura si faccia riferimento alla figura 3.19 per introdurre alcune quantità. Siano L_{ref} e L_{test} le lunghezze dei contatori di riferimento e di test, t_0 l'istante in cui una particella che vola verso il basso incide sul primo contatore (test) ed l lo spazio che percorre tra i due contatori a velocità βc . v_{ref} e v_{test} siano poi le velocità della luce misurate all'interno dei due contatori, mentre per i tempi si userà la seguente notazione:

- una t indica un tempo "assoluto", associato cioè al sistema di riferimento in cui la particella incide sulla prima slab all'istante t_0 ;
- una t' indica un tempo misurato da un TDC;

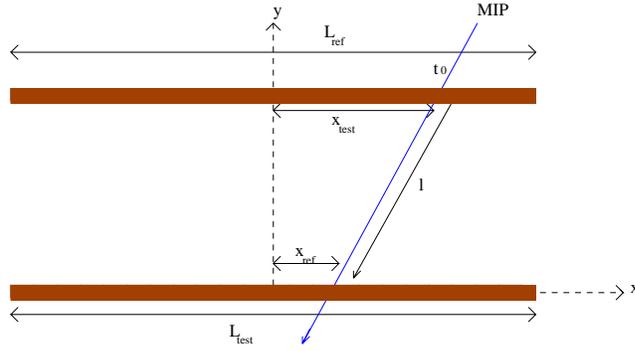


Figura 3.19: Disegno schematico della struttura del telescopio. Lo schema è abbastanza generale, e può rappresentare in modo semplificato ogni apparato per la misura del tempo di volo.

- una L come apice indica che si fa riferimento al lato sinistro del contatore, mentre una R sarà usata per il lato destro;
- i pedici ref e $test$ indicano a quale contatore si sta facendo riferimento (reference o test);
- δt rappresenta il ritardo introdotto dal discriminatore a causa del *time walk*.

Ad esempio, t_{ref}^L indica il tempo misurato dal TDC per il lato sinistro del contatore di riferimento.

Dalla figura 3.19 è facile convincersi che valgono le seguenti espressioni per i tempi assoluti:

$$t_{ref}^L = t_0 + \frac{l}{\beta c} + \frac{L_{ref}/2 + x_{ref}}{v_{ref}} + \tau + \delta t_{ref}^L \quad (3.6)$$

$$t_{ref}^R = t_0 + \frac{l}{\beta c} + \frac{L_{ref}/2 - x_{ref}}{v_{ref}} + \tau + \delta t_{ref}^R \quad (3.7)$$

$$t_{test}^L = t_0 + \frac{L_{test}/2 + x_{test}}{v_{test}} + \tau + \delta t_{test}^L \quad (3.8)$$

$$t_{test}^R = t_0 + \frac{L_{test}/2 - x_{test}}{v_{test}} + \tau + \delta t_{test}^R \quad (3.9)$$

dove τ riassume i ritardi dovuti ai cavi e all'elettronica tra le estremità dei contatori e gli ingressi dei TDC. I valori di tempo forniti da questi ultimi saranno la differenza tra i tempi assoluti t_i^j ed il tempo di formazione del *trigger* $t_{trigger}$:

$$t_i^j = t_i^j - t_{trigger} \quad (3.10)$$

ma quest'ultimo, come si è visto, è governato da t_{ref}^L a causa di un ritardo di $16ns$ inserito prima dell'ingresso di questo segnale nella coincidenza che crea il *fast trigger*; quindi, indicando con $\tau_{trigger}$ l'insieme dei ritardi tra l'arrivo del segnale dal contatore e la conseguente generazione del *fast trigger*, si ha

$$t_{trigger} = t_{ref}^L - \tau_{trigger} \quad (3.11)$$

Ora è possibile esprimere il tempi misurati dai TDC in modo da far sparire i tempi assoluti, semplicemente sostituendo a t_{ref}^L la sua espressione data in 3.6.

Le espressioni che ne risultano sono:

$$t_{ref}^L = \tau_{trigger} \quad (3.12)$$

$$t_{ref}^R = -2\frac{x_{ref}}{v_{ref}} - \delta t_{ref}^L + \delta t_{ref}^R + \tau_{trigger} \quad (3.13)$$

$$t_{test}^L = -\frac{l}{\beta c} + \frac{L_{test}}{2v_{test}} - \frac{L_{ref}}{2v_{ref}} + \frac{x_{test}}{v_{test}} - \frac{x_{ref}}{v_{ref}} + \delta t_{test}^L - \delta t_{ref}^L + \tau_{trigger} \quad (3.14)$$

$$t_{test}^R = -\frac{l}{\beta c} + \frac{L_{test}}{2v_{test}} - \frac{L_{ref}}{2v_{ref}} - \frac{x_{test}}{v_{test}} - \frac{x_{ref}}{v_{ref}} + \delta t_{test}^R - \delta t_{ref}^L + \tau_{trigger} \quad (3.15)$$

Facendo poi le semisomme dei due lati di ogni contatore e sottraendoli tra loro si arriva alla seguente formula

$$\begin{aligned} \frac{t_{test}^L + t_{test}^R}{2} - \frac{t_{ref}^L + t_{ref}^R}{2} &= \frac{1}{2} \left(\frac{L_{test}}{v_{test}} - \frac{L_{ref}}{v_{ref}} \right) - \frac{l}{\beta c} - 2\tau_{trigger} + \\ &+ \frac{\delta t_{test}^L + \delta t_{test}^R}{2} - \frac{\delta t_{ref}^L + \delta t_{ref}^R}{2} \end{aligned} \quad (3.16)$$

dove si può portare al primo membro i termini correttivi per il time walk che vanno a sottrarsi ai rispettivi tempi misurati dai TDC.

Chiamando allora il tempo misurato corretto per il time walk $t''_i^j = t'_i^j - \delta t_i^j$ la 3.16 diventa

$$\frac{t''_{test}^L + t''_{test}^R}{2} - \frac{t''_{ref}^L + t''_{ref}^R}{2} = \frac{1}{2} \left(\frac{L_{test}}{v_{test}} - \frac{L_{ref}}{v_{ref}} \right) - \frac{l}{\beta c} - 2\tau_{trigger} \quad (3.17)$$

Sul lato destro di questa equazione ci sono tre termini costanti, i primi due ($L_i/2v_j$) che dipendono soltanto dalla geometria del rivelatore e dall'indice di rifrazione del plastico scintillante, ed il terzo ($2\tau_{trigger}$) che riassume i contributi dei vari ritardi dovuti alla generazione del *fast trigger*. Siccome il termine rimanente $l/\beta c$ non è altro che il tempo di volo t_{of} , si ha:

$$t_{of} = \frac{t''_{ref}^L + t''_{ref}^R}{2} - \frac{t''_{test}^L + t''_{test}^R}{2} + Cost. \quad (3.18)$$

La 3.18 mostra che per ottenere il tempo di volo basta calcolare la differenza delle semisomme dei tempi misurati dai TDC sui due lati dei contatori dopo averli corretti per l'effetto di *time walk*. Il termine costante aggiuntivo viene determinato sperimentalmente in fase di calibrazione dell'apparato, in modo tale da averne una stima precisa che renda accurata la misura del tempo di volo.

Rimane ancora un punto importante da evidenziare: i termini correttivi δt_i^j che compaiono nelle ultime formule non sono esattamente quelli calcolati nel paragrafo 3.3.4 perchè per il calcolo del parametro a che compare nella 3.3.4 è stato fittato il plot dei tempi *misurati* e non di quelli assoluti; essi sono influenzati quindi dal *time walk* del canale che pilota il *trigger* (nella fattispecie δt_{ref}^L). Più precisamente le correzioni temporali Δt_i^j realmente calcolate tramite la misura dei parametri a_i^j rappresentano le differenze tra le correzioni *effettive* δt_i^j e il ritardo che entra nel *trigger* δt_{ref}^L :

$$\Delta t_i^j = \delta t_i^j - \delta t_{ref}^L$$

in tabella 3.3.6 si riporta un esempio dei valori ottenuti per tali parametri nel caso del contatore prototipo.

$\Delta t_{ref}^L [ps]$	$\Delta t_{ref}^R [ps]$	$\Delta t_{test}^L [ps]$	$\Delta t_{test}^R [ps]$
0.1 ± 0.1	3.0 ± 0.2	22.2 ± 0.2	24.3 ± 0.2

In particolare si nota che $\Delta t_{ref}^L = \delta t_{ref}^L - \delta t_{ref}^L$ è compatibile con il valore atteso (0 ps).

Purtroppo il sistema 3.3.6 non è invertibile a causa dell'equazione relativa al lato sinistro del contatore di riferimento che è un'identità. Si può, nel migliore dei casi, trovare il valore dei δt_i^j a meno di una costante additiva incognita.

Tornando alla 3.18, se sommando un termine nullo al primo membro pari a $\delta t_{ref}^L - \delta t_{ref}^L$ si vede che è possibile esprimere t_{of} proprio in funzione dei parametri Δt_i^j misurati; infatti, ridefinendo $\bar{t}_i^j = t_i^j - (\delta t_i^j - \delta t_{ref}^L) \equiv t_i^j - \Delta t_i^j$, si ha

$$t_{of} = \frac{\bar{t}_{ref}^L + \bar{t}_{ref}^R}{2} - \frac{\bar{t}_{test}^L + \bar{t}_{test}^R}{2} + Cost. \quad (3.19)$$

che esprime t_{of} in funzione soltanto dei parametri ottenuti sperimentalmente.

In figura 3.20 sono riportati gli spettri di tempo di volo calcolati senza e con le correzioni di *time walk*. Nel secondo caso la risoluzione migliora di 128ps.

In realtà i valori di risoluzione riportati in figura 3.20 sono falsati dal fatto che il *trigger* è formato dal contatore di riferimento. Di conseguenza il valore di risoluzione che si ottiene è la somma in quadratura delle risoluzioni del contatore di riferimento e di quello sotto *test*. Per ottenere un valore di risoluzione attendibile è necessario

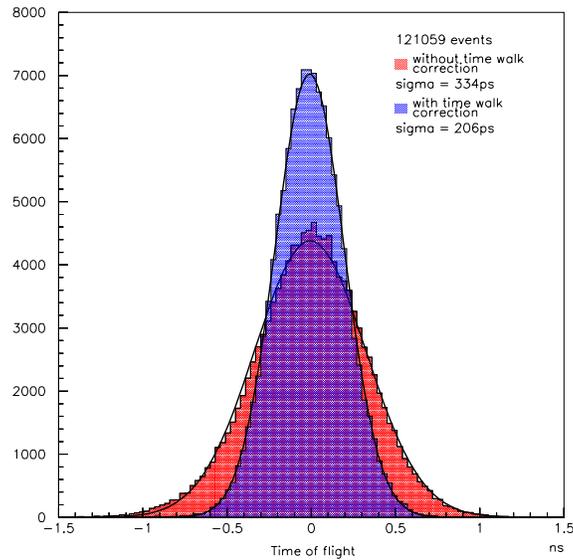


Figura 3.20: Spettro di tempo di volo calcolato senza le correzioni di time walk (rosso) e con le correzioni (blu). Il miglioramento nella risoluzione è di $128ps$.

stimare e sottrarre in quadratura la risoluzione del contatore di riferimento. Come stima di questa quantità è stata utilizzata la media delle risoluzioni sulla semidifferenza dei tempi di arrivo dei segnali sui due lati, per 12 posizioni (selezione di 10 cm) distribuite su tutta la lunghezza del contatore. Nella semidifferenza infatti sia le correzioni di *time walk* (l'ampiezza sui due lati è fortemente correlata) che il rumore elettronico tendono ad annullarsi reciprocamente. Il valore ottenuto, pari a

$$\sigma_{ref} = 150ps$$

è stato poi sottratto in quadratura dalla risoluzione sul tempo di volo, ottenendo:

$$\sigma_{TOF} = 141ps$$

3.3.7 Risoluzione intrinseca

Per una completa analisi delle caratteristiche del contatore è stato valutato l'andamento in funzione della posizione di incidenza della risoluzione che si ha sui singoli

lati del contatore, sulla loro semidifferenza, e sul tempo di volo.

Lo spettro del tempo di arrivo su un lato del contatore è determinato dalle quantità che nelle 3.8 e 3.9 sono state scritte come $(L_{test}/2 \pm x_{test})/v_{test}$; d'ora in poi saranno riferite con t^{*L} e t^{*R} .

Per ottenere le espressioni più convenienti per queste due quantità basta fare i seguenti passaggi:

1. esprimere le 3.8 e 3.9 in funzione di t^{*L} e t^{*R} ;
2. scrivere esplicitamente $t_{trigger}$ in funzione delle quantità che appaiono in 3.6;
3. sottrarre $t_{trigger}$ a t_{test}^i per ottenere due espressioni che invertite danno t^{*L} e t^{*R} ;
4. notare che la quantità $-\tau_{trigger} + x_{ref}/v_{ref}$ non è altro che il *mean time* \bar{T} definito da:

$$\bar{T} = \frac{t_{ref}^L - \Delta t_{ref}^L + t_{ref}^R - \Delta t_{ref}^R}{2}$$

Si ottengono infine le espressioni

$$t^{*i} = t_{test}^i - \Delta t_{test}^i + t_{of} - \bar{T} + \frac{L_{ref}}{2v_{ref}} \quad (3.20)$$

La risoluzione ottenuta dagli spettri di queste quantità è la somma in quadratura della risoluzione del lato del contatore prototipo e di quella complessiva del contatore di riferimento; per ottenere la risoluzione intrinseca sul singolo lato, ai valori ottenuti è stata sottratta (in quadratura) la stima già data della risoluzione del contatore di riferimento.

Analogamente a quanto fatto per quest'ultimo, sono state selezionate 12 zone larghe 10 cm su tutta la lunghezza del contatore, valutando la risoluzione su tutti i punti. L'andamento generale in funzione della posizione è parametrizzato da:

$$\sigma_i(x) = \sqrt{\frac{\sigma_1^2}{N} + \frac{\sigma_2^2 x^2}{N} + \sigma_3^2} \quad (3.21)$$

le quantità che vi compaiono sono:

- x : distanza dal fotomoltiplicatore;
- N : numero di fotoelettroni;
- σ_1/\sqrt{N} : risoluzione della risposta del fotomoltiplicatore e dell'elettronica del *trigger*;

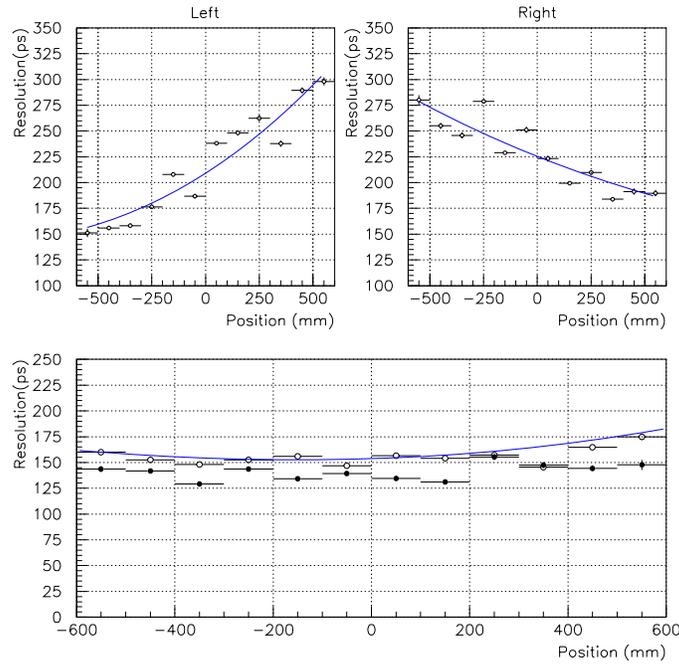


Figura 3.21: Andamento della risoluzione intrinseca per il contatore studiato. Le risoluzioni su lato singolo (i due grafici in alto) e sul tempo di volo (pallini neri nel grafico in basso) sono state corrette per le risoluzioni del contatore di riferimento. I pallini bianchi nel grafico in basso sono relativi alla semidifferenza.

- $\sigma_2 x / \sqrt{N}$: termine di risoluzione dovuto alla dispersione dei cammini ottici;
- σ_3 : rumore elettronico.

Per determinare N in funzione della posizione sul contatore vanno utilizzati la lunghezza di attenuazione e il numero totale di fotoelettroni calcolati in precedenza. Più precisamente, si ha

$$N(x) = \frac{N_{phe}}{2} e^{\frac{x-L/2}{\lambda}} \quad (3.22)$$

se L è la lunghezza del contatore.

Oltre alla risoluzione sui singoli lati, è stato studiato l'andamento in funzione della posizione della risoluzione su semidifferenza dei due lati; in questo caso, come si è già accennato, i contributi di *time walk*, di fluttazioni introdotte dal contatore di

riferimento e di rumore tendono ad eliminarsi. Nel caso del tempo di volo sono invece state applicate le correzioni analogamente a quanto fatto nel paragrafo precedente. In figura 3.21 sono riportati i grafici relativi a lati singoli, alla semidifferenza e al tempo di volo della risoluzione in funzione della posizione. I grafici relativi ai lati singoli sono stati *fittati* con la 3.21, mentre nel grafico in basso è disegnata la funzione

$$\sigma(x) = \frac{\sqrt{\sigma_L^2(x) + \sigma_R^2(x)}}{2} \quad (3.23)$$

$\sigma_L^2(x)$ e $\sigma_R^2(x)$ sono espresse dalla 3.21 dove al posto di σ_1 , σ_2 e σ_3 sono stati sostituiti i valori forniti dal *fitting*.

