

UNIVERSITA' DEGLI STUDI DI CATANIA
FACOLTA' DI INGEGNERIA

Corso di laurea di I livello in Ingegneria Informatica

TESI DI LAUREA

**Sviluppo di un sistema di posizionamento micrometrico
per applicazioni con fasci di particelle**

Sergio Simone Sciré Scappuzzo
Matricola 616/000759

Relatore:
Prof. Ing. Michele Malgeri

Correlatori:
Dott. Paolo Finocchiaro
Dott. Luigi Cosentino

Anno Accademico 2006 – 2007

Ringraziamenti

Scrivo con grande emozione queste righe, per ringraziare tutte le persone che hanno reso possibile la riuscita di un così lungo e faticoso lavoro.

Prof. Michele Malgeri: grazie per la cortesia e la disponibilità che non mi ha mai negato durante il lungo percorso di tirocinio e scrittura della tesi. Mi ritengo fortunato ad aver avuto una guida così cordiale, sempre pronta a dare incoraggiamenti e giudizi lusinghieri sul lavoro svolto.

Dott. Paolo Finocchiaro: non mi stupirò mai abbastanza della passione che spendi sul tuo lavoro e di come riesci ad affascinare chi, come me, si avvicina per caso agli studi applicati della fisica. Ciò che più ammiro in te è la naturalezza con la quale riesci a tirar fuori dai tuoi collaboratori ingegno e capacità inaspettate, oltre naturalmente ad offrire un rapporto di amicizia piacevolissimo. Grazie per aver creduto in me dal primo momento e avermi dato questa opportunità.

Dott. Luigi Cosentino: devo dire che la maggior parte dei problemi che mi si sono presentati li ho risolti grazie ai suggerimenti che venivano fuori del tuo senso critico, non hai mai risparmiato un consiglio o una parola d'incoraggiamento. Grazie per la tua cura meticolosa di questa tesi, ricorderò sempre con simpatia le tue note curiose tra i fiumi di parole da me scritti.

Dott. Alfio Pappalardo: anche se ti sei occupato di lavori differenti dal mio ti ho sempre considerato un compagno di avventure, ti ringrazio di cuore per i consigli, l'aiuto che mi hai sempre offerto e la tua simpatia.

Ing. Pedro Vynck della Vrije Universiteit Brussels: i tuoi lavori per me sono stati fonte inesauribile di informazioni e base di sostegno solida per il lavoro affidatomi. Grazie inoltre per l'accoglienza che mi hai offerto presso il Dipartimento di Fisica Applicata e Fotonica (IR-TONA) durante il mio soggiorno di studi a Bruxelles.

Dottori ricercatori dei LNS: *Francesca Amorini, Agatino Musumarra, Pierpaolo Figuera, Francesca Rizzo, Giuseppe Cardella, Angelo Campisi*. Grazie per la vostra disponibilità nel chiarire con naturalezza, e allo stesso tempo grande cura, gli aspetti tecnici legati ai vostri lavori.

Grazie anche a *Salvatore Pulvirenti* per la disponibilità offerta nel trovare i bugs nascosti nel programma sviluppato.

Biagio Teseo e Luciano Allegra per il supporto alle realizzazioni grafiche.

Grazie di cuore ai miei fratelli, *Sebastiano* e *Linda*, perché non avete mai fatto mancare il vostro conforto e supporto, spesso anche economico, nei momenti di difficoltà. Lo stesso vale per *Carlotta*, con te ho avuto modo inoltre di lavorare insieme in occasione del tirocinio, grazie per avermi sopportato. Grazie anche a *Giovanni*, per l'affetto che mi hai dimostrato.

Un ringraziamento particolare va alla mia ragazza, *Rosanna Barrera*. Penso che gran parte di questo traguardo sia frutto dello sconvolgimento che hai dato alla mia vita. Sei stata sempre presente nei momenti di gioia e soprattutto mi hai confortato quando ho ottenuto dei fallimenti. Non mi stancherò mai di esserti grato per le emozioni, il sostegno e il rispetto che mi regali ogni giorno. Ti amo da impazzire.

Grazie inoltre a tutti i componenti della tua famiglia, per avermi accolto e voluto bene dal primo giorno.