Si nota che i punti relativi alla semidifferenza seguono precisamente l'andamento della funzione $\sigma(x)$, mentre la risoluzione sul tempo di volo si assesta su valori più bassi; la causa è che la stima ottenuta della risoluzione intrinseca del contatore di riferimento, valutata sulla semidifferenza, risulta maggiore del suo valore reale perchè è comprensiva di effetti non simmetrici sui due lati e che quindi non si annullano. L'interpolazione ottenuta con la funzione 3.23 si attesta al centro del contatore sul valore di $\sigma = 152$ ps.

3.4 Fase 2: VME

Dopo una fase di presa dati con il sistema di acquisizione basato su *bus* CAMAC si è passati all'assemblaggio della versione VME. La scelta di migrare a questo tipo di *bus* è basata su tre considerazioni fondamentali:

- il VME permette un miglior interfacciamento tra il PC ed i moduli di elettronica;
- il mercato mette a disposizione una notevole quantità di dispositivi VME con prestazioni elevate;
- esso consente un facile adattamento dell'elettronica CAMAC non sostituibile grazie ad interfacce che mettono in comunicazione i due *buses*.

L'*upgrade* al *bus* VME permette in generale di ottenere prestazioni più elevate mantenendo la possibilità di utilizzare l'elettronica CAMAC non disponibile nel nuovo *standard*.

Nella logica del sistema di acquisizione del telescopio la modifica principale è stata il passaggio dal funzionamento in *common start* a quella in *common stop* per i TDC. Questa modifica permette di evitare l'utilizzo degli ingombranti cavi utilizzati come catena di ritardo perchè in questa configurazione i singoli canali fanno partire le

misure dei tempi che si fermano in corrispondenza di un segnale comune di *stop* dato dal *fast trigger*.

A parte questo i blocchi di *trigger* e *front end* non hanno subito variazioni sostanziali, mentre qualche aggiustamento è stato fatto sul resto del DAQ. Il motivo che ha reso necessario un certo adattamento dello schema è che il funzionamento dei moduli VME utilizzati è simile ma non identico a quello dei moduli CAMAC, ed in particolare i TDC (CAEN mod. v488) sono dotati di un ingresso di *reset* che, oltre ad eliminare l'evento acquisito, reinizializza il modulo al suo stato iniziale, ma non hanno un *clear*. Si renderebbe così necessaria la reimpostazione dei parametri di lavoro del modulo in seguito ad ogni *fast trigger*, operazione lunga e complessa che non può essere fatta durante la presa dati.

La soluzione adottata è quella di utilizzare lo *slow trigger* come stop per i TDC dedicati all'acquisizione delle *strips*, e di acquisire tutti gli eventi forniti dal *fast trigger* per quanto riguarda i contatori.

I segnali delle *strips*, infatti, non necessitano dell'ottima risoluzione temporale necessaria per i contatori, e quindi non risentono delle fluttazioni introdotte dalla presenza delle *strips* nella coincidenza.

Non potendo effettuare un *clear* ai TDC dei contatori, nel loro caso è necessario acquisire anche gli eventi non confermati dal secondo. Gli eventi non voluti vengono eliminati in seguito tramite un filtro software.

Il Constant-Fraction Discriminator

Come si è visto in precedenza, il difetto principale dei discriminatori a soglia è l'introduzione di un ritardo nella formazione del segnale dovuto al tempo di salita dell'impulso generato dal fototubo dalla sua *baseline* al valore corrispondente alla discriminazione (*time walk*). Questo ritardo dipende dall'ampiezza del segnale generato dal PM.

Una soluzione alternativa alla correzione empirica spiegata nel paragrafo 3.3.4 consiste nel modificare il funzionamento del discriminatore. Facendo riferimento alla figura 3.22 si nota che il tempo di salita totale dell'impulso è pressapoco costante a parità di fototubo e di tensione di alimentazione, essendo dovuto fondamentalmente al tempo di attraversamento del fototubo stesso della valanga di elettroni; la *slope* del segnale si adatta di conseguenza. Se invece di fissare la soglia di discriminazione si fa in modo che la formazione del segnale cominci quando esso raggiunge una frazione costante della sua altezza, il ritardo introdotto diventa indipendente dall'altezza dell'impulso.

Infatti siano $L_1 > L_2$ le altezze di due impulsi dati dallo stesso PM. Essendo il tempo di salita T lo stesso per entrambi, gli angoli sottesi dalla rampa di salita saranno $\Theta_1 > \Theta_2$. I ritardi introdotto dal discriminatore saranno $T_1 = kL_1 \cot \Theta_1$ e

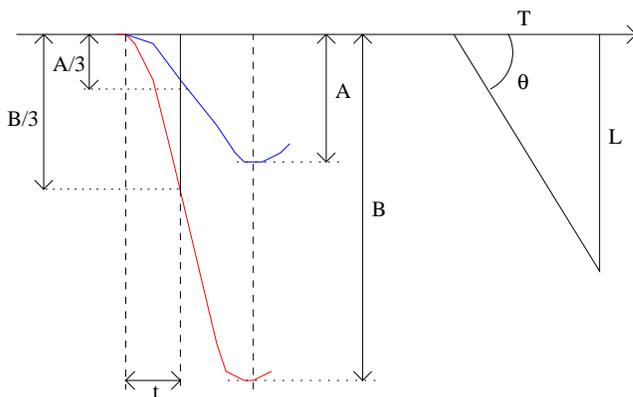


Figura 3.22: Funzionamento di un discriminatore a frazione costante. Il segnale viene discriminato quando raggiunge un'altezza pari ad una frazione costante l'altezza di picco del segnale (in questo caso $1/3$).

$T_2 = kL_2 \cot \Theta_2$ dove k è la frazione di discriminazione, e siccome $\cot \Theta_i = T/L_i$ si ha

$$T_1 = T_2 = kT$$

Per fissare la soglia dinamicamente è però necessario conoscere l'altezza massima del segnale, ed è quindi necessario attendere che il segnale raggiunga il suo massimo. Solo una volta identificato quest'ultimo l'impulso NIM viene prodotto, ed esso risulta quindi affetto da un ulteriore ritardo che, essendo costante, non influisce negativamente sul *timing* del segnale.

Nel DAQ del telescopio si è fatto uso del *Constant Fraction Discriminator* (CFD) v812 prodotto dalla CAEN per la formazione dei segnali anodici dei contatori (*test* e *reference*). L'analisi delle prestazioni ha messo in evidenza una perdita di efficienza della catena di acquisizione rispetto al numero di eventi di circa il 5% a causa dell'errata identificazione della soglia di discriminazione da parte del suddetto modulo. In questo gruppo di eventi, il segnale discriminato subisce un ritardo di circa $10ns$ rispetto agli altri, e si è quindi deciso di scartarli dalle analisi.

Per gli eventi buoni l'andamento delle curve di correzione per il *time walk* risulta effettivamente piatto, dimostrando che la soluzione utilizzata risolve effettivamente il problema del *time walk*; sembra però che vengano introdotte, rispetto ai discriminatori a soglia usati nella fase 1, notevoli fluttuazioni temporali (*jitter*) che compromettono la misura della risoluzione. Ciò rende sconveniente l'utilizzo del discriminatore a frazione costante per lo studio delle caratteristiche temporali dei contatori.

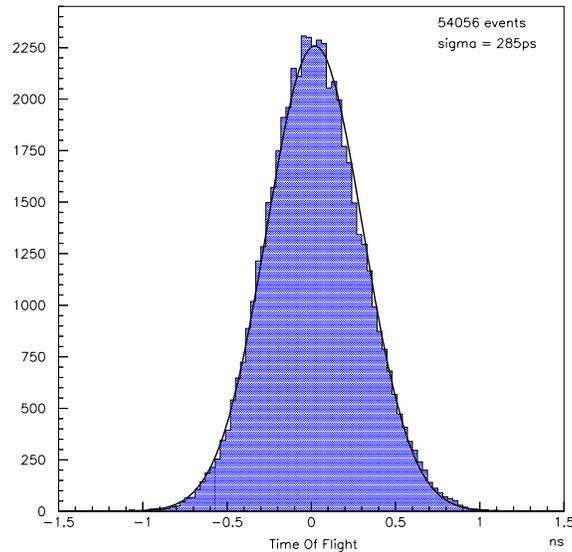


Figura 3.23: Spettro di TOF con il sistema di acquisizione VME.

3.4.1 Confronto con la fase 1

Dalle prime analisi fatte con il nuovo *setup* si notano alcuni peggioramenti nella misura della risoluzione. In figura 3.23 si può vedere lo spettro di TOF ottenuto con esso, dove non sono state applicate le correzioni di *time walk* visto l'utilizzo del discriminatore CFD. La risoluzione, che risulta pari a $285ps$ (peggiore a quella ottenuta con il *setup* CAMAC) va corretta sottraendo in quadratura la risoluzione del contatore di riferimento.

Ripetendo il procedimento fatto nella fase 1 si ottiene per quest'ultima un valore pari a

$$\sigma_{ref} = 204ps$$

Anch'esso risulta quindi notevolmente più alto di quanto ottenuto con il sistema basato su CAMAC.

Sottraendo in quadratura questo valore alla σ_{TOF} di figura 3.23 si ottiene

$$\sigma_{TOF} = 199ps$$

che risulta maggiore di quasi $60ps$ rispetto al valore ottenuto con il sistema CAMAC. Lo studio della risoluzione intrinseca dei contatori in questa seconda fase evidenzia

alcuni problemi dell'elettronica (figura 3.24). Infatti non si riescono ad estrapolare gli stessi risultati ottenuti nella fase 1 a causa di due fattori:

- fluttuazioni temporali introdotte dal discriminatore a frazione costante (*jitter*);
- effetti di non linearità e dipendenza dalla temperatura del TDC.

L'insieme di questi fattori permettere di identificare come soluzione più sensata la realizzazione di un sistema di acquisizione dati ibrido CAMAC-VME che permetta di utilizzare elettronica ormai testata e ben conosciuta assieme a nuove soluzioni che gradualmente rimpiazzeranno le componenti obsolete del sistema.

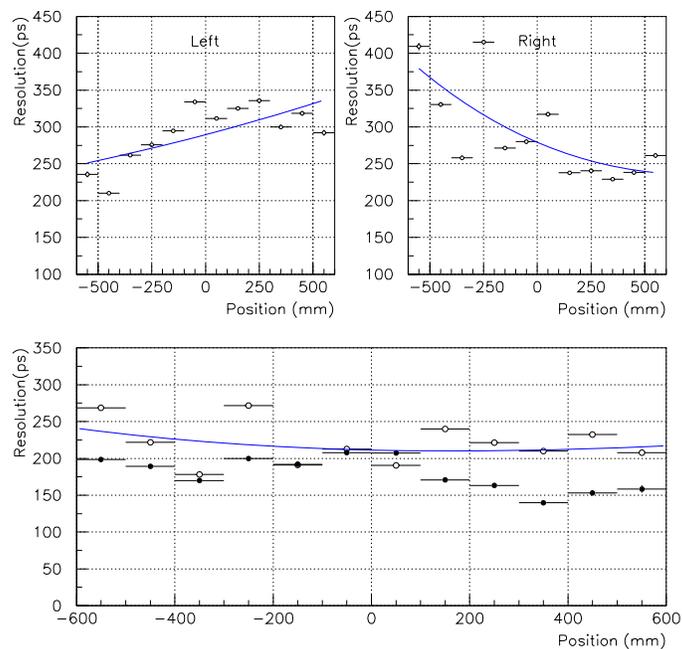


Figura 3.24: Studio della risoluzione intrinseca del contatore prototipo ottenuta con il sistema basato su VME. Si ha ancora la sovrapposizione della curva estrapolata dagli andamenti delle risoluzioni dei singoli lati con i punti relativi alla semidifferenza (pallini bianchi), e analogamente a caso precedente il punti relativi al tempo di volo (pallini neri) si assestano al di sotto della curva a causa della sovrastima della risoluzione intrinseca del contatore di riferimento.

Capitolo 4

Il software di acquisizione dati

Il lavoro più impegnativo per completare la migrazione al bus VME è stato la scrittura del codice per l'acquisizione. Il sistema scelto per lo sviluppo è stato un PC Linux (RedHat7.2) interfacciato al bus tramite una scheda SBS 616 (che collega i bus PCI e VME), i cui *driver* di gestione a basso livello dell'interfaccia sono distribuiti dalla casa costruttrice.

Si è scelto di utilizzare come linguaggio di programmazione il C/C++, ottimo compromesso tra versatilità (essendo *object oriented*) e velocità (è un linguaggio compilato). Quest'ultima è il parametro chiave che si deve massimizzare nei sistemi di acquisizione HEP visti i *rate* di eventi che talvolta si hanno a disposizione.

4.1 Struttura generale

Per farsi un'idea del risultato che si è cercato di ottenere si può partire dai requisiti richiesti al DAS (*Data Acquisition Software*):

- *velocità*: il DAS deve essere in grado di gestire il *rate* fisico imposto dall'elettronica di acquisizione dati;
- *versatilità*: a priori la struttura e la logica del sistema non è definita, di conseguenza i parametri del DAS devono essere facilmente modificabili;
- *indipendenza tra i diversi processi funzionali*: la fase di acquisizione deriva dalla commistione di diversi processi ognuno dei quali svolge un compito diverso. Tipicamente si ha un processo di *acquisition*, uno di *monitoring* ed un di *slow control*, che devono interfacciarsi mantenendo più indipendenza possibile.
- *aggiornabilità*: siccome l'aggiornamento dell'elettronica utilizzata in HEP è continuo, il DAS deve essere in grado di seguire facilmente questo processo.

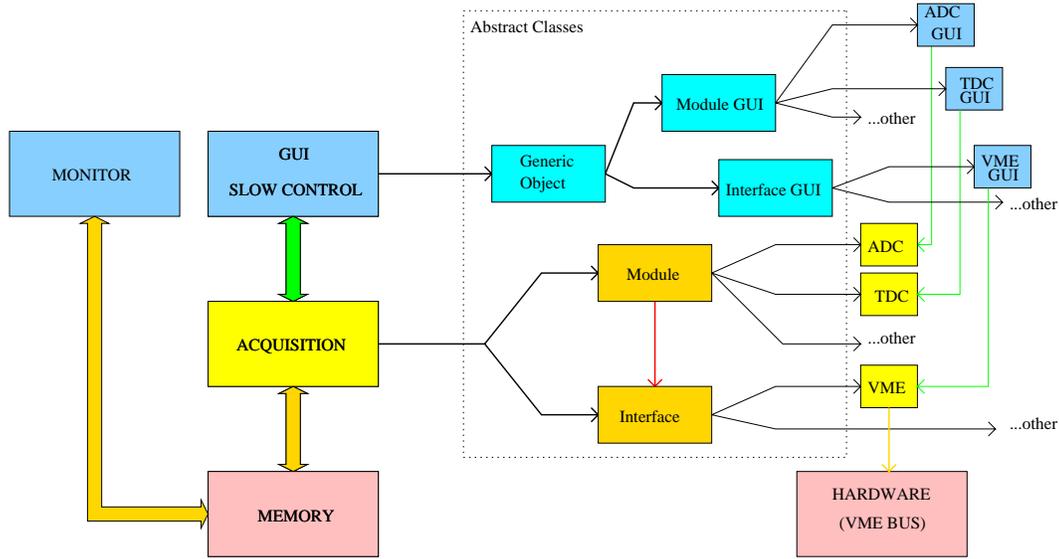


Figura 4.1: Struttura generale del DAS (Data Acquisition Software).

- *semplicità*: il software non verrà probabilmente utilizzato da specialisti, e dovrà quindi essere facilmente accessibile a qualsiasi tipo di utente.

Per ottenere contemporaneamente questi requisiti si è impostato il DAS su una struttura stratificata (figura 4.1) che permette di rendere indipendenti i processi che svolgono compiti diversi e che quindi soddisfano diverse richieste. Ad esempio, lo strato che gestisce l'interfacciamento con l'utente (blocchi in blu) è orientato verso la semplicità di utilizzo e configurazione, mentre lo strato sottostante (in giallo), che si occupa dell'acquisizione vera e propria, ha come *target* fondamentale la velocità. Parallelamente a questo tipo di stratificazione verticale ne è presente un'altra (probabilmente la più importante) che si sviluppa orizzontalmente. È grazie al meccanismo nascosto dietro a questa ramificazione che si ottiene la versatilità del sistema.

Infatti, i blocchi principali di figura 4.1 (*GUI* e *Acquisition*), che si interfacciano all'utente tramite le fasi di *setup* (*GUI*, *Graphical User Interface*) e di presa dati (*Acquisition*), NON possono accedere direttamente ai blocchi sottostanti (come ad esempio ADC, TDC, VME), ma lo fanno attraverso oggetti intermedi che gestiscono l'interfacciamento tra i due. Questi oggetti, che nella fattispecie prendono il nome di *classi astratte* dal gergo della programmazione *object oriented* (OO), rappresenta le fondamenta sulle quali è costruito il sistema di acquisizione, che ora verrà descritto in dettaglio.

4.1.1 Utilizzo delle classi

La programmazione OO è basata sul dogma che ogni cosa esistente rientri nella categoria “oggetto”, ed ogni tipo di oggetto non sia altro che un elemento di una generica “classe” di oggetti che condividono diverse proprietà. Nel caso specifico della programmazione, una classe non è altro che l’unione di una generica struttura dati con un insieme di funzioni che operano su questa struttura.