Infine *Pippo* e *Ninetta*, Papà e Mamma. Con me siete stati cocciuti dal primo momento, ammiro la tenacia con cui avete creduto in me ogni giorno, non mi avete mai abbandonato nei momenti di sconforto e avete scommesso su di me. Il raggiungimento di questa meta è stato possibile grazie a voi.

Sergio

*Dedicata a tutti coloro
che hanno sempre creduto in me*

Indice dei contenuti

0	Introduzione	4
1	Strumenti	6
1.1	Modalità di funzionamento degli attuatori di precisione	6
1.1.1	Motorini DC	6
1.1.2	Motorini Stepper (passo passo)	7
1.1.3	Motorini piezoelettrici “inchworm”	9
1.2	Strumenti a disposizione	10
1.2.1	Attuatori Lineari Newport Serie 850G	10
1.2.2	Piastre di movimentazione	11
1.2.3	Controllore ESP 6000 DCIB	13
1.3	Controllo Retroazionato	15
2	Rivelatori di particelle cariche	21
2.1	I prototipi progettati da INFN-LNS e prodotti da ST-Microelectronics	22
2.2	Presentazione delle caratteristiche dei prototipi sottoposti a misurazione	25
3	Software	28
3.1	Scrivere un programma in LabVIEW	28
3.1.1	Tipi di dati	30
3.1.2	Arrays	30
3.1.3	Clusters	31
3.1.4	Strutture Dati in LabVIEW	32
3.1.5	Tools, Controls e Functions Palettes	35
3.2	VI's Newport per la gestione dei parametri	36
3.3	Interfaccia Utente	37

3.3.1	Strutture dati implementate.....	41
3.3.2	Arrays di bit	43
3.3.3	Arrays numerici	44
3.3.4	File di inizializzazione	45
3.4	Descrizione del programma	46
0.1.1	Pagina 0.....	46
3.4.1	Pagina 1.....	48
3.4.2	Pagina 2.....	48
3.4.3	Pagina 3.....	50
3.5	Perché un grafico?	50
3.6	Coordinate e Spostamenti Relativi	53
3.7	Arrays statici e dinamici per la visualizzazione dei puntatori	53
3.8	I pulsanti di Motor Array	54
3.9	I tre cursori: Axis, Beam e Target.....	54
3.10	Schema del rivelatore da inserire sotto il grafico	56
3.11	Zoom.....	58
3.12	Grafico del collimatore	61
4	Allineamento	62
4.1	Errori, Risoluzione del Sistema, Misura di tali entità.....	62
4.2	Gradi di libertà del sistema	66
4.3	Allineamento del sistema di movimentazione al fascio.....	67
4.3.1	Simulazione con collimatore spesso e fascio laser	68
4.3.2	Simulazione con pattern cartaceo e visualizzazione su monitor.....	70
4.3.3	Collimazione del fascio in tre fasi: 2mm, 1mm, 50 μ m	71
4.4	Montaggio del sistema finale, soluzione adottata.....	72
4.4.1	Allineamento ottico del sistema.....	72
4.4.2	Allineamento del prototipo	74
4.4.3	Creazione del file di inizializzazione.....	76
4.4.4	Diagnostica fascio.....	77
4.5	Rototraslazione del sistema di riferimento cartesiano	78
4.5.1	Richiami generali.....	78
4.5.2	Funzioni di allineamento adottate durante il turno di misura	80
4.5.3	Lecture dei conteggi per allineare il primo rivelatore	81

4.5.4	Allineamento strumentale del secondo rivelatore.....	83
5	Misure effettuate	85
5.1	Tabella posizioni.....	85
5.2	Elettronica.....	88
5.3	Altro turno di misura.....	91
6	Perfezionamenti	93
6.1	Controllo allineamento e proposta correzioni automatiche.....	93
6.1.1	Allineamento di un piano per 3 punti.....	93
6.1.2	Raster.....	98
6.2	Soluzione Real-Time.....	99
7	Appendice	102
7.1	INFN-LNS: I Laboratori Nazionali del Sud.....	102
7.2	L'acceleratore di particelle "Tandem" dei LNS.....	103
8	Riferimenti bibliografici	106

0 Introduzione

La complessità crescente della ricerca scientifica ha indotto i ricercatori a progettare sistemi di rivelazione sempre più raffinati, capaci di identificare e caratterizzare in modo completo i prodotti delle reazioni nucleari.

Metodi di identificazione comunemente usati si basano sull'utilizzo della tecnica ΔE - E , che fa uso di due rivelatori posti in successione lungo la traiettoria delle particelle (in una configurazione chiamata telescopio).

La ricerca odierna ha permesso di sfruttare le proprietà di questo tipo di rivelatori di particelle e progettare particolari package modulari, con particolare geometria dell'area attiva, che permettono il loro utilizzo come rivelatori di particelle sensibili alla posizione spaziale seguendo un sistema di riferimento cartesiano X-Y.

L'obiettivo di questo lavoro di tesi - maturato dal candidato durante il periodo di tirocinio didattico (Maggio - Dicembre 2006) svolto presso i Laboratori Nazionali del Sud, Istituto Nazionale di Fisica Nucleare - consiste nella progettazione di un software per posizionare con precisione micrometrica tali rivelatori sotto un fascio di particelle accelerate, in modo da poter studiare i segnali in uscita dai dispositivi, per caratterizzarne risoluzione spaziale e rendimento.

L'esigenza di un sistema altamente accurato deriva dalle esigue dimensioni delle zone attive del rivelatore: ogni singola strip del rivelatore ha infatti dimensioni inferiori al millimetro e il corretto posizionamento del fascio in un dato punto prestabilito al suo interno ricopre quindi un'esigenza di rilevante importanza.

Per venire incontro a tali esigenze di precisione si è dovuto operare con un sistema assemblato ad hoc per la misura da svolgere, utilizzando strumenti che permettessero di

avere una accuratezza di uguale ordine di grandezza agli spostamenti micrometrici comandati dall'operatore.

A tale scopo sono stati utilizzati un set di piastre a scorrimento traslatorio, utilizzate abitualmente per applicazioni di tipo ottico, movimentate da attuatori lineari DC ad alta precisione e ripetibilità negli spostamenti. Gli attuatori, gestiti per mezzo di un controllore, sono stati poi direttamente comandati da un Personal Computer su cui venivano impostate traiettorie e percorsi di esplorazione del fascio incidente sulla superficie del rivelatore.

1 Strumenti

In questo capitolo viene presentata la strumentazione Hardware resa disponibile per la misura: viene mostrata una panoramica sul modello di funzionamento degli attuatori lineari oggi maggiormente utilizzati per applicazioni di precisione (*DC*, *Stepper*, *Piezoelettrici*), viene inoltre illustrato il funzionamento del controllore *ESP6000* utilizzato per la gestione dei movimenti comandati agli attuatori, soffermandosi sugli aspetti di programmazione di basso livello nonché sugli aspetti di controlli automatici necessari per avere un controllo robusto delle traiettorie imposte.

1.1 Modalità di funzionamento degli attuatori di precisione

1.1.1 Motorini DC

Un *motorino DC* (in corrente continua) ha una parte che ruota attorno ad un asse detta *rotore* o armatura (in grigio con gli avvolgimenti colorati nella Figura 1-1) e una parte che genera un campo magnetico fisso (nell'esempio i due magneti colorati) detta *statore*. Un interruttore rotante detto commutatore o collettore a spazzole (nella figura l'anello color rame, fissato all'albero rotante del motore, con i due contatti striscianti + e - collegati alla parte ferma) inverte due volte per ogni giro il verso della corrente elettrica che scorre nei due avvolgimenti, generando un campo magnetico che comporta la produzione di forze di attrazione e repulsione con i magneti permanenti fissi (indicati con N ed S nella figura).

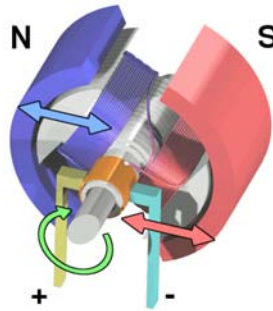


Figura 1-1: Un tipico motore DC.

Quando la corrente scorre negli avvolgimenti si genera un campo magnetico intorno al rotore.

La parte sinistra del rotore è respinta dal magnete di sinistra ed attirata da quello di destra.