L’implementazione di un programma prevede la modellizzazione di ciò che si vuole ottenere in termini di una struttura gerarchica di classi, le cui *istanze* (gli oggetti reali che appartengono a tali classi) gestiscono l’evoluzione del programma.

Per fare un esempio inerente all’argomento trattato in questo lavoro, un sistema di acquisizione si può ritenere appartenente ad una ipotetica classe di oggetti detta DAQ, e tale classe sarebbe certamente dotato di *metodi* (il nome con cui si indicano le funzioni che agiscono sull’oggetto) quali *start acquisition*, *stop acquisition*, *update histograms* e via dicendo.

Ogni singola istanza di questa classe avrebbe però proprietà diverse dalle altre: una potrebbe ad esempio essere costituita da un semplice modulo ADC, ma nulla vieta che un’altra possa essere il complesso sistema di acquisizione di un esperimento di HEP.

Ogni elemento di questa ipotetica classe DAQ si può pensare composto da oggetti più piccoli ma più dettagliati che svolgono compiti specifici (ad esempio TDC, ADC, discriminatori, etc.) riuniti però da caratteristiche comuni (indirizzo, posizione nel crate, etc.), dando origine ad una struttura nidificata che si propaga fino alla granularità massima che si voglia immaginare.

Con questo semplice esempio sono già state introdotte alcune relazioni fondamentali che si possono presentare tra classi diverse: *l’appartenenza*, che si ha quando un oggetto è in realtà costituito da altri oggetti minori, e *l’ereditarietà*, che descrive come oggetti dalle caratteristiche comuni ereditano queste caratteristiche da una classe di oggetti più generale.

4.1.2 Moduli e interfacce: classi di lettura

Il DAS è basato su due classi fondamentali che prendono il nome di *Modulo* e *Interfaccia*. Sono in particolare classi *astratte*, il che significa che non è possibile creare istanze direttamente di queste classi ma soltanto attraverso le loro classi *figlie*, ovvero quelle che ereditano la loro struttura.

Questa suddivisione nasce dall’idea che ogni sistema di acquisizione dati sia costituito essenzialmente da questi due tipi di oggetti. Le interfacce si occupano di gestire i dispositivi che permettono la comunicazione tra il PC ed i moduli di acquisizione

dati (quali ad esempio un bus, una porta parallela, un link di rete, etc.), mentre i moduli rappresentano quegli oggetti che vengono effettivamente utilizzati per l'acquisizione ed il controllo (TDC, ADC, I/O register, etc.).

Interfacce

Si riporta come esempio un estratto della dichiarazione della classe interfaccia (il cui nome è `interface_c`).

```
class interface_c {
    /* ... dichiarazioni di variabili private ... */
public:
    interface_c();
    virtual ~interface_c();
    /* ... dichiarazioni di alcuni metodi di servizio ... */
    virtual int reset() = 0;
    virtual int read( void *structPointer );
    virtual int write( void *structPointer );
    virtual int generic( void *structPointer );
    static interface_c * factory(string & type);
};
```

La parte più importante è quella che segue l'istruzione `public`, la dichiarazione dei metodi pubblici (utilizzabili quindi dall'esterno dell'oggetto stesso) tramite i quali si agisce sull'interfaccia. Il primo di questi è il cosiddetto *costruttore*, un metodo che ha lo stesso nome della classe e che viene eseguito non appena la classe viene istanziata. Viceversa il *distruttore* è dichiarato con il nome della classe preceduto da una tilde, e viene eseguito al momento della distruzione dell'oggetto.

Il costruttore sarà quindi utilizzato in generale per connettersi ed impostare le proprietà basilari dell'interfaccia, mentre il distruttore di occuperà della chiusura.

Alle dichiarazioni di costruttore e distruttore seguono le dichiarazioni dei metodi generici dell'interfaccia, la maggior parte dei quali è preceduta dalla parola `virtual`; questi metodi sono appunto detti *virtuali*.

Un metodo virtuale può avere una implementazione vera e propria nel contesto della classe in cui viene dichiarato, ma non è detto che questa venga usata. Infatti se in una classe che eredita un metodo virtuale questo metodo viene ridefinito con una nuova implementazione, al momento della chiamata al metodo il corpo eseguito è quello ridefinito nella classe figlia, anche se il riferimento all'oggetto avviene tramite un puntatore della classe madre. In particolare, se la dichiarazione del metodo è

seguita da `= 0`, non è possibile istanziare questo tipo di oggetti finchè tali metodi non vengono reimplementati all'interno di una classe figlia. Questo tipo di classe si dice *puramente astratta*, e `interface_c` è proprio una di esse.

Riassumendo, la classe `interface_c` è puramente astratta a causa del metodo `reset()` che termina con `= 0`. Non è quindi possibile istanziare oggetti di questo tipo prima di definire una classe figlia (che eredita quindi la struttura di `interface_c`) in cui sia presente l'implementazione del metodo `reset()`. I quattro metodi la cui dichiarazione è riportata nell'estratto di codice precedente sono le funzioni fondamentali che permettono all'interfaccia di funzionare. Il significato del metodo `reset()` è riassunto nel nome stesso; ogni interfaccia reale implementerà questo metodo in modo che ripristini le impostazioni di default che le sono assegnate. Spesso molte interfacce (il VME ne è un esempio) sono dotate di un segnale che assolve proprio questo compito.

Le tre funzioni rimanenti sono quelle tramite le quali viene gestito il dialogo tra il programma e i moduli che saranno associati all'interfaccia. L'unico parametro di questi metodi è un generico puntatore (`void *`) il cui scopo è di indicare all'interfaccia stessa l'indirizzo in memoria della struttura che contiene i dati che il modulo ritiene giusto passargli.

Si consideri il seguente esempio la seguente situazione: durante l'esecuzione del programma, un modulo verrà interpellato e gli verrà chiesto di compiere una determinata azione. Esso per assolvere il suo compito dovrà scambiare informazioni (bits) con l'*hardware* vero e proprio a cui è assegnato (ad esempio protrebbe essere un modulo TDC). Questo modulo in particolare sarà interfacciato al PC tramite una interfaccia apposita (ad esempio un bus VME) che nell'ambito del DAS è descritto da una istanza della classe (figlia di `interface_c`) che descrive quella particolare interfaccia.

Questo scambio di informazioni avviene tramite i metodi dichiarati in `interface_c`: `read(void *)`, `write(void *)`, `generic(void *)`.

Nel momento in cui il modulo necessiterà di leggere qualche dato dall'*hardware* richiamerà il primo di questi metodi passandogli il puntatore ad una struttura che potrebbe contenere ad esempio indirizzo, *slot*, numero di byte, posizione in memoria dove sistemare i dati, etc., e l'operazione vera e propria verrà portata a termine quindi dall'interfaccia.

Viceversa, volendo scrivere qualcosa sull'elettronica (le soglie di un discriminatore, i valori da impostare su un I/O register, la scala di un TDC, etc.) sarà il secondo di questi metodi ad essere invocato, ed anche in questo caso la struttura identificata dal puntatore passato a `write()` conterrà le informazioni necessarie all'interfaccia per assolvere il suo compito.

Questo tipo di meccanismo è possibile grazie a quello che si chiama *dynamic casting*,

che consiste nel cambiare durante l'esecuzione del programma il tipo di dato a cui una variabile è associata (nel nostro caso il puntatore alla struttura dei parametri). Se, ad esempio, `Pointer` è un puntatore generico (di tipo `void *`) e `parameterStruct_t` è il nome della struttura, la sintassi del casting è la seguente:

```
parameterStruct_t * nuovoPuntatore = (parameterStruct_t*)Pointer;
```

Questa istruzione crea un nuovo puntatore di tipo `parameterStruct_t` e lo inizializza con il valore `Pointer` considerato a sua volta NON un puntatore generico ma anch'esso un puntatore a `parameterStruct_t`.

Con questo sistema la classe `interface_c` mantiene una notevole generalità, non essendo a conoscenza a priori del tipo di dati che necessita il dialogo tra un particolare modulo e la sua interfaccia. Diventa quindi molto semplice inserire nel DAS nuove interfacce e nuovi moduli, definendo la struttura dati necessaria ad ognuna di esse.

Il metodo `generic(void *)` è presente per consentire l'esecuzione di operazioni più generali che una pura lettura o scrittura, ed anch'esso sfrutta il meccanismo precedente per il passaggio dei dati.

L'ultimo metodo non ancora analizzato è quello chiamato `factory()`. Esso restituisce un puntatore ad un oggetto di tipo `interface_c`, è preceduto dalla parola `static`, ed il parametro passato è una stringa. La parola `static` assume diversi significati nel C++ a seconda del contesto in cui è applicata: nel caso specifico in cui preceda la dichiarazione di un metodo di una classe indica che il metodo è *comune a tutte le istanze della classe*. Non è quindi un metodo che si riferisce ad un oggetto specifico, ma ad una classe di oggetti nella sua totalità, ed in particolare non è necessario che venga creata una istanza della classe prima che questo metodo possa essere utilizzato.

Nel caso specifico della classe in esame questo metodo ricopre il ruolo di "fabbrica di oggetti", ovvero il processo di allocazione di nuove interfacce reali (`vme_sbs_c`, `cbd8210_c`, etc.) viene ricoperto dal metodo statico `factory()` appartenente alla classe astratta `interface_c`. La stringa passata come parametro è il nome dell'interfaccia che si intende istanziare. Si riporta un estratto della sua implementazione:

```
interface_c * interface_c::factory(string & type) {
    if(type == "vme_sbs") return new vme_sbs_c;
    if(type == "cbd8210") return new cbd8210_c;
    return NULL;
}
```

Come si vede, la funzione crea un nuovo oggetto del tipo richiesto e ne ritorna il puntatore che viene interpretato come puntatore a `interface_c`. Utilizzando questo

puntatore per richiamare i metodi virtuali descritti in precedenza, in metodi eseguiti in realtà saranno quelli implementati nelle classi `vme_sbs_c`, `cbd8210_c`, etc.

Con questa tecnica si mantiene la più totale indipendenza dei processi principali (*Acquisition*, *GUI*) dalla specificità delle interfacce. I processi principali sono al corrente soltanto dell'esistenza della classe `interface_c` (figura 4.1) e, dialogando con essa (tramite `factory()`, `read()`, etc.), sono in grado di trasmettere le informazioni date dall'utente per creare, impostare, gestire le interfacce reali.

Ciò che è necessario fare a questo punto, è la scrittura delle classi reali che ereditano la struttura di `interface_c`, ed ognuna di esse rappresenterà una vera e propria interfaccia *hardware*. Le prime due classi realizzate sono state quelle relative al bus VME (chiamata `vme_sbs_c`) interfacciato tramite la scheda SBS 616, e al bus CAMAC (`camac_c`) collegato al PC tramite una CBD8210 alloggiata in un crate VME.

Moduli

La struttura utilizzata per la parte relativa ai moduli ricalca abbastanza quella delle interfacce; di seguito si riporta un estratto della dichiarazione della classe `module_c`:

```
class module_c {
    /* ... dichiarazioni di variabili private ... */
public:
    /* ... dichiarazioni di alcuni metodi di servizio ... */
    module_c(interface_c *);
    virtual ~module_c();
    virtual int initialize() = 0;
    virtual int clear() = 0;
    virtual int action( void *structPointer );
    virtual int thereisdata();
    virtual int event( int ** bf );
    static module_c * factory(string & type, interface_c * interface);
};
```

La differenza sostanziale dei moduli rispetto alle interfacce è che essi sono sempre associati ad una interfaccia. È quindi necessario istanziare l'interfaccia a cui il modulo sarà connesso prima di istanziare il modulo stesso; al momento della creazione del modulo la connessione con l'interfaccia avviene passando al costruttore del modulo il puntatore all'interfaccia relativa.

Anche in questo caso si fa uso del metodo statico `factory()` che essendo incaricato della creazione dei moduli reali necessita, come informazione aggiuntiva rispetto al

tipo di modulo da creare, del puntatore all'interfaccia a cui il modulo va collegato. Il suo corpo sarà quindi qualcosa di questo tipo:

```
module_c * module_c::factory(string & type, interface_c * interface)
{
    if( type == "adc_v265" ) return new adc_v265_c( interface );
    if( type == "disc_v812" ) return new disc_v812_c( interface );
    /* ... */
    if( type == "statusA" ) return new statusA_c( interface );
    return NULL;
}
```

Come si vede il puntatore `interface` viene passato come parametro al costruttore dell'interfaccia reale (`adc_v265_c`, `statusA_c`, etc.).

La classe astratta `module_c` è dotata anch'essa di alcuni metodi virtuali di uso generico che consentono l'utilizzo dei moduli da parte dei processi utente. Alcuni di essi (`initialize()`, `clear()`) sono addetti all'impostazione dei parametri del modulo e alla sua pulizia, mentre i tre metodi rimanenti, i più importanti, sono quelli che consentono l'acquisizione dati vera e propria.

Quello chiamato `generic()` ha come parametro un puntatore generico (come alcuni metodi di `interface_c`), ed il suo scopo è quello di permettere al modulo di compiere durante la fase di acquisizione compiti generici stabiliti dai parametri passati dalla struttura puntata da `structPointer`. Ad esempio un I/O Register potrebbe ad certo punto dell'acquisizione cambiare i valori di uscita impostati in precedenza, oppure un status A potrebbe reimpostare alcune delle sue linee per modificare il funzionamento del *trigger*, e via dicendo.

L'acquisizione dati vera e propria passa attraverso i due metodi `thereisdata()` ed `event(int ** bf)`. La prima assume valore logico vero quando nel modulo è presente un dato valido, mentre la seconda restituisce in `bf` il puntatore ad un *buffer* di memoria che contiene un *array* di numeri interi. Il primo elemento dell'*array* contiene il numero di canali forniti dal modulo, e i seguenti contengono in sequenza i valori acquisiti dal modulo su ogni canale. Se l'operazione di lettura ha avuto successo `event(int ** bf)` assume valore logico falso (0), viceversa ritorna un codice di errore negativo specifico per il tipo di errore che si è verificato sull'interfaccia.

Come esempio ecco un riassunto dell'implementazione del metodo `event()` per la classe `adc_camac_c` (generico ADC su bus CAMAC):

```
int adc_camac_c::event( int ** bf )
{
    args.f = 2;          // struttura dei parametri da
```

```
args.ext = ext;      // passare all'interfaccia.
int sz = 0;
for ( unsigned int i = 0; i < ADC_CAMAC_NUM_CH; i++ )
{
    if ( channels[ i ] == ACTIVE )
    {
        sz++;
        args.sa = i;
        if ( status = get_interface()->generic( &args ) )
            return status;
        buffer[ i + 1 ] = args.d;
    }
}
buffer[ 0 ] = sz;
*bf = buffer;
return status;
}
```

Il primo passo è l'impostazione dei parametri da passare all'interfaccia tramite la struttura `args`. dopodiché viene inizializzata la dimensione del `buffer` ed inizia un *loop* su tutti i canali del modulo. Quando un canale risulta attivo viene letto il suo valore richiamando il metodo `generic()` dell'interfaccia a cui il modulo è connesso. In caso di errore (`status` diverso da zero) viene restituito il codice di errore fornito dall'interfaccia e l'esecuzione del metodo ha termine, viceversa il dato letto dall'interfaccia (restituito sempre tramite la struttura `args`) viene memorizzato nel `buffer`. Una volta terminati i canali, la dimensione del `buffer` viene salvata nel suo primo elemento (`buffer[0]`) ed il puntatore all'*array* di dati `bt` viene impostato alla base del `buffer`.

I metodi dichiarati virtuali in `module_c` e `interface_c` che non terminano con `= 0` sono dotati di una implementazione di *default* nella classe astratta; nel caso che il metodo non venga ridefinito nella classe figlia, questa implementazione è quella che viene eseguita in caso di chiamata del metodo.

Per fare un esempio, l'implementazione in `module_c` di `thereisdata()` restituisce sempre valore logico falso, e tale metodo non viene ridefinito nella classe `disc_v812_c` che descrive un discriminatore CFD (CAEN), che quindi non fornisce dati. Ogni volta che si richiama il metodo `thereisdata()` su questo discriminatore viene quindi eseguito il codice scritto per `module_c`, ottenendo uno zero logico.

Dalle classi astratte agli oggetti reali

Una volta stabilita la struttura di base fondata sulle due classi astratte appena descritte, non rimane altro da fare se non implementare le vere e proprie classi loro figlie, ognuna delle quali rappresenterà un particolare tipo di *hardware*.

La sintassi appare chiara dall'esempio successivo (relativo al TDC CAEN usato per i contatori del telescopio):

```
class tdc_v488a_c : public module_c {
    int buffer[ 2 * TDC_V488_NUM_CH + 1 ];
    /* ... altre variabili private ... */
public:
    int initialize();
    int clear();
    int thereisdata();
    int event( int ** bf );
    int set_mode( int );
    int set_start();
    /* ... altri metodi particolari ... */
    tdc_v488a_c( interface_c *, unsigned long baseAddress = DEF_ADDR );
    ~tdc_v488a_c();
};
```

La dichiarazione della classe è seguita da due punti, dalla parola **public** e dal nome della classe madre da cui si ereditano le proprietà; **public** in questo contesto sta ad indicare che l'eredità è pubblica, e che quindi tutti i metodi pubblici di `module_c` rimangono tali anche in `tdc_v488a_c`. Seguono le dichiarazioni del *buffer* per i dati e di alcune variabili private di uso interno, per poi arrivare alla dichiarazione dei metodi. Tra essi si ritrovano quelli ormai noti ereditati dalla madre, più alcuni metodi particolari di questo modulo che permettono l'impostazione corretta delle sue proprietà in fase di interfacciamento con l'utente (*common start/stop*, canali attivi, soglie, etc.).