Analogamente fa la parte in basso a destra. La coppia genera la rotazione.

La velocità di rotazione dipende da: tensione applicata, corrente assorbita dal rotore e carico applicato (chiamato *coppia di carico*). [9]

La precisione nel posizionamento a ciclo chiuso ed il controllo della velocità sono garantiti dall'encoder relativo montato sull'asse, che riesce a distinguere il numero di posizioni discrete per rivoluzione, analogo allo "*steps a rivoluzione*" dei motorini *Stepper*. Gli attuatori lineari e rotanti che utilizzano encoder restituiscono indirettamente il dato lineare, misurato tramite l'angolo di rotazione dell'asse filettato o del motorino.

1.1.2 Motorini Stepper (passo passo)

Una valida alternativa ai motorini DC sono i motorini di tipo *Stepper* (passo-passo) che funzionano sfruttando i principi base di attrazione/repulsione magnetica. La rotazione dell'albero motore è gestita come una successione di steps angolari di stati di equilibrio, inviando al motore una serie di impulsi di corrente secondo un'opportuna sequenza. L'avanzamento del moto è definito contando il numero di impulsi mandati al motorino, mentre la velocità risulta essere una funzione della frequenza dei suddetti segnali.

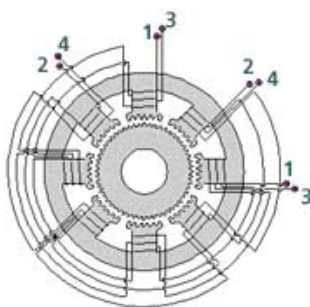


Figura 1-2: Sezione di un motorino Stepper



Figura 1-3: Struttura di un motorino Stepper

I motori passo-passo si dividono tradizionalmente in tre grandi gruppi: *motori a magnete permanente*, *motori a riluttanza variabile* e *motori ibridi*; questi ultimi hanno prestazioni migliori. In realtà la quasi totalità di quelli oggi reperibili sono proprio del terzo tipo.

Un motore ibrido è costituito da un *rotore* e da uno *statore*. Il rotore appare come una coppia di ruote dentate affiancate e solidali all'albero (i "denti" sono chiamati coppette) costituite da un nucleo magnetico (le due ruote sono permanentemente magnetizzate, una come NORD, l'altra come SUD) e le coppette in materiale ferromagnetico. Tra le due ruote è presente uno sfasamento esattamente pari ad $1/2$ del passo dei denti: il dente di una delle due sezioni corrisponde quindi alla valle dell'altra. Nel rotore non sono presenti fili elettrici e quindi manca completamente ogni connessione elettrica tra la parte in movimento e quella fissa.

Facendo scorrere corrente attraverso i poli dello statore del motore, si genera il momento di forza che consente al rotore di ruotare fino ad una posizione definita, e rimanervi fin quando non viene alimentato un altro polo.

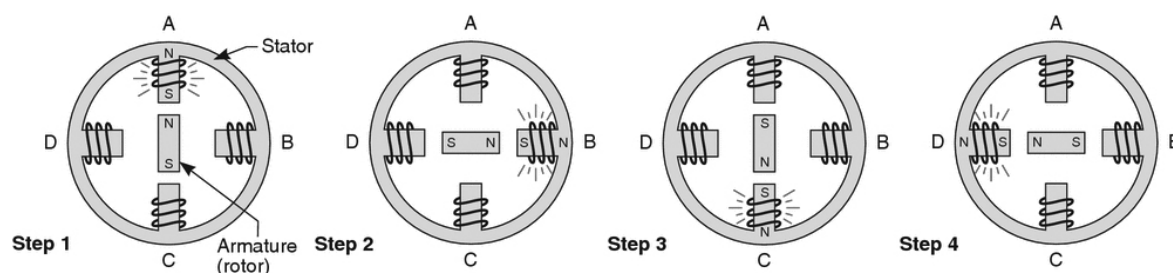


Figura 1-4: La sequenza di attivazione ai poli dello statore (A-B-C-D) provoca una rotazione in senso orario del rotore

I motorini Stepper sono spesso usati in cicli aperti (non retroazionati) e forniscono un'alternativa più economica ai costosi sistemi basati sui motori in continua, tuttavia non

sono indicati per applicazioni sotto vuoto, perché scaldano troppo e potrebbero contaminare la camera con degasaggi. [10, 11, 14]

1.1.3 Motorini piezoelettrici “inchworm”

I motorini di tipo *inchworm* utilizzano degli attuatori piezoelettrici (PZT) per muovere un’asta con precisione nanometrica. Il termine “inchworm” è stato coniato da Exfo, l’azienda che li ha brevettati, in quanto il moto ha una dinamica molto simile ai movimenti dei lombrichi e hanno limite massimo di corsa pari a un pollice (1 inch), cioè 2,54 cm.

I motorini inchworm sfruttano le proprietà di espansione dei tre componenti piezoelettrici (Figura 1-5) quando viene applicata una tensione ai loro capi; il movimento lungo una direzione avviene mediante l’attivazione sequenziale dei tre anelli: i due elementi esterni (indicati con i numeri 1 e 3) si comportano come una morsa (si espandono o si contraggono intorno l’asta); l’elemento centrale invece varia la propria lunghezza lungo l’asse a steps discreti (dell’ordine di qualche nanometro). La rampa che rappresenta l’espansione dell’elemento centrale (Figura 1-7), che può essere interrotta o invertita in qualunque punto del moto, contiene centinaia di posizioni intermedie tra le condizioni di minima e massima estensione e ciò permette di ottenere una risoluzione spaziale altrimenti non raggiungibile con i comuni motorini di tipo elettromagnetico. Per il motorino mostrato in Figura 1-6, burleigh TSE-150 con risoluzione dell’encoder pari a 50 nm, uno step corrisponde a 4 nm.

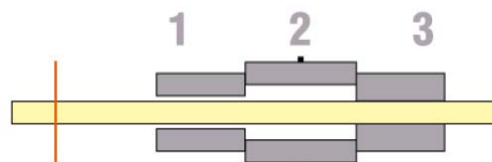


Figura 1-5: Disposizione dei tre elementi piezoelettrici lungo l’asta

Di seguito viene illustrata la sequenza di attivazione dei tre elementi per effettuare lo spostamento lungo una direzione.

L’elemento PZT 1 stringe l’asta quando viene applicata una tensione ai suoi capi. Quindi viene applicata una tensione all’elemento PZT 2 che ne causa l’espansione e

l'allontanamento del primo elemento e dell'asta che trattiene. Quando l'elemento centrale ha raggiunto l'espansione massima (o quella desiderata) viene applicata una tensione pure al terzo elemento che così stringe a sua volta l'asta. La tensione viene quindi rimossa dall'elemento 1 (facendogli rilasciare l'asta) e da quello centrale (che si contrarrà fino alla dimensione originale); a questo punto viene riattivato l'elemento 1, viene rilasciato l'elemento 3 e si può ricominciare con il moto. [12, 13]



Figura 1-6: Due attuatori burleigh TSE-150 montati su piastre in configurazione X-Y