In questo caso si nota il primo esempio di un meccanismo non ancora utilizzato in precedenza. I parametri passati al costruttore della classe sono due: il primo, come abbiamo già visto, è il puntatore all'interfaccia a cui il modulo è connesso, mentre il secondo rappresenta il *base address* assegnato al modulo. Non interessa tanto il significato di questo parametro quanto la sintassi che appare nell'esempio precedente, essendo il *base address* posto uguale ad una variabile del preprocessore (DEF_ADDR, definita in precedenza nello stesso file) nella *dichiarazione* stessa del costruttore. Il suo significato è il seguente: se il costruttore viene richiamato con

entrambi i parametri a `baseAddress` viene assegnato il valore fornito dal chiamante, viceversa viene assegnato ad esso il valore di `default DEF_ADDR`.

4.1.3 I processi di acquisizione e monitoring

Con questo insieme di strumenti a disposizione si può descrivere come è stata realizzata la parte centrale del lavoro, quella che gestisce l'acquisizione dati vera e propria. Essendo il processo in questione fortemente legato al *monitoring* dei dati acquisiti, le due fasi verranno discusse parallelamente.

È però importante precisare che manca a questo livello ancora un aspetto essenziale la cui descrizione è stata rimandata per chiarezza al paragrafo successivo: prima di iniziare l'acquisizione è fondamentale impostare i parametri che descrivono il *setup* utilizzato, e per fare ciò si utilizza un'apposita interfaccia utente (*GUI*, *Graphical User Interface*).

Il risultato della GUI è un file di testo che contiene tutte le informazioni relative all'*hardware* usato che sono necessarie al processo di acquisizione (d'ora in poi AP, *Acquisition Process*) per svolgere il suo compito; l'interpretazione di tale file è compito dei moduli e delle interfacce stesse, a cui viene passato quello che solitamente viene chiamato *parser* tramite l'*overloading* del costruttore. Il *parser* si occupa di impostare correttamente il funzionamento del modulo o dell'interfaccia durante l'inizializzazione del sistema.

La richiesta fondamentale ed irrinunciabile all'AP è la velocità. Per ottenerla esso non può perdere tempo a formattare, processare o visualizzare i dati, deve semplicemente prenderli e memorizzarli da qualche parte. Qualcun'altro (il *monitor*) si occuperà poi di fare con questi dati ciò che l'utente richiede.

L'integrazione e la comunicazione tra questi due processi (che hanno priorità totalmente diverse) è fondamentale; il comportamento di ognuno di essi dipende in realtà dallo stato in cui si trova l'altro. La soluzione utilizzata per implementare questo tipo di comunicazione è la condivisione tra i due processi di una zona di memoria.

Shared Memory

Il sistema operativo utilizzato permette che due o più processi condividano alcune locazioni di memoria; il *kernel* (che si occupa di gestire il *multiprocessing* del sistema) si occupa di regolare l'accesso alla memoria condivisa nascondendo all'utente l'implementazione a basso livello e rendendo trasparente il suo utilizzo.

Nel caso in esame, un blocco di memoria viene allocato dal *monitor* e viene posto in condivisione con l'AP. Tramite essa avviene la comunicazione tra i due, nella quale il *monitor* funge da *master*.

La gestione diretta della memoria condivisa è mascherata ai due processi tramite

la creazione di una classe (chiamata **Buffer**) che si occupa della sua organizzazione funzionale.

La dichiarazione della classe **Buffer** e l'implementazione dei costruttori sono riportate in appendice A. Essa è dotata di due costruttori diversi, il primo utilizzato dal *monitor* ed il secondo dall'AP. Ognuno di questi due processi istanzia infatti un oggetto di tipo **Buffer**, ma soltanto il primo alloca la memoria, mentre il secondo si aggancia al segmento allocato dal primo.

Gli argomenti che il *monitor* passa al costruttore sono appunto una stringa che contiene il nome del file in cui salvare i dati e la dimensione (in kiloByte) del *buffer* di memoria da allocare, mentre al secondo costruttore viene passato dall'AP un numero identificativo del segmento di memoria condivisa fornito dal sistema operativo; ogni processo che è a conoscenza di questo numero è in grado di agganciarsi al segmento di memoria.

Scorrendo il codice relativo al costruttore usato dal *monitor* si vede come queste operazioni vengono effettivamente realizzate: la prima istruzione apre uno *stream* di uscita connesso al file sul quale verranno riversati i dati. dopodiché si trova una chiamata alla funzione di sistema `shmget()`

```
segment_id = shmget( IPC_PRIVATE, bfSize * ONE_KBYTE,
                    IPC_CREAT | IPC_EXCL | S_IRUSR | S_IWUSR );
```

che alloca la memoria richiesta utilizzando le impostazioni determinate dai vari *flags* (non è importante il loro significato ai fini della comprensione del meccanismo) e restituisce l'identificativo da usare per agganciare il segmento. Subito dopo, infatti, il segmento viene agganciato dal processo stesso con la chiamata a `shmat` che restituisce il puntatore alla base della memoria allocata in precedenza

```
smPointer = (char*)shmat( segment_id, 0, 0 );
```

e con la tecnica del *casting* dinamico già utilizzata in passato essa viene organizzata in una struttura di tipo `runInfo_t` (alla base) più il *buffer* vero e proprio che conterrà i dati acquisiti (*array* di interi):

```
runInfo = (runInfo_t*)smPointer;
data = (int*)&smPointer[ BASE_DATA_POINTER ];
```

Le due istruzioni seguenti interrogano il sistema operativo che fornisce in una struttura apposita tutte le informazioni relative al segmento di memoria identificato da `segment_id`, mentre le istruzioni successive inizializzano le variabili locali e quelle condivise con l'AP tramite la struttura `runInfo_t`.

Il costruttore utilizzato dall'AP accetta un solo parametro: il `segment_id` fornito in

precedenza al *monitor* da `shmget()`. L'AP conosce l'ID fornito dal sistema operativo al *monitor* perchè è quest'ultimo a creare l'AP stesso tramite il procedimento chiamato *forking*. Il codice relativo è il seguente:

```
pid_t acquisitionProcessPid = fork();
char * acquisitionProcessName = "acq";
char * acquisitionProcessArgs[] = {
    "acq",
    segment_id,
    NULL
};
if ( acquisitionProcessPid == 0 )
    execvp ( acquisitionProcessName, acquisitionProcessArgs );
```

Il *monitor* chiede al sistema operativo di creare un secondo processo identico a se stesso (chiamata a `fork()`). Il valore restituito (di tipo `pid_t`) è il *pid* del processo figlio all'interno del processo padre, mentre è zero per il processo appena creato. In questo caso con la chiamata alla funzione `execvp()` viene lanciata l'esecuzione del processo chiamato *acq* (l'AP) passandogli come parametri il suo nome e un *array* di stringhe il cui ultimo valore è un puntatore nullo. In questo caso l'unico parametro passato è proprio l'ID del segmento di memoria condivisa allocato in precedenza.

La presa dati

Dopo aver istanziato un **Buffer** inizializzandolo con l'identificatore fornito dal *monitor*, l'AP interpreta il file che contiene le informazioni sul *setup* e inizializza tramite esso interfacce e moduli. Verificata la coerenza delle informazioni acquisite inizia la fase di acquisizione vera e propria, il cui diagramma di flusso è riportato in figura 4.2 (per il codice di veda sempre l'appendice A).

Come si può vedere esso è suddiviso in tre blocchi:

- *Start up and controls*: si occupa di inizializzare la fase di acquisizione e verificare la correttezza delle impostazioni (ad esempio che la dimensione del *buffer* sia sufficiente a contenere almeno un evento);
- *Controls every event*: gestisce la parte di controlli che vengono fatti durante l'acquisizione dati vera e propria;
- *Event acquisition*: porta a termine l'acquisizione vera e propria dei singoli eventi.

Ogni volta che si verifica una condizione che porterebbe fuori dall'acquisizione, l'unica cosa che l'AP fa è l'invio di un segnale al *monitor*. Il modo per fare questo è l'utilizzo di due funzioni di sistema: `getppid()` e `kill()`.

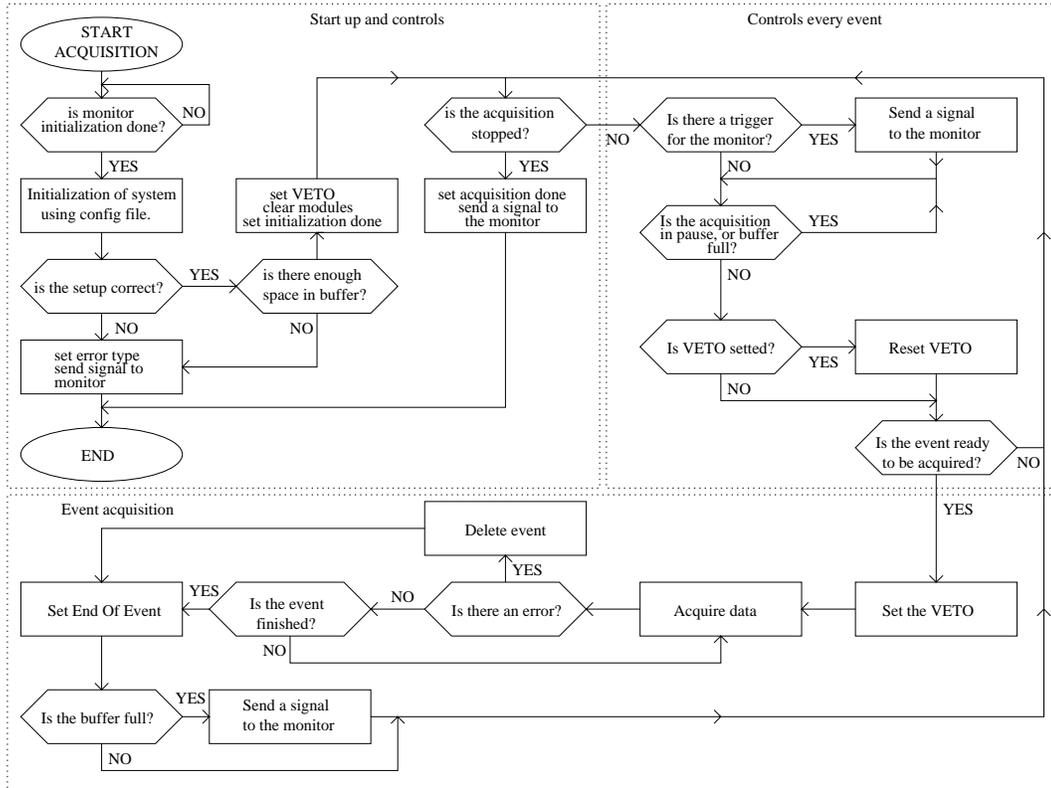


Figura 4.2: Diagramma di flusso del processo di acquisizione dati (AP).

```

pid_t parentPid = getppid();
kill( parentPid, SIGALRM );
    
```

La prima istruzione restituisce il PID del processo padre (che nel caso dell'AP è sempre il *monitor*, essendo lui che con il *forking* lancia l'acquisizione), mentre la seconda gli invia il segnale di sistema **SIGALRM**.

Il *monitor*, a sua volta, si preoccupa di intercettare tutti i segnali che si aspetta dall'AP utilizzando il seguente meccanismo:

```

void update(int sig);
int main(/*...*/)
{
/*...*/
    struct sigaction sa;
    
```

```
memset( &sa, 0 ,sizeof(sa) );
sa.sa_handler = &update;
sigaction( SIGALRM, &sa, NULL );
/*...*/
}
void update(int sig)
{
/* istruzioni da eseguire quando viene ricevuto il segnale SIGALRM */
}
```

Viene prima dichiarata una funzione che verrà eseguita all'atto della ricezione del segnale (`update()`); dopodiché viene dichiarata (e azzerata con `memset()`) una struttura che descrive le azioni da eseguire in caso di ricezione del segnale, e nel caso specifico viene inizializzata con un puntatore alla funzione `update()` (detta generalmente *signal handler*). In fine con la chiamata a `sigaction()` viene intercettato il segnale `SIGALRM`, e da questo punto in poi ad ogni ricezione del segnale da parte del *monitor* corrisponde l'esecuzione della funzione `update()`.

A quest'ultima è riservato il compito di gestire il salvataggio dei dati su disco e l'aggiornamento degli istogrammi visualizzati dal *monitor*.

4.2 GUI e slow control

Per creare il file di descrizione del *setup* necessario all'AP è stata sviluppata un'interfaccia utente apposita. Allo scopo sono state utilizzate le librerie grafiche QT create dalla *trolltech* (www.trolltech.com) e distribuite gratuitamente per lo sviluppo di software *open source*. Una delle loro caratteristiche fondamentali è che sono multiplatforma, ed è quindi possibile compilare lo stesso codice su diversi sistemi operativi.

La struttura su cui è fondata la GUI ricalca fedelmente quella già vista per l'AP (figura 4.1). Essa ha a disposizione un numero virtualmente infinito di classi grafiche ognuna delle quali rappresenta il corrispondente modulo o interfaccia per cui è stata creata, alle quali accede attraverso una generica classe astratta chiamata `objectgui`. Le due figlie dirette di questa classe (ancora astratte ma più specifiche) sono `interfacegui` e `modulegui`, che si ramificano poi nelle classi reali che rappresentano graficamente le singole componenti *hardware*.

Ognuna di queste classi ha a disposizione un puntatore all'oggetto che rappresenta (freccie verdi nella figura 4.1); tramite esso avviene la modifica in *real time* delle impostazioni dell'oggetto stesso.

Una volta raggiunto il *setup* richiesto ogni oggetto grafico si preoccupa di salvare sull'apposito file le informazioni necessarie secondo una sintassi prestabilita e nota

alle classi di lettura. Il costruttore di queste ultime (o meglio, una seconda versione di esso creata per mezzo dell'*overloading*) interpreta poi questo stesso file durante la fase di impostazione che precede l'acquisizione e imposta l'*hardware* in base alle informazioni raccolte.

Durante la fase di acquisizione stessa gestita dal'AP, la GUI agisce parallelamente come *slow control* del sistema, permettendo di modificare dinamicamente i parametri statici del *setup* (soglie, tensioni, etc.).

4.3 Analisi dell'efficienza

Il primo vero *test* sul sistema di acquisizione si è svolto al CERN sul *setup* che ho assemblato per il *test beam* che si svolgerà dal 14 al 19 ottobre 2002.

Lo scopo del *test* per quanto riguarda il TOF è lo studio delle caratteristiche di ampiezza d'impulso dei contatori; per fare questo essi verranno irradiati con nuclei leggeri ($H - Fe$) sul fascio SPS del CERN.

Il *test beam* vedrà integrati con i due contatori del TOF (d'ora in poi saranno indicati con miniTOF), un prototipo del RICH (miniRICH) ed uno del TRACKER (miniTRACKER). Due scintillatori a monte dell'*hardware* intercetteranno il fascio e forniranno il segnale di *trigger*, che sarà distribuito all'elettronica dei tre sottosistemi (miniTOF, miniRICH e miniTRACKER) da un controllo centrale che terrà conto dei segnali di *busy* da essi forniti.

Da esso verrà generato in corrispondenza di ogni segnale di *trigger* un numero binario progressivo che sarà diffuso tramite 24 linee ECL (*eventID*); la lettura di questo numero effettuata dai tre apparati consentirà la sincronizzazione degli eventi acquisiti durante l'analisi *offline*.

Per quanto riguarda il miniTOF il sistema di acquisizione è basato sul DAS da me sviluppato, ed il suo compito sarà la lettura dei segnali di carica e di tempo forniti dai due contatori utilizzati per il *test*. Dato che ogni contatore fornisce 4 segnali di carica (somma analogica di anodi e dinodi di ogni lato) e 4 di tempo (gli stessi segnali di carica ma discriminati), la lettura riguarda 8 canali di ADC e 8 canali di TDC, più naturalmente l'*eventID*.

Il *setup* di acquisizione del miniTOF quindi è sostanzialmente una catena formata da:

- *fan out*, che mette a disposizione i segnali di anodo e dinodo sdoppiati;
- ADC CAMAC 2249W, per l'acquisizione della carica di anodi e dinodi;
- discriminatore NIM, per la formazione dei segnali NIM relativi ad anodi e dinodi;

- TDC CAMAC 2228A, per l'acquisizione dei tempi relativi ad anodi e dinodi;
- *latch* CAMAC 4448, per l'acquisizione dell'*eventID*;
- catene di ritardo, per la sincronizzazione dei segnali.

La logica di gestione (creazione del *gate* dell'ADC, *busy*, *clear*, etc.) è fatta con moduli NIM, mentre l'acquisizione (temporizzata da un apposito segnale fornito ad uno STATUS A) è fatta attraverso un bus VME con interfaccia CBD8210 verso il bus CAMAC e SBS616 verso il bus PCI del PC.

Per la verifica del funzionamento del *setup* è stato creato temporaneamente un *trigger* locale facendo la coincidenza dei due lati di un contatore in modo da poter utilizzare i raggi cosmici.