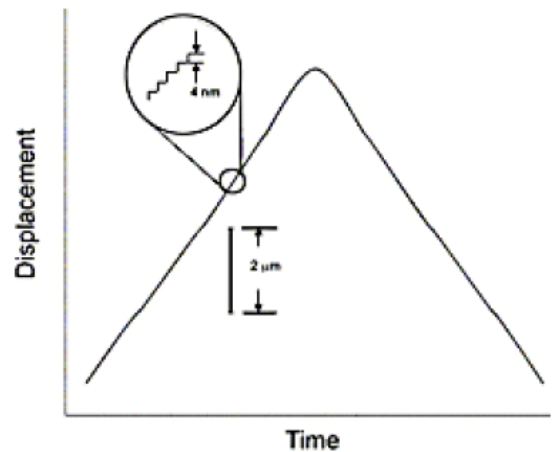


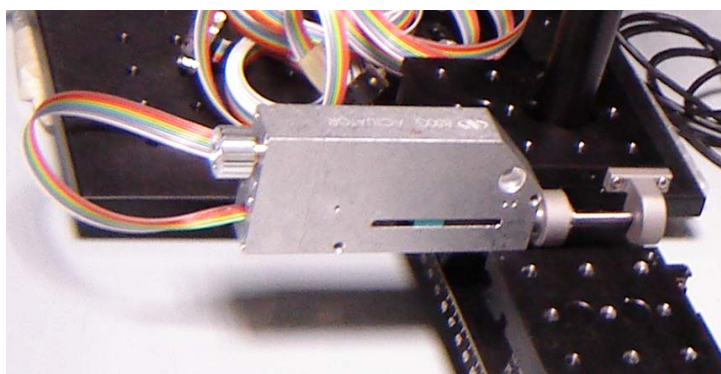
Figura 1-7: La rampa a steps discreti di 4 nm può benissimo essere approssimata ad una retta continua

1.2 Strumenti a disposizione

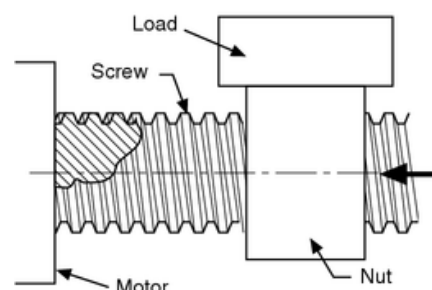
1.2.1 Attuatori Lineari Newport Serie 850G

Gli attuatori lineari Newport Serie 850G (Figura 1-8) possiedono un design versatile che ne permette l'utilizzo con una grande varietà di sistemi di traslazione. Il range della corsa è di 50 mm e i fine corsa sono realizzati con interruttori meccanici, che permettono di delimitare il moto dell'asse tra i due estremi. La struttura interna è composta da una vite senza fine (*lead screw*, Figura 1-9) che ruotando trasmette il moto lineare all'asse che serve a muovere le tavole. La precisione micrometrica viene garantita grazie ad una *gearbox* (rapporto moltiplicativo di ingranaggi) a bassa isteresi che consente di raggiungere risoluzioni altrimenti difficili da produrre con motorini standard.

Questi attuatori sono stati progettati appositamente per lavorare in condizioni di vuoto (a valori di 10^{-3} - 10^{-6} mbar). [15]



**Figura 1-8: Un attuttore lineare Serie 850G
montato su una piastra**



**Figura 1-9: La rotazione della vite
senza fine viene convertita
in moto traslazionale**

Per il lavoro svolto sono stati utilizzati due attuatori DC serie 850G ed uno serie 850G-HS (High Speed), per i quali sono riportate di seguito le caratteristiche

Tabella 1-1: Caratteristiche a confronto per gli attuatori lineari utilizzati

	850G	850G-HS
Risoluzione Encoder	0,051 μm	0,605 μm
Gearbox	262:1	22:1
Velocità Massima	500 $\mu\text{m/s}$	6000 $\mu\text{m/s}$
Isteresi	< 20 μm con carico di 1 Kg	
Accuratezza	< 0,1 % dello spostamento, cumulativo	
Ripetibilità	Migliore di 1 μm quando l'isteresi viene corretta dal controllore	
Encoder	Magnetico, 2 KHz	
Massimo carico	8 Kg	

1.2.2 Piastre di movimentazione

Si tratta di piastre in acciaio temperato che scorrono linearmente lungo delle guide mobili con sfere e cilindri che riducono al minimo l'attrito (Figura 1-10); al loro interno

sono presenti delle molle antagoniste che si oppongono all'estensione indotta dagli attuatori, in modo da ridurre considerevolmente l'isteresi.

Ciascuna piastra, attuata da un motorino o da una micrometro manuale, permette di effettuare traslazioni lungo un asse, con precisioni dell'ordine dei micron, inoltre possono essere assemblate per creare configurazioni che consentono il moto su uno, due e tre assi.