Una volta verificato il corretto funzionamento del sistema sono stati effettuati alcuni *test a rate* costante per esaminare le prestazioni fornite dal DAS. In particolare il *trigger* è stato sostituito da un impulsatore costruito con una *dual timer* configurata come oscillatore; dopodiché sono state effettuate diverse raccolte di dati della durata di 60 s per *rate* compresi tra 10 Hz e 1kHz. Tramite uno *scaler* è stato possibile acquisire il numero di segnali generati dall'impulsatore ed il numero di eventi acquisiti dal DAS. Quindi i dati ottenuti sono relativi alla lettura di 4 moduli CAMAC:

- 1 ADC, 8 canali;
- 1 TDC, 8 canali;
- 1 LATCH, 3 canali;
- 1 SCALER, 2 canali.

I dati ottenuti sono quelli riportati in appendice B.

Si nota che c'è un certo numero di eventi (funzione del *rate*) che viene sistematicamente perso all'inizio di ogni presa dati; l'effetto è dovuto al tempo che intercorre tra l'azzeramento dello *scaler* durante l'inizializzazione e la partenza della fase di acquisizione. Questa perdita non è quindi da imputarsi all'efficienza del processo di acquisizione.

Il numero eventi persi a causa dell'efficienza del DAS (N_{lost}) è quindi la differenza tra il totale degli eventi persi (N_{tl}) e questo *offset* iniziale (N_{delay}). Dal rapporto tra il numero ottenuto con il numero di fronti prodotti dall'impulsatore ($N_{trigger}$) si ottiene la percentuale di eventi persi in funzione del *rate*. L'efficienza è data quindi da:

$$\epsilon = 1 - \frac{N_{tl} - N_{delay}}{N_{trigger}} \quad (4.1)$$

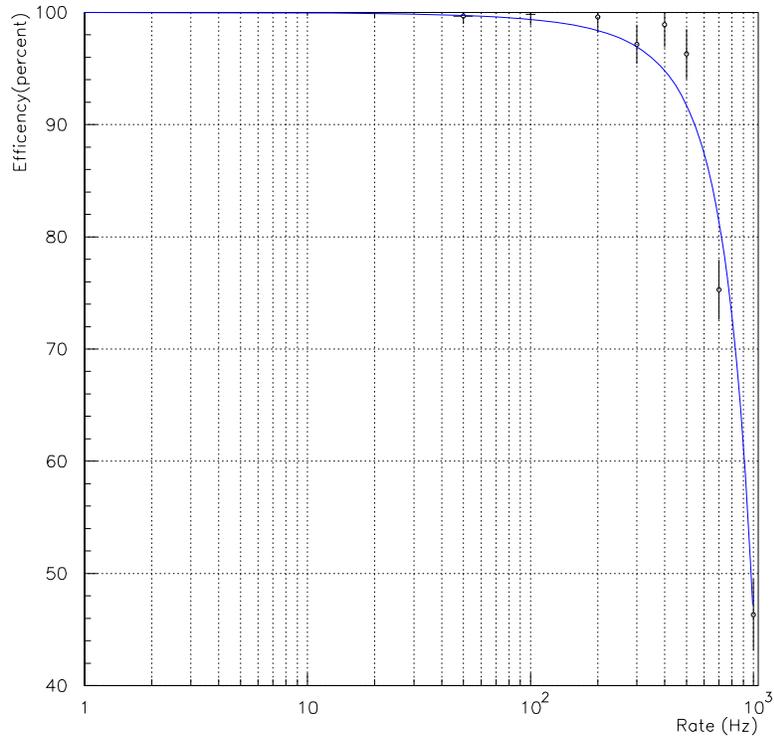


Figura 4.3: Curva di efficienza del DAS.

In figura 4.3 è riportato il grafico dell'efficienza in funzione del *rate* fittato con una curva esponenziale. Fino a 500 *Hz* l'efficienza è maggiore del 95%, e diventa esattamente unitaria a 20 *Hz*.

4.4 Utilizzo del software

È sicuramente utile, al fine di chiarire i motivi che hanno portato a fare le scelte sulla struttura del DAS discusse in precedenza, descrivere brevemente il corretto utilizzo del *software*.

La fase di creazione vera e propria del *setup* procede solitamente in parallelo alla creazione del file di impostazioni tramite la GUI. Man mano che si inseriscono ADC, TDC, scaler, discriminatori e via dicendo, gli oggetti grafici relativi vengono inseriti

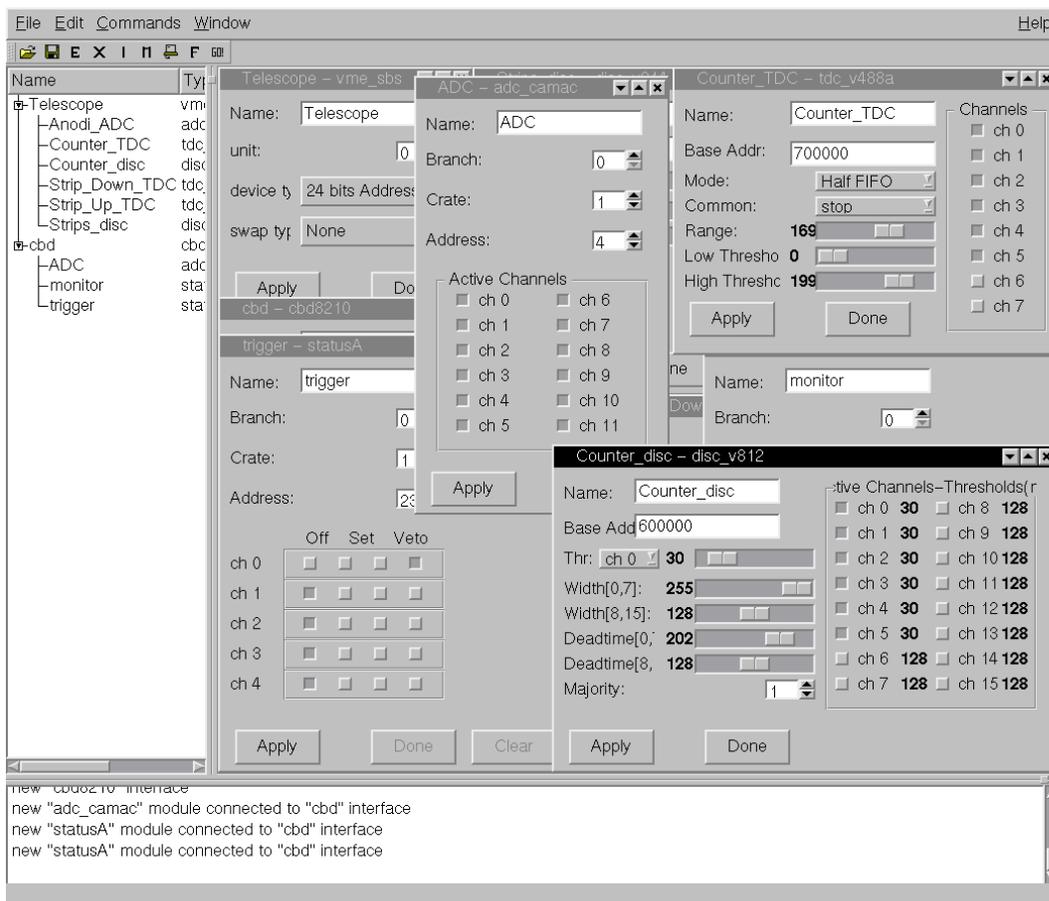


Figura 4.4: Schermata della GUI del DAS.

(con gli appositi comandi *Add module* e *Add interface*) nella GUI, che presenterà una finestra specifica per ogni oggetto dalla quale sarà possibile modificarne i parametri (soglie, scale, canali attivi, modalità di funzionamento, etc.). In figura 4.4 si vede un esempio di come si presenta il DAS in questa fase.

Una volta soddisfatti del sistema ottenuto, si passa alla fase di impostazione dei *flags*; ad ogni modulo possono essere assegnati uno o più dei seguenti compiti:

- *trigger*: il modulo fa parte del *trigger* del sistema, di conseguenza viene imposto il veto su di esso durante l'acquisizione dell'evento per evitare che altri segnali di *trigger* interferiscano con esso;

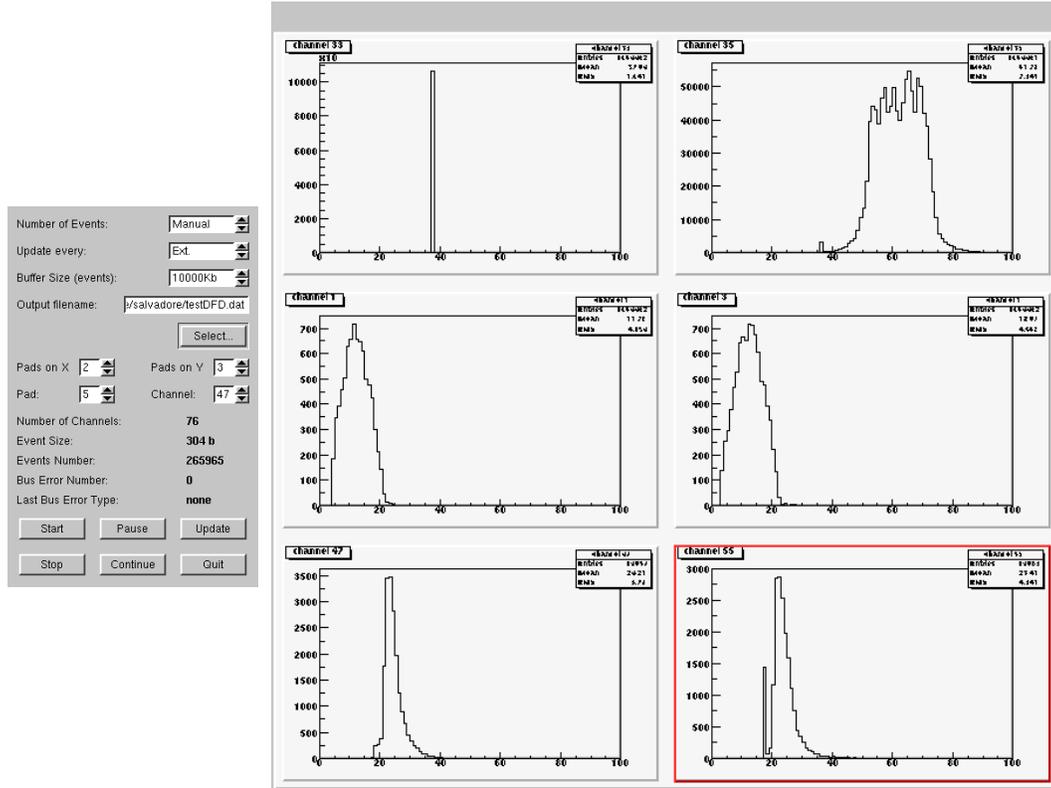


Figura 4.5: Schermata del monitor *online* del DAS.

- *polling*: i moduli dotati di questo *flag* vengono interpellati con il metodo `thereisdata()`. Soltanto quando tutti i moduli in *polling* presentano un dato pronto inizia l'acquisizione dell'evento (tipicamente si usa il *polling* sullo STATUS A);
- *acquire*: questi moduli forniscono dati, quindi l'acquisizione dell'evento consiste nel raccogliere i dati presenti su tutti i moduli dotati di questo *flag*;
- *monitor*: i moduli impostati come *monitor* sono simili ai moduli di *polling*, ma in caso di dato pronto viene inviato un segnale al *monitor* del DAS che invita a salvare i dati ed aggiornare gli istogrammi. Uso tipico di questo *flag* è su un canale di uno STATUS A che riceve il segnale *End of Spill*. In questo modo l'*update* degli istogrammi e il salvataggio dei dati avviene nell'*interspill*, evitando di rallentare l'acquisizione e ridurre il *rate*.

Una volta assegnati i *flags* si lancia il processo di *monitor* che permette di gestire la fase di acquisizione dati vera e propria (figura 4.5). Tra i vari parametri che è possibile impostare ci sono:

- numero di eventi da acquisire;
- l'intervallo di tempo che separa due aggiornamenti degli istogrammi e salvataggi dei dati;
- dimensione del *buffer* in kilobyte;
- il nome del file in cui salvare i dati;
- il numero di istogrammi da visualizzare e a quali canali dell'evento essi sono associati.

Mentre le quantità monitorate dal processo sono:

- numero di canali dell'evento;
- dimensione dell'evento in kilobyte;
- numero di eventi acquisiti;
- numero di errori avvenuti durante l'acquisizione;
- tipo dell'ultimo errore avvenuto.

Una volta che i parametri sono impostati correttamente l'acquisizione ha inizio e viene aperta una finestra che presenta gli istogrammi relativi ai dati che vengono acquisiti. Questa fase ha termine una volta raggiunto il numero di eventi richiesto o a causa di una interruzione manuale da parte dell'utente.

Capitolo 5

Il DAQ di AMS

5.1 Sistema di acquisizione dati di AMS-02

AMS-02 necessita di un sistema elettronico di controllo e di raccolta dati molto complesso; l'ambiente in cui si troverà ad operare, inoltre, rende praticamente impossibile qualsiasi intervento diretto, e presenta un flusso di radiazione che può incidere pesantemente sul funzionamento dell'elettronica.

Il progetto di quest'ultima deve quindi seguire una serie di vincoli che la rendano adatta ad operare in un ambiente così ostile per tutta la durata della missione. Uno dei parametri fondamentali a cui non si può rinunciare è una forte *rindondanza* in tutti quei punti critici nei quali un qualsiasi malfunzionamento potrebbe compromettere il successo della missione.

5.1.1 Resistenza alla radiazione

Il flusso di radiazione a cui saranno sottoposti i componenti elettronici di AMS preoccupa non tanto per la dose cumulativa che verrà assorbita, ma più per i *Single Event Effects* (SEE). Mentre gli effetti cumulativi provocano un degrado graduale durante l'intera vita dell'elettronica esposta, i SEE sono dovuti all'energia depositata in un componente da una singola particella. Questo tipo di effetto può dunque avvenire in ogni momento, con una probabilità espressa in funzione della sezione d'urto della particella incidente nel materiale di cui il componente esposto a radiazione è costruito.

Gli effetti di evento singolo si classificano solitamente in tre categorie:

- *Transienti*: sono segnali spuri che si propagano nel circuito;
- *Statici*: avviene la sovrascrittura di un dato immagazzinato;

- *Distruttivi*: provocano la rottura irreversibile di un componente.

I transienti sono errori tipici dell'elettronica analogica e della logica combinatoria, e possono tramutarsi in errori statici se l'impulso spurio si propaga fino a impostare un *latch*.

Gli errori statici causano l'inversione di un bit memorizzato nel dispositivo; un certo numero di questi errori può essere tollerato se il sistema utilizza metodi di identificazione ed eventualmente correzione di errori (parità, CRC, etc.).

Gli effetti distruttivi provocano il danneggiamento definitivo di un componente; in alcuni casi è possibile identificare l'effetto nella sua fase iniziale ed un veloce intervento permette il salvataggio del sistema.

Single Event Upset (SEU)

Avviene quando una particella ionizzante passa in prossimità di una giunzione pn. Nell'attraversare il materiale la particella perde energia attraverso il processo di *scattering* multiplo ionizzando alcuni degli atomi del reticolo. Le coppie elettrone-lacuna così create possono, invece che ricombinarsi, allontanarsi per effetto di un campo elettrico, e dare origine ad un impulso di corrente; in certi casi (SDRAM) l'impulso generato può essere sufficiente a provocare la transizione di un bit allo stato logico opposto producendo un errore statico (non distruttivo).

L'effetto si produce soltanto quando il passaggio della particella avviene in zone *attive* del dispositivo; inoltre l'energia persa deve essere sufficiente a provocare la transizione di stato. L'effetto è quindi solitamente studiato utilizzando le due quantità V_S (volume sensibile) ed E_C (energia critica).

Single Event Burnout (SEBO)

I DMOS e i BJT di potenza sono soggetti solitamente a questo effetto, che si produce solitamente quando il componente si trova nello stato OFF; in questa situazione esso subisce una elevata differenza di potenziale *drain-source* (emettitore-collettore). Uno ione (la carica rilasciata necessaria a produrre SEBO è maggiore che nel SEU) attraversando il componente produce un impulso di corrente che viene amplificato a causa del *feedback* attivo. La valanga che si produce brucia il componente.

Single Event Gate Rupture (SEGR)

I dispositivi sensibili a SEGR sono i MOSFET di potenza e i CMOS. Anche questo effetto si produce nello stato OFF, quando il campo elettrico presente nel dielettrico (ossido di silicio, SiO_2) che separa il *gate* dal *drain* è elevato. In corrispondenza al

passaggio di uno ione, le cariche libere prodotte derivano producendo un incremento del campo, che in certi casi può superare il valore critico che provoca la rottura del dielettrico; il passaggio di corrente che ne deriva provoca un innalzamento di temperatura che brucia velocemente il componente.

Single Event Latchup (SEL)

Sono i dispositivi CMOS ad essere interessati da questo fenomeno; un semplice modello per descrivere il meccanismo di *latchup* è costituito da due BJT collegati in modo tale che la corrente di collettore di ognuno di essi pilota la corrente di base dell'altro. Se a causa di un transiente (dovuto ad esempio al passaggio di uno ione) avviene un incremento della corrente di collettore di uno dei due transistor, l'aumento si ripercuote sulla corrente di base dell'altro con conseguente aumento della sua corrente di collettore.

Il processo così innescato porta la corrente a divergere velocemente fino a produrre la rottura del componente, a meno che non venga repentinamente tolta l'alimentazione al dispositivo.

5.1.2 Rindondanza

Per garantire una sufficiente affidabilità di funzionamento per un apparato così complesso in un ambiente difficile come lo spazio è necessario prendere notevoli precauzioni. Si è visto in precedenza che ogni contatore ad esempio è dotato di due fotomoltiplicatori per lato, in modo da avere segnale anche in caso di rottura di uno dei due.

Oltre a questo, ci sono altri sistemi di rindondanza in ogni punto dell'apparato. Ogni parte dell'elettronica ad esempio è formata da due sistemi assolutamente identici, uno attivo (*Hot*) ed uno di riserva (*Cold*) che sostituisce la parte *Hot* in caso di rottura.

5.1.3 Schema generale

La figura 5.1 descrive il modo in cui è strutturata l'elettronica di AMS: la configurazione ad albero permette di isolare i singoli compiti e renderli indipendenti gli uni dagli altri. Ad ogni *subdetector*, infatti, è assegnato il compito di fornire i dati alla catena generale seguendo un formato predefinito, ma ognuno di essi è libero di adottare soluzioni tecniche differenti dagli altri.

AMS-02 è fornito di due **MDC** (*Main DAQ Computer*) basati su *PowerPC 750* (qualificati per lo spazio) interfacciati alla ISS che gestiscono il controllo e l'acquisizione di tutto il sistema. Ad essi è assegnato il ruolo di *master*: sono situati al vertice della piramide che raccoglie i dati forniti dai *subdetectors* e trasmettono i

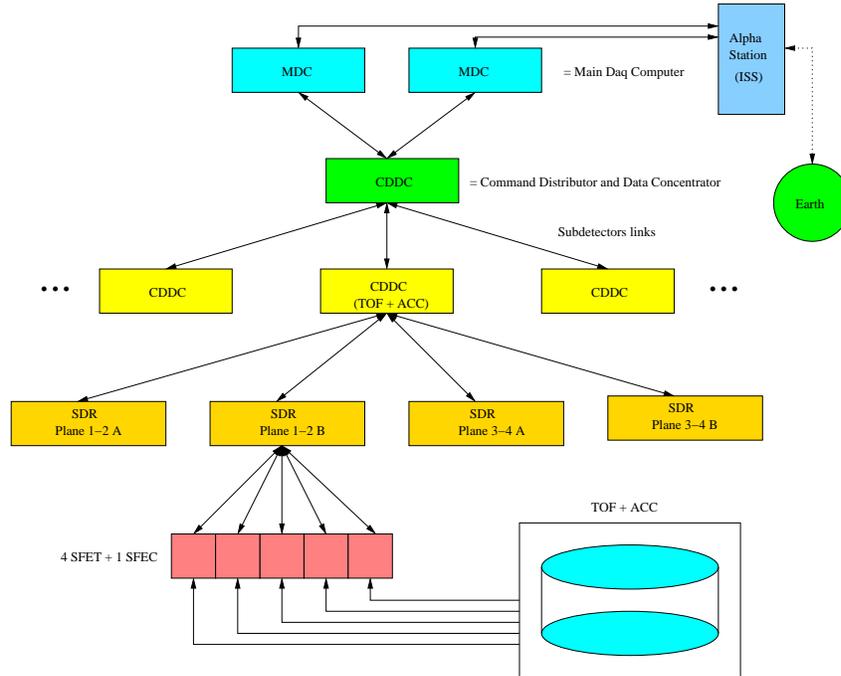


Figura 5.1: Schema a blocchi del sistema di acquisizione di AMS-02.

comandi verso questi ultimi.

Appena sotto agli **MDC** si trova un modulo chiamato **CDDC** (*Command Distributor and Data Concentrator*) che si occupa appunto della diramazione degli ordini impartiti dal *master* e della raccolta del flusso di dati. La comunicazione tra i due avviene tramite un collegamento battezzato con il nome *AMSwire*; esso è composto da due linee differenziali di tipo LVDS, una per la trasmissione (TX) e una per la ricezione (RX), ed utilizza un protocollo riadattato da uno *standard* industriale (MIL-STD-1553).

Il compito di questo primo **CDDC** è di raccogliere i flussi dati e distribuire i comandi verso i 13 moduli ad esso subordinati, che sono funzionalmente identici ad esso. Questi 13 **CDDC** che si trovano al livello inferiore sono direttamente collegati all'elettronica dei singoli *subdetector*.

In questa configurazione ogni modulo è sempre *master* di quelli ad esso subordinati. Di conseguenza nessuno dei modulo può mai assumere il controllo della parte della catena superiore ad esso. L'unico modo che hanno i *subdetector* di influire sulla catena generale è il segnale *busy* che informa il *trigger* quando l'apparato è occupato in qualche operazione.

Lo *slow control* del sistema (settaggio tensioni, controllo temperature, etc.) segue uno schema generale che ricalca quello relativo al *data acquisition*; in questo caso però le trasmissioni avvengono tramite lo standard CANbus.

Da questo livello in poi l'elettronica dipende fortemente dal tipo di *subdetector* a cui è dedicata. Nel caso del TOF le richieste sono:

- 192 canali di HV (*High Voltage*) corrispondenti ai 192 fototubi installati nei 48 contatori (12 contatori per piano, 4 fototubi per contatore);
- 96 canali di lettura di tempo corrispondenti ai 96 lati dei 48 contatori (2 lati per contatore, in ognuno dei quali i segnali di anodo e dinodo sono sommati);
- 96 canali di lettura della carica anodica;
- 96 canali di lettura della carica dinodica.

La scelta fatta è stata quella di installare quattro *crates* per il TOF, contenenti i seguenti moduli:

- 3 **SFET** (*Scintillators Frontend Electronics - Time*), ognuna delle quali sarà adibita alla lettura dei tempi di 8 canali anodici e alla produzione dei segnali necessari al *trigger*;
- 1 **SFEC** (*Scintillators Frontend Electronics - Charge*), dedicata all'integrazione della carica raccolta da 24 canali dinodici;
- 1 **SFET** dedicata alla raccolta dei segnali prodotti dalle anticoincidenze;
- 1 **SDR** (*Scintillators Data Readout*), il cui scopo è la raccolta dei dati forniti dalle **SFE[X]**;
- 4 *HV generators* per la distribuzione delle alte tensioni;
- 1 **USCM** (*Universal Slow Control Module*), dedicata alla distribuzione dei segnali di *slow control*.

Ogni modulo è, come si è accennato in precedenza, rindondante. È infatti presente nelle due versioni *Hot* (solitamente in funzione) e *Cold* (di riserva).

La **SDR** ricalca la struttura dei moduli **CDDC**: funge sempre da *master* per i moduli ad esso subordinati (**SFE[X]**), mentre ricopre il ruolo di *slave* nei confronti della **CDDC** a cui è collegato (sempre tramite AMSwire). La differenza principale nella sua struttura è che i *link* verso i moduli *slave* sono sostituiti da un *sequencer* che legge serialmente i dati da 5 canali.

Nel paragrafo successivo verrà data una descrizione dettagliata dei singoli moduli che compongono il sistema.

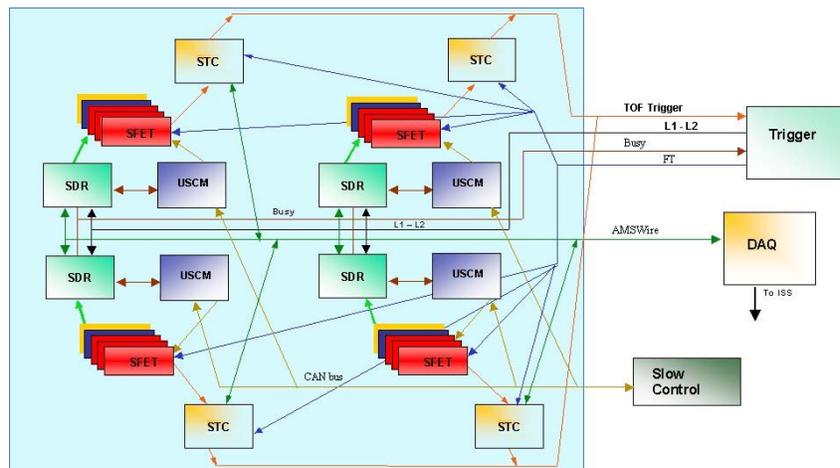


Figura 5.2: Schema a blocchi dettagliato del sistema di acquisizione del TOF.

5.2 Definizione del sistema di acquisizione del TOF

In figura 5.2 è riportato lo schema a blocchi che descrive il funzionamento della parte di acquisizione dati relativa al TOF.

Il significato ed il tipo delle diverse linee in funzione del colore è:

- *verde scuro*: collegamento AMSWire per il trasporto del flusso di dati e comandi;
- *verde chiaro*: collegamento CANbus per il trasporto delle informazioni di *slow control*;
- *arancione*: coincidenze prodotte dai segnali delle **SFET** per la generazione del *fast trigger*;
- *blu*: segnale di *fast trigger*;
- *marrone*: segnale di *busy* per il veto del *trigger*;
- *nero*: *trigger* di primo livello.

La parte relativa alla produzione dei segnali di *fast trigger* (**STC**, *Scintillator Trigger Coincidence*) verrà integrata sulla **SFET**.

Prima di analizzare in dettaglio le varie componenti dell'elettronica del TOF è necessario introdurre e descrivere brevemente il sistema utilizzato per lo sviluppo ed il test dei moduli stessi.

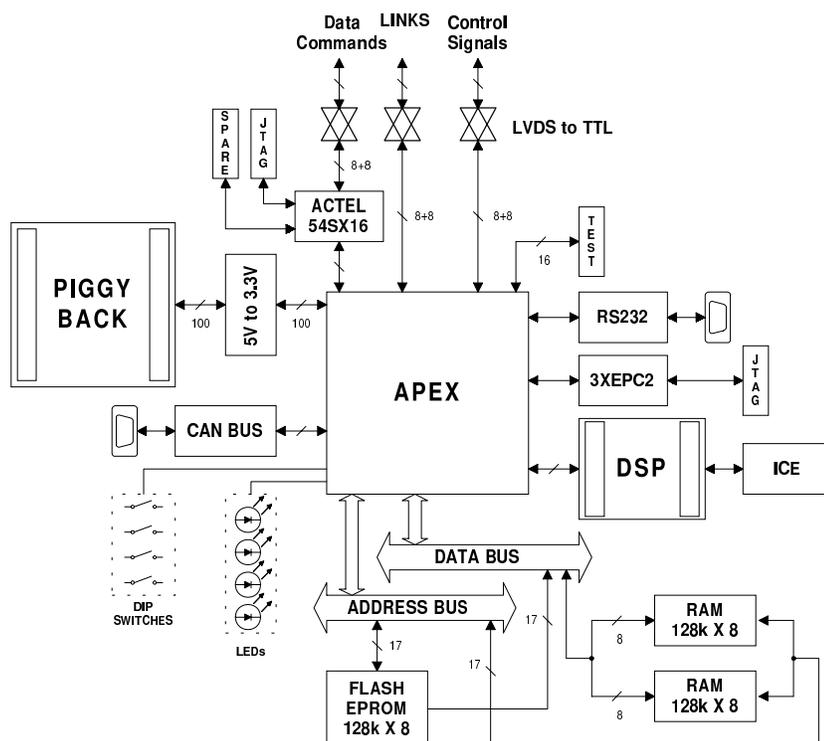


Figura 5.3: Schema a blocchi della s9007.

5.2.1 Scheda di sviluppo S9007

Per rendere efficiente lo sviluppo dei componenti elettronici che fanno parte delle catene di *acquisition* e *slow control* di AMS-02, è stata progettata, in collaborazione con la CAEN, una scheda elettronica (S9007, detta *Guido Board*) costruita con componenti di volo (figura 5.3).

La Guido Board è dotata di una FPGA APEX 20K (prodotta dalla ALTERA) interfacciata con 16 banchi di memoria RAM da 128KB, 8 banchi di memoria FLASH da 128KB, un DSP (ADSP-2187L della *Analog Devices*), un driver per interfaccia seriale RS232, uno per il *bus* JTAG, uno per il CANbus, 16 linee di test (per l'analizzatore di stati logici), e 16 linee di Controllo, 16 links I/O, e 16 linee *data* e *control* di tipo LVDS. Sono inoltre presenti 4 DIP Switch e 4 diodi LED per il *debugging* e la possibilità di aggiungere schede aggiuntive (*Piggy Back*) sia connesse all'FPGA che al DSP.

L'FPGA usata è compatibile con quelle prodotte dalla ACTEL (famiglia SX) che

verranno utilizzate per le schede di volo; queste ultime infatti, diversamente dalle ALTERA, sono basate su tecnologia *fused* che le rende tolleranti alle radiazioni. Esse possono essere programmate una sola volta, ed è per questo che per lo sviluppo vengono utilizzati i *chips* ALTERA, meno tolleranti alle radiazioni ma riprogrammabili.

Questa complessa scheda contiene praticamente tutti gli elementi che saranno presenti nell'elettronica dell'apparato, e permettere di simulare, testare e controllare ogni elemento della catena elettronica di AMS-02.

L'FPGA utilizzata (versione EP20K400) mette a disposizione tipicamente 400'000 gates con un massimo di 502 I/O pins, rendendo possibile l'implementazione al suo interno di complessi schemi funzionali.

5.2.2 SFET

La **SFET** è la scheda che si occupa digitalizzazione dei segnali forniti dai contatori. Ognuna di esse è in grado di leggere 8 canali, e da ogni canale si ottengono diverse informazioni:

- tre segnali che corrispondono al superamento da parte del segnale anodico di tre soglie:
 - *low* detta $Z \geq 1$, tipicamente dell'ordine di 30 *mV*
 - *high* detta $Z > 1$, tipicamente dell'ordine di 150 *mV*
 - *super high* detta $Z > 2$, proporzionale alla soglia *high*

questi tre segnali vengono utilizzati dalla logica di *trigger* per la produzione del *fast trigger* (*FT*) e del *trigger level 1*;

- una misura del tempo che intercorre tra la ricezione del segnale anodico (in corrispondenza del superamento della soglia *low*) e la ricezione del *FT* moltiplicato per un fattore 40 (*time expansion*) per ottenere una risoluzione di 25 *ps*;
- una misura dello stesso tempo ma non espanso con dinamicità di 6 μs (risoluzione 1 *ns*) che funge da *history*;
- una misura del tempo sopra soglia del segnale anodico, corrispondente alla misura della carica rilasciata, utilizzata in sede di analisi per le correzioni del *time walk*.

La presenza delle tre soglie di discriminazione consente alla logica di *trigger* di classificare velocemente l'evento e quindi produrre i segnali necessari all'acquisizione.

Il componente usato per la misura dei tempi e delle cariche (tempo sopra soglia) è un TDC LeCroy MTD133 modificato cambiando le uscite da ECL a CMOS per ridurre il consumo di potenza.

La logica della **SFET** è costruita su due FPGA ACTEL. Una è dedicata al *front end* e l'altra al *read out*. La prima interfaccia i segnali discriminati con la *time expansion* l'integratore ed il TDC, mentre la seconda legge questi ultimi e interfaccia la scheda con la **SDR**.

La risoluzione nominale di questi TDC è 1 *ns*; per ottenere la risoluzione necessaria sul tempo (25 *ns*) è quindi necessario espandere di 40 volte quest'ultimo prima di effettuarne la digitalizzazione. Una volta campionato, il tempo reale e la sua risoluzione saranno quelle misurate dal TDC diviso 40.

Quando viene inviato un *FT* alla **SFET** questa acquisisce l'evento, diventando "sorda" a nuovi eventi: la logica programmata sulla ACTEL blocca i segnali che arriverebbero al TDC tranne la *history*.; L'importanza di questa misura non riguarda infatti il tempo di arrivo, ma soltanto la presenza o meno di un segnale (*history*). Se infatti la misura di tempo su scala espansa non viene influenzata dall'arrivo di un nuovo segnale sopra soglia, non si può dire la stessa cosa per la misura della carica. La misura della carica è data dal tempo durante il quale il segnale anodico rimane al di sopra della soglia di discriminazione *low*. Nel caso di *hits* multipli quindi questo tempo cambia e la misura della carica viene falsata, ma grazie alla *history* è possibile identificare e scartare questi eventi in fase di analisi.

Quando la **SFET** acquisisce un evento rimane "sorda" finché non riceve un segnale di *clear*, che può arrivare dopo la lettura dell'evento da parte della **SDR**, può arrivare dalla logica di *trigger* se l'evento non è stato selezionato per l'acquisizione, o può essere generato dalla **SFET** stessa nel caso in cui non arrivi il *fast trigger* entro 250 *ns* dalla lettura dell'evento (*time out*).

5.2.3 SFEC

La SFEC è dedicata alla lettura della carica dei 24 dinodi relativi ai canali in carico al *crate*. La scheda è praticamente identica alla SFET, se non fosse che i canali dedicati a *history* e *time expansion* sono utilizzati per semplici letture di carica.

In definitiva, quindi, i 24 canali di lettura necessari sono ottenuti con gli stessi componenti presenti sulla SFET.