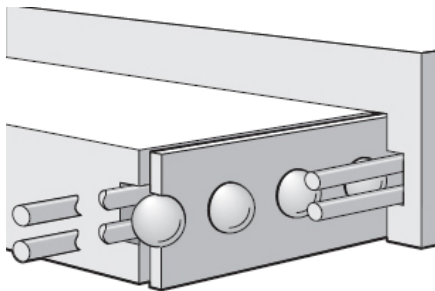


Figura 1-10: Lo scorrimento di sfere su cilindri riduce al minimo l'attrito

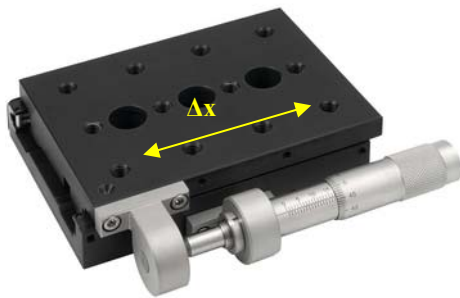


Figura 1-12: Piastra con micrometro



Figura 1-13: Piastra con motorino

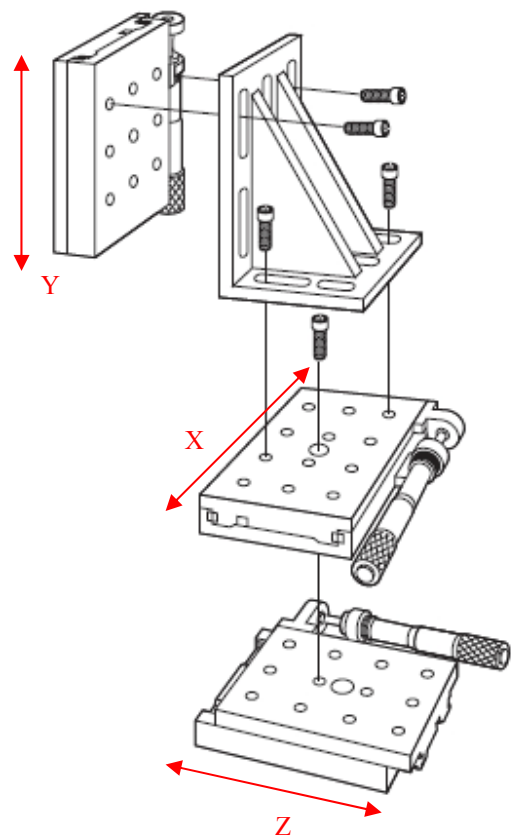


Figura 1-11: Configurazione X-Y-Z per il movimento lungo tre assi

La piastra di inclinazione/rotazione “Tip and Rotation” permette di effettuare correzioni angolari talvolta necessarie al posizionamento di un dato oggetto.



**Figura 1-14: Piastra per correzioni angolari
“Tip and Rotation”**

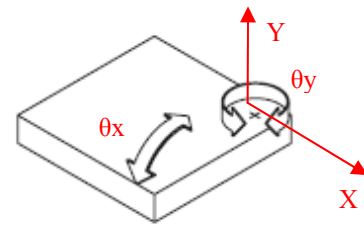


Figura 1-15: Angoli di rotazione

1.2.3 Controllore ESP 6000 DCIB

L'architettura *ESP (Enhanced System Performance)* utilizzata per il presente lavoro è composta dai seguenti componenti prodotti da *Newport Corporation*: *ESP6000 controller card*, *ESP6000 DCIB (DC Interface Box)* motor driver e attuatori lineari compatibili ESP serie 850 G.

L'ESP 6000 controller card è una periferica che si connette tramite interfaccia PCI alla scheda madre del computer, può essere programmata in Visual BASIC, Visual C/C++ e LabVIEW e consente di gestire la movimentazione di più motorini contemporaneamente, che possono essere sia passo-passo che lineari. La risoluzione del DAC (*Digital to Analogic Converter*) è 16 bit, la frequenza massima è 16 MHz (2.5 MHz per gli Stepper).
[16]



Figura 1-16: Il controllore ESP 6000 DCIB

L'ESP 6000 DCIB motor driver contiene al suo interno 4 moduli DC, per la gestione di 4 attuatori lineari indipendenti tra loro ed è ottimizzato l'uso degli attuatori della serie 850G. Il DCIB ha per ogni driver un relé controllato dalla scheda che abilita/disabilita il modulo DC, in modo da offrire il minimo di resistenza meccanica quando richiesto. Il DCIB è alimentato direttamente dalla scheda presente sul PC, ma per pilotare gli attuatori necessita di una fonte di alimentazione esterna. [17]

L'utente può ottenere la movimentazione desiderata