5.2.4 SDR

La **SDR** (figura 5.4) è dotata di un link AMSwire di interfacciamento verso la **CDDC** dedicata al TOF e 5 *link* punto-a-punto per il collegamento con le schede di *front end* alloggiate nello stesso *crate*. Il processore che gestisce la scheda è il

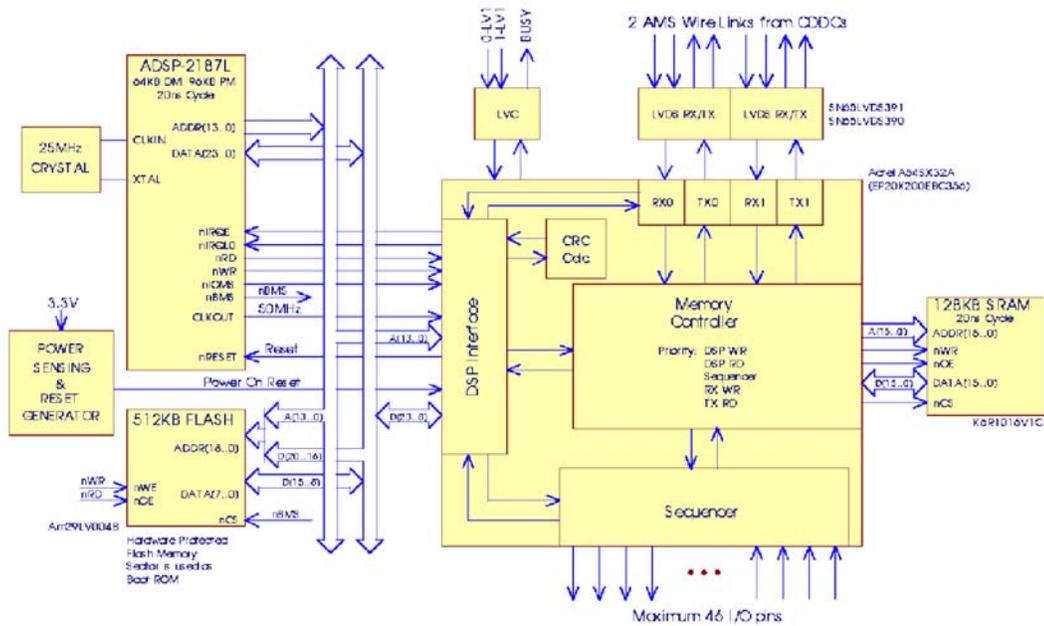


Figura 5.4: Schema a blocchi della SDR.

DSP ADSP-2187L (già visto nella *Guido Board*) interfacciato tramite una FPGA ACTEL ad un banco di memoria RAM (128 KB), una FLASH (512 KB) utilizzata per il software del DSP, ed i vari *link* esterni. Nella FPGA saranno implementate l'interfaccia con il DSP (che è collegato in modo diretto alla FLASH dalla quale riceve il *software* da eseguire), l'interfaccia con la memoria RAM e con i *link* di I/O. Ogni passaggio di informazioni tra due componenti interfacciati tramite l'FPGA viene filtrato da un CRC (*Cyclic Redundancy Check*) per l'identificazione degli errori di trasmissione. Ad ogni blocco dati viene accodata una parola di 16 bits (*Frame Check Sequence*, FCS, calcolato come resto della divisione del blocco dati per una parola fissa) che consente al dispositivo ricevente l'identificazione di eventuali errori di trasmissione.

5.2.5 USCM

L'**USCM** è basata su un microcontrollore DS80C390, ed è dotata delle seguenti porte di I/O:

- 16 linee LVDS di *input/output*;

- 8 linee LVDS di *output*;
- 3 linee LVDS di controllo;
- 32 canali ADC di *input*, intervallo di tensione ammesso $[0.0, 4.096] V_{DC}$, risoluzione di 12 *bits*;
- 16 linee di DAC di *output*, intervallo di tensione prodotto $[0.0, 4.096] V_{DC}$, risoluzione di 12 *bits*;
- 8 porte per catene di sensori di temperatura DALLAS di tipo DS1820;
- 2 linee seriali CANbus con protocollo CAN 2.B;
- 2 linee seriali RS232 con velocità massima di 38400 *bps*;

Le linee CANbus sono utilizzate per il trasferimento delle informazioni lungo la catena di *slow control* dell'apparato. Il CANbus è uno *standard* industriale basato su un pacchetto dati di 16 *bits* oltre a 3 *bits* per la sincronia; su ogni *bus* possono essere connessi più dispositivi, uno solo dei quale ricopre il ruolo di *master*. L'identificazione del dispositivo cercato avviene tramite la diramazione sul *bus* di un pacchetto contenente una parola di identificazione, alla quale risponderà soltanto lo *slave* corrispondente. Dopodiché avviene la trasmissione dei pacchetti contenenti i dati.

Nel caso del TOF l'**USCM** sarà utilizzata per l'impostazione delle soglie, delle maschere di *trigger* e per il controllo della temperatura sui vari componenti dell'apparato; in particolare essa comunicherà con **SDR**, **SFET** e **SFEC** tramite le linee digitali LVDS.

Per il controllo di temperatura sarà possibile utilizzare fino a 256 sensori visto che ogni porta ad esso dedicata è in grado di gestire catene composte da 32 sensori.

5.3 Integrazione SDR-SFE[X]

Per sviluppare l'interfacciamento tra i due moduli principali dell'elettronica del TOF è stata utilizzata l'**S9007**, scrivendo in VHDL il codice relativo all'FPGA.

Il primo passo è stato quello di effettuare il *boot* del DSP, ovvero la fase in cui il software che esso dovrà eseguire viene caricato in memoria ed il processore viene configurato per l'avvio.

Il boot del DSP

Il DSP utilizzato permette due modalità di *boot*:

- IDMA: i *buses* indirizzo e dati interni vengono resi disponibili all'esterno in modo che sia possibile scrivere il codice direttamente nella memoria interna del processore. Quando la locazione 0 viene scritta il DSP inizia l'esecuzione;
- BDMA: la prima operazione effettuata dal DSP è il trasferimento di 32 parole da 24 *bits* da una memoria esterna alla memoria interna. Queste 32 parole devono contenere il *boot loader* che si occuperà di caricare il resto del codice.

I modi di avvio attualmente implementati sono tre, e si differenziano nel dispositivo che fornisce il codice di avvio:

1. *boot da UART*: il codice viene acquisito dall'interfaccia seriale (solitamente collegata al PC) e traferito nella memoria del DSP in modalità IDMA;
2. *boot da FLASH*: il codice si trova nella memoria FLASH e il DSP carica da essa il *boot loader* che si occupa del trasferimento del resto del programma;
3. *generico IDMA*: questa modalità consente l'acquisizione del codice da sorgenti diverse, che potrebbero essere ad esempio AMSwire, la memoria RAM o altro.

Nella configurazione finale della **SDR** il *boot* sarà normalmente fatto dalla FLASH, anche se saranno possibili aggiornamenti del codice via AMSwire. La modalità di *boot* da UART è utilizzata soltanto durante la fase di sviluppo.

La gestione del DSP in questa prima fase viene fatta da un componente chiamato **BOOT_IDMA** che accetta come ingressi due segnali (oltre ai consueti **RESET** e **CLOCK**); il primo, **IDMA_WR_EN**, viene attivato quando una parola (indirizzo o codice) è pronta ad essere scritta nella memoria interna, mentre il secondo, **IDMA_AD_EN**, informa il componente se la parola è un indirizzo o un codice.

Le uscite principali di **BOOT_IDMA** sono i segnali che si interfacciano direttamente al DSP:

- **DSP_IS**: *IDMA Select*, seleziona la porta IDMA del DSP (attivo basso);
- **DSP_IAL**: *IDMA Address Latch*, informa il DSP che l'indirizzo è pronto per essere memorizzato (attivo alto);
- **DSP_IRD**: *IDMA Read enable*, chiede al DSP di eseguire la lettura della parola indirizzata dall'indirizzo precedentemente memorizzato (attivo basso);
- **DSP_IWR**: *IDMA Write enable*, chiede al DSP di eseguire la scrittura della parola nella locazione indirizzata in precedenza (attivo basso).

Le parole di indirizzo e codice (16 *bits*) vengono poste sul *bus* connesso alla porta IDMA dal dispositivo che fornisce il codice *boot*; l'interfacciamento con **BOOT_IDMA** permette la sua sincronizzazione con il DSP.

Per l'implementazione dell'architettura di **BOOT_IDMA** si faccia riferimento all'appendice C; si noti che sono presenti due segnali di I/O aggiuntivi rispetto a quelli appena descritti:

- **DSP_IACK**: *IDMA port ACKnowledge*. Quando è attivo il DSP è occupato e bisogna attendere che si liberi per ottenere l'accesso alla porta IDMA. In realtà per il *boot* questo segnale è ininfluente dato che prima del *boot* il DSP è disattivo;
- **IDMA_DONE**: informa i dispositivi connessi a **BOOT_IDMA** quando la sequenza di scrittura è finita;

Il sequencer per la lettura delle SFE[X]

Attualmente è in fase di sviluppo la parte relativa all'acquisizione dati dalle schede di *front end*. Come si è spiegato ogni **SDR** sarà addetta alla lettura ed al controllo di 4 **SFET** (3 per il TOF e 1 per le anticoincidenze) e 1 **SFEC**; in tutto 5 schede. La lettura avverrà serialmente su 5 linee connesse punto-a-punto. Il protocollo di comunicazione è mantenuto il più semplice possibile: l'**SDR** invia alla scheda di *front end* un carattere che rappresenta un comando (*reset*, *getData*, etc.) a cui essa risponde o confermando o meno il successo dell'operazione, o con l'invio del pacchetto dei dati.

I dati acquisiti, impacchettati in parole da 16 *bits* tramite uno *shift register*, vengono posti temporaneamente in memoria RAM; sarà poi compito del DSP la loro compressione/formattazione per l'invio al punto di raccolta (**CDDC**).

5.3.1 Prospettive future

L'elettronica di AMS-02 sta entrando ora nella fase centrale dello sviluppo; dovrebbero essere ultimate nel giro di poche settimane una scheda di interfacciamento PCI-AMSwire e un *plug-in* per la **S9007** che comprende un'interfaccia VME e un connettore PMC per l'eventuale utilizzo delle schede *PowerPC* sviluppate per i livelli superiori del DAQ di AMS. Nella stessa scheda ci saranno 4 *link* AMSwire.

La prima renderà molto più rapido il *debugging* dei vari moduli della parte dedicata all'acquisizione dati, consentendo in particolare di iniziare la fase di integrazione fra i moduli diversi.

La seconda renderà la scheda **S9007** facilmente integrabile con i sistemi di acquisizione standard.

Per quanto riguarda il TOF, il *software* descritto nel capitolo 4 sarà integrato con il codice necessario sia all'utilizzo dell'interfaccia PCI-AMSwire per lo sviluppo dell'elettronica di acquisizione, che alla lettura tramite **S9007** (su AMSwire o VME) delle schede di *front end* per la loro caratterizzazione finale.

Conclusioni

Il lavoro sperimentale descritto in questa tesi ha chiarito diversi aspetti relativi allo sviluppo del sistema per il tempo di volo di AMS-02. In questo paragrafo conclusivo verranno analizzati uno per volta.

Il prototipo dei contatori

Sono state effettuate le prime analisi sul prototipo dei contatori che formeranno il TOF. I risultati hanno messo in luce che

- per rendere più affidabile il telescopio per raggi cosmici è consigliabile sostituire il contatore usato come riferimento con uno dalle caratteristiche migliori; in quello attuale l'invecchiamento dello scintillatore plastico ha causato un certo degrado delle prestazioni.
- il contatore prototipo presenta un'andamento per la risoluzione intrinseca sostanzialmente corretto ($\sigma = 152$ ps al centro) ma leggermente superiore a quanto atteso. Probabilmente l'utilizzo di fototubi *fine mesh* gioca un ruolo fondamentale in questo effetto.

Il sistema di acquisizione dati del telescopio

Attualmente il laboratorio di Bologna è dotato di un sistema funzionante con il quale acquisire velocemente i dati necessari alla caratterizzazione dei contatori. Una volta identificato un contatore adatto verrà sostituito all'attuale contatore di riferimento portando in condizioni ottimali il sistema di acquisizione.

Il confronto tra i dati forniti dall'*hardware* basato su *bus* CAMAC e quello basato su *bus* VME mette in evidenza due fattori fondamentali:

- l'elettronica di nuovo tipo è sostanzialmente più versatile, controllabile e dall'utilizzo più immediato rispetto a quella vecchia. Il lato negativo è che necessita

ancora di una fase di *debug* volta a migliorarne le prestazioni: la risoluzione intrinseca ottenuta sul contatore prototipo risulta pari a 210 *ps* al centro, peggiore di circa 60 *ps* rispetto al valore ottenuto con il sistema CAMAC;

- i moduli CAMAC forniscono prestazioni mediamente superiori, anche se l'invecchiamento ne riduce notevolmente l'affidabilità e, soprattutto, la disponibilità.

In particolare il *Constant Fraction Discriminator* utilizzato nella versione VME del *setup* non risponde ai requisiti necessari ad ottenere una buona risoluzione temporale; la soluzione con discriminatori a soglia e correzione per il *time walk* in fase di analisi rimane quella più affidabile.

Il TDC utilizzato necessita di una importante modifica: l'inserimento di un segnale di *clear* che permetta la pulizia della *fifo* dati senza la reimpostazione del modulo ai parametri iniziali.

Sarebbe inoltre utile la presenza di un modulo I/O *register* da utilizzarsi come il classico *Status A* CAMAC per semplificare il sistema di *trigger* attualmente usato. A mio parere il confronto tra i due sistemi andrebbe approfondito, in particolare ora che sono disponibili un *software* che consente l'integrazione immediata tra di essi, e una certa quantità di moduli di entrambi i tipi.

Una soluzione che mi appare sensata è la realizzazione di un sistema che integri entrambi i tipi di elettronica, in modo da poter sfruttare gli aspetti positivi di ognuno di essi: la disponibilità dei moduli CAMAC e la profonda conoscenza che si ha di essi, e la versatilità dei moduli VME che permette di semplificare notevolmente la gestione del sistema di acquisizione.

Il software per l'acquisizione dati

La scrittura del DAS si è rivelata l'aspetto più impegnativo di quest'anno di lavoro; il risultato ottenuto risponde però pienamente agli obiettivi posti inizialmente.

In particolare risulta ottimale la scelta fatta sulla struttura gerarchica delle classi (figura 4.1), che ha reso quasi banale il processo di integrazione di nuovi moduli ed interfacce nel DAS. Basti pensare che, mentre la scrittura dello scheletro del *software* ha richiesto alcuni mesi, l'integrazione al suo interno della parte di gestione del *bus* CAMAC e di alcuni moduli (TDC, ADC, SCALER, LATCH, STATUS A) si è esaurita in una settimana.

Un'altro aspetto che ritengo di fondamentale importanza è la completa separazione tra i processi di acquisizione e *monitoring* e quello che gestisce la realizzazione del *setup* (GUI); questa caratteristica, oltre che mantenere la velocità dell'acquisizione ai livelli richiesti (più del 95% di efficienza a 500 *Hz*) rende semplice l'ipotesica

migrazione della parte relativa all'acquisizione in sistemi *embedded*, con conseguente aumento di prestazioni.

Proprio al fine di un incremento della velocità di acquisizione penso sia utile il passaggio a sistemi operativi *real time* parallelamente all'introduzione della gestione degli *interrupt* nel DAS (che attualmente lavora in *polling*).

Ci sono inoltre notevoli prospettive di utilizzo future, le più immediate delle quali sono:

- aggiornamento del *setup* per lo studio dei fototubi che attualmente presenta alcuni problemi di affidabilità;
- integrazione di AMSwire tramite l'interfaccia PCI-AMSwire in fase di realizzo che consentirà di facilitare lo sviluppo dell'elettronica di acquisizione del TOF (soprattutto nella fase di *debug*);
- integrazione della versione VME della **S9007**.

In particolare si prevede l'utilizzo del DAS per la lettura delle **SFE[X]** nella fase di acquisizione dei dati necessari alla loro caratterizzazione. Per fare questo le **SFE[X]** saranno lette tramite la **S9007**, e lo sviluppo del sistema di lettura sarà il primo passo per l'integrazione tra le **SDR** e le schede di *front end*.

Integrazione SDR-SFE[X]

L'integrazione tra i due moduli principali del sistema di acquisizione del TOF è ancora al livello iniziale, ma i risultati parziali ottenuti permettono di prevedere uno sviluppo veloce.

La fase di *boot* del DSP è attualmente implementata; una volta ultimato il *sequencer* per la lettura delle **SFE[X]** esso sarà programmato per la riduzione e formattazione dei dati, e per il loro invio tramite AMSwire alla **CDDC** assegnata al TOF.

Appendice A

Alcuni estratti del codice del DAS

Buffer

Quello che segue è il codice relativo alla dichiarazione della classe `Buffer` e della struttura `runInfo_t` usata per organizzare lo scambio di informazioni tra i processi di acquisizione e *monitoring* tramite la *shared memory*, nonché l'implementazione dei due costruttori e del distruttore di cui la classe è dotata.

```
#define BASE_DATA_POINTER ( sizeof( runInfo_t ) + 10 )
#define STRNAME_SIZE 200
#define ONE_KBYTE 1024

struct runInfo_t {
    unsigned long eventNum;
    unsigned long eventsToAcquire;
    unsigned long channelsNum;
    unsigned long eventSize;
    unsigned long bufferSize;
    unsigned long totalDataSpace;
    unsigned long currentSpace;
    unsigned long currentAcqEvt;
    unsigned long currentMonEvt;
    unsigned long currentAcqData;
    unsigned long currentMonData;
    unsigned long busErrorNumber;
    int errorTyp;
};
```

```
int lastBusErrTyp;
char infoFilename [STRNAME_SIZE];
bool canDoAcq;
bool stopAcq;
bool acqInitDone;
bool acqDone;
bool bufferIsFull;
bool evtAcquired;
bool monitorInitDone;
};

class Buffer
{
private:
    char * smPointer;
    ofstream * storage;
    runInfo_t * runInfo;
    int *data;
    int *scale;
    int currentScale;
    int segment_id;
    int baseDataPointer;
    bool iAmMonitor;
    bool eventIsGood;
public:
    Buffer( const char * file, unsigned long bfSize );
    Buffer( int segment_id );
    ~Buffer();
    int getShmID();
    int errorType();
    int lastBusErrorType();
    void setChannelsNumber( unsigned long num );
    void setEventsToAcquire( unsigned long num );
    void setInfoFilename( char * name );
    void setEventIsAquired( bool v );
    void setErrorTyp( int typ );
    void startAcquisition();
    void pauseAcquisition();
    void stopAcquisition();
```

```

void setAcqInitDone();
void setMonitorInitDone();
void done();
unsigned long eventsToAcquire();
unsigned long eventNumber();
unsigned long bufferSize();
unsigned long dataBufferSize();
unsigned long eventSize();
unsigned long eventChannelsNumber();
unsigned long busErrorsNumber();
char * infoFilename();
int * getMonEvent();
int * scales();
bool acqInitDone();
bool monitorInitDone();
bool iCanAcquire();
bool isStopped();
bool isFull();
bool isDone();
bool eventIsAcquired();
bool iCanUpdateHisto();
void getScales( int * newScales );
void getEvent( int * newData );
void endMonEvent();
void delEvent( int errTyp );
void endEvent();
bool save();
};

Buffer::Buffer( const char * file, unsigned long bfSize )
{
    storage = new ofstream( file );

    segment_id = shmget( IPC_PRIVATE, bfSize * ONE_KBYTE,
                       IPC_CREAT | IPC_EXCL | S_IRUSR | S_IWUSR );

    smPointer = (char*)shmat( segment_id, 0, 0 );
    runInfo = (runInfo_t*)smPointer;
    data = (int*)&smPointer[ BASE_DATA_POINTER ];
}

```

```

struct shmid_ds shmbuffer;
shmctl( segment_id, IPC_STAT, &shmbuffer );

currentScale = 0;
runInfo->eventNum = 0;
runInfo->eventsToAcquire = 0;
runInfo->channelsNum = 0;
runInfo->eventSize = 0;
runInfo->bufferSize = shmbuffer.shm_segsz;
runInfo->currentAcqEvt = 0;
runInfo->currentMonEvt = 0;
runInfo->currentAcqData = 0;
runInfo->currentMonData = 0;
runInfo->canDoAcq = false;
runInfo->stopAcq = false;
runInfo->bufferIsFull = false;
runInfo->monitorInitDone = false;
iAmMonitor = true;
}

Buffer::Buffer( int s_id )
{
    segment_id = s_id;
    /* Shared Memory Allocation */
    char * smPointer = (char*)shmat( s_id, 0, 0 );
    runInfo = (runInfo_t*)smPointer;
    data = (int*)&smPointer[ BASE_DATA_POINTER ];
    currentScale = 0;
    runInfo->busErrorNumber = 0;
    runInfo->lastBusErrTyp = 0;
    runInfo->acqInitDone = false;
    runInfo->acqDone = false;
    runInfo->evtAcquired = false;
    runInfo->totalDataSpace = 0;
    eventIsGood = true;
    iAmMonitor = false;
}

```

```
Buffer::~Buffer()
{
    shmdt( (char*)runInfo );
    if ( iAmMonitor ) shmctl( segment_id, IPC_RMID, 0 );
}
```

Acquisition Process

Quello che segue è il codice del processo che si occupa della presa dati.

```
segmentID = atoi( argv[1] );
Buffer buffer( segmentID );
parentPid = getppid();

while( !buffer.monitorInitDone() );

/*
 * In questo punto si trova in realta' il codice relativo
 * all'interpretazione del file di configurazione, parte
 * che e' stata omessa per chiarezza.
 */

if( status != READY ||
    triggerMod.size() == 0 ||
    pollingMod.size() == 0 ||
    acquireMod.size() == 0
    )
{
    buffer.setErrorType( BAD_INIT );
    buffer.done();
    kill( parentPid, SIGUSR2 );
    return EXIT_FAILURE;
}

if ( buffer.eventSize() > buffer.dataBufferSize() )
{
    buffer.setErrorType( BUFFER_TOO_SMALL );
    buffer.done();
    kill( parentPid, SIGUSR2 );
    return EXIT_FAILURE;
}
```

```

    }

for (unsigned int i = 0; i < triggerMod.size(); i++ )
    {
        triggerMod[ i ]->clear();
        triggerMod[ i ]->set_veto();
    }
vetoIsSetted = true;
for (unsigned int i = 0; i < pollingMod.size(); i++ )
    pollingMod[ i ]->clear();
for (unsigned int i = 0; i < acquireMod.size(); i++ )
    acquireMod[ i ]->clear();

buffer.setAcqInitDone();

while( !buffer.isStopped() )
    {
        if( monitorTrigger != NULL &&
            monitorTrigger->thereisdata() )
            {
                kill( parentPid, SIGALRM );
                monitorTrigger->reset_veto();
            }

        while( !buffer.iCanAcquire() || buffer.isFull() );

        buffer.setEventIsAquired( false );

        if ( vetoIsSetted )
            {
                for (unsigned int i = 0; i < triggerMod.size(); i++ )
                    triggerMod[ i ]->reset_veto();
                vetoIsSetted = false;
            }

        dataReady = true;
        for (unsigned int i = 0; i < pollingMod.size(); i++ )
            if ( !pollingMod[ i ]->thereisdata() ) dataReady = false;
    }

```

```
if( dataReady )
{
    for (unsigned int i = 0; i < triggerMod.size(); i++ )
        triggerMod[ i ]->set_veto();
    vetoIsSetted = true;

    for (unsigned int i = 0; i < acquireMod.size(); i++ )
    {
        if ( ( busError = acquireMod[ i ]->event( &bfp ) ) )
        {
            buffer.delEvent( busError );
            buffer.setEventIsAquired( true );
            break;
        }
        else
            buffer.getEvent( bfp );
    }
    buffer.endEvent();
    if ( buffer.isFull() )
        kill( parentPid, SIGALRM );
}

    buffer.setEventIsAquired( true );
}
buffer.done();
kill( parentPid, SIGALRM );
return EXIT_SUCCESS;
```


Appendice B

Studio dell'efficienza del DAS

Si riportano i dati relativi allo studio dell'efficienza del DAS effettuato sul *setup* utilizzato per il *test beam* che si svolgerà tra il 14 ed il 19 ottobre 2002 sul fascio SPS del CERN. Le quantità riportate sono

- R : rate (Hz);
- $N_{trigger}$: numero di impulsi prodotti dalla *dual timer*;
- N_{acq} : numero di eventi acquisiti dal DAS;
- N_{lost} : $N_{trigger} - N_{acq}$;
- N_{delay} : eventi persi a causa del ritardo tra l'azzeramento dello *scaler* e l'inizio dell'acquisizione;
- $\epsilon = 1 - \frac{N_{tl} - N_{delay}}{N_{trigger}}$

Ogni acquisizione dura 60 s.

R (Hz)	$N_{trigger}$	N_{acq}	N_{lost}	N_{delay}	ϵ (%)
10	313	312	1	1	100
20	621	619	2	2	100
50	1548	1535	13	8	99.7
100	3090	3066	24	19	99.8
200	6141	6093	48	23	99.6
300	9370	9059	311	45	97.2
400	12626	12411	215	76	98.9
500	15667	14986	681	99	96.3
700	22393	16737	5656	123	75.3
1000	31464	14405	17062	173	46.3

Appendice C

Il codice della SDR

Si riporta il codice relativo all'architettura del componente `BOOT_IDMA`, che gestisce l'interfacciamento tra il dispositivo che fornisce il codice di *boot* (FLASH, UART, AMSwire, etc.) e la porta IDMA del DSP.

Architecture `ARCH_IDMA` of `BOOT_IDMA` is

```
--
--
-- IS IAL IWR IDMA_DONE
constant IDLE      : std_logic_vector(3 downto 0) := "1010";
constant AD_LATCH_1 : std_logic_vector(3 downto 0) := "0110";
constant AD_LATCH_2 : std_logic_vector(3 downto 0) := "0111";
constant DT_WRITE_1 : std_logic_vector(3 downto 0) := "0000";
constant DT_WRITE_2 : std_logic_vector(3 downto 0) := "0001";

signal state : std_logic_vector(3 downto 0);

begin

    DSP_IS  <= state(3);
    DSP_IAL <= state(2);
    DSP_IRD <= '1';
    DSP_IWR <= state(1);
    IDMA_DONE <= state(0);

    process (CLOCK, RESET)
    begin
        if RESET = '0' then
            state <= IDLE;
        else
```

```

if rising_edge(CLOCK) then
  case state is

    when IDLE =>
      if (IDMA_WR_EN = '1') then
        if (IDMA_AD_EN = '1')
          then
            state <= AD_LATCH_1;
          else
            state <= DT_WRITE_1;
          end if;
        else
          state <= IDLE;
        end if;

      when AD_LATCH_1 =>
        state <= AD_LATCH_2;

      when DT_WRITE_1 =>
        state <= DT_WRITE_2;

      when AD_LATCH_2 =>
        state <= IDLE;

      when DT_WRITE_2 =>
        state <= IDLE;

      when others =>
        state <= IDLE;

    end case;
  end if;
end if;
end process;

end ARCH_IDMA;

```

Bibliografia

- [1] N. Cabibbo. Unitary symmetry and leptonic decays. *Phys. Rev. Lett.*, 10:531, 1963.
- [2] M. Kobayashi and T. Maskawa. Cp violation in the renormalizable theory of weak interaction. *Prog. Theor. Phys.*, 49:652, 1973.
- [3] J. H. Christenson et. al. Evidence for the 2π decay of K_2^0 meson. *Phys. Rev. Lett.*, 13:138, 1964.
- [4] F. Abe et. al. Measurement of the CP-violation parameter $\sin(2\beta)$ in $B_d^0/\bar{B}_d^0 \rightarrow J/\Psi K_S^0$ decays. *Phys. Rev. Lett.*, 81:5513, 1998.
- [5] A. D. Sakharov. Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe. *JETP Lett.*, 5:24, 1967.
- [6] G. t'Hooft. Symmetry breaking through Bell-Jackiw anomalies. *Phys. Rev. Lett.*, 37:8, 1976.
- [7] A. Riotto and M. Trodden. Recent progress in baryogenesis. *Ann. Rev. Nucl. and Part. Sci.*, 49:35, 1999.
- [8] I.I. Bigi and A.I. Sanda. *CP Violation*. Cambridge Univ. Press, 2000.
- [9] Lewis H. Ryder. *Quantum Field Theory*. Cambridge University Press, 1989.
- [10] Edward E. Kolb and Michael S. Turner. *The Early Universe*. Addison-Wesley, 1990.
- [11] C.B. Netterfield *et al.* A measurement by BOOMERANG of multiple peaks in the angular power spectrum of the cosmic microwave background. *The Astrophysical Journal*, 571:604, 2002.
- [12] J.C. Mather *et al.* Calibration design for the COBE far infrared absolute spectrophotometer (FIRAS). *The Astrophysical Journal*, 512:511, 1999.

-
- [13] C. Pryke *et al.* Cosmological parameter extraction from the first season of observations with DASI. *The Astrophysical Journal*, 568:46, 2000.
- [14] A.H. Broeils K.G. Begeman and R.H. Sanders. Extended rotation curve of spiral galaxies: dark haloes and modified dynamics. *Mon. Not. R. Astr. Soc.*, 249:523, 1991.
- [15] S.M. Faber. Masses and mass-to-light ratios of galaxies. *Ann. Rev. A&A*, 17:135, 1979.
- [16] V. Trimble. Existence and nature of dark matter in the universe. *Ann. Rev. A&A*, 25:423, 1987.
- [17] S. Perlmutter *et al.* Measurements of Ω and Λ from 42 high-redshift supernovae. *The Astrophysical Journal*, 517:565, 1999.
- [18] P. Bhattacharjee and G. Sigl. Origin and propagation of extremely high-energy cosmic rays. *Phys. Reports*, 327:109, 2000.
- [19] Particle Data Group. Cosmic rays. <http://pdg.lbl.gov/>, 2002.
- [20] Malcolm S. Longair. *High Energy Astrophysics*, volume 1, 2. Cambridge University Press, 2 edition, 1992, 1994.
- [21] A.R. Bell. The acceleration of cosmic rays in shock fronts - I. *Mon. Not. R. Astr. Soc.*, 182:147, 1978.
- [22] A.R. Bell. The acceleration of cosmic rays in shock fronts - II. *Mon. Not. R. Astr. Soc.*, 182:443, 1978.
- [23] P.O. Lagage and C.J. Cesarsky. The maximum energy of cosmic rays accelerated by supernovae shocks. *Astronomy and Astrophysics*, 125:249, 1983.
- [24] J. Alcaraz *et al.* Search for antihelium in cosmic rays. *Phys. Lett.*, B461:387–396, 2000.
- [25] J. Alcaraz *et al.* Helium in near earth orbit. *Phys. Lett.*, B494:193–202, 2000.
- [26] J. Alcaraz *et al.* Cosmic protons. *Phys. Lett.*, B490:27–35, 2000.
- [27] William R. Leo. *Techniques for Nuclear and Particle Physics Experiments. A How-to Approach*. Springer-Verlag, 1987.
- [28] R.L. Gluckstern. Uncertainties in track momentum and direction due to multiple scattering. *Nucl. Instrum. Methods Phys. Res.*, 24:381, 1963.

- [29] V. Plyaskin E. Choumilov, Yu. Galaktionov. A sampling electromagnetic calorimeter for alpha magnetic spectrometer. *Nucl. Instrum. Methods Phys. Res.*, A426:625–632, 1999.
- [30] L. Baldini. I contatori e l'elettronica per la misura del tempo di volo dei raggi cosmici dell'esperimento spaziale ams. Tesi di laurea, Gruppo AMS, Corso di Laurea in Fisica dell'Università di Bologna, Anno Accademico 1997-1998.

Ringraziamenti

Vorrei ringraziare tutte le persone che mi hanno aiutato ad arrivare fino in fondo, in particolare:

- | | |
|-----------|---|
| Palmonari | Per la disponibilità, la professionalità, l'esperienza che mi ha messo a disposizione e per la simpatia che lo distingue. |
| Beppe | Perchè sa sempre dove mettere le mani quando c'è qualche problema in laboratorio! |
| Federico | Perchè è l'unico che abbia mai conosciuto in grado di fare così tante cose contemporaneamente, soprattutto con tre ragazzini sempre intorno... |
| AMSBO | Un insieme di persone fantastiche con cui ho condiviso questo anno di tesi: Cristina, la PR del gruppo, Diego, il tuttologo, Veronica (anzi Veronihha), la produttrice di faccine <;) più fantasiosa che conosca, Roberto-SuperKumac-Hamamatsu, Lucio, causa principale dei chili di troppo che mi si sono attaccati addosso quest'anno, Nada, che spesso ha fatto da segretaria rispondendo sempre gentile alle mie telefonate da Ginevra, e Francesca, che tra un po' parte per Madrid e ci tradisce per il RICH!
Ringrazio anche Pilastrini, per i preziosi consigli sulla Svizzera, Lolli, per la ricetta della pasta ai gamberi, super-Massera, che con la sua super-simpatia rende il piano -1 il più divertente, e i colleghi di ALICE nostri vicini. |
| UNIBO | Tutti gli amici trovati in questi cinque anni: Totta, Silvia, Giulio, Paola, ... siete troppi! scusate ma non posso scrivervi tutti! però non posso non scrivere... |

- Fillow Che dire di Fillo? innanzi tutto è il custode ufficiale di tutte le mie cose: quando non trovo un *mio* libro, dei *miei* appunti o altro basta che vada da lui... e per ricambiare mi ha offerto da bere diverse volte (anche se il giorno dopo non se lo ricorda mai!). Comunque in questi ultimi anni ho passato più tempo con lui andando avanti e indietro dall'università che con i miei genitori, e devo ammettere che se sono stati anni favolosi è stato anche grazie a lui!
- il Super Bar Bocciù, che quando si tratta di snowboard o mountain bike è sempre pronto, Joe, il più matto che conosco, Maxx, che è matto come Joe, e tutti gli altri: ... anche voi siete troppi!
- Southern & Andrea, la Dani, Giulio e Scaglia, veramente insuperabili! grazie soprattutto per aver sopportato la mia "trasfer-ta estiva" a Ginevra! Giulio e Scaglia poi sono miei colleghi musicali (ma soprattutto amici) da quando ero uno "sbarbino", e con loro condivido alcuni dei ricordi più belli. Grazie!
- Catia La mia sorellina, che mi sopporta da circa 21 anni e continua a volermi bene...
- Daniela Che mi sopporta da meno tempo di mia sorella ma lo fa comunque benissimo, soprattutto viste le mie fughe domenicali con lo snowboard, la mountain bike, la palla da calcio, ...
- Mamma e Papà Infine le due persone a cui devo di più. Sono loro che mi hanno permesso di arrivare a questo risultato, oltre che pagandomi gli studi, dandomi sempre buoni consigli e appoggiando le mie scelte. Grazie davvero di tutto!

Meritate tutti molto di più delle poche righe che vi ho dedicato, e spero di avere il modo di rimediare in futuro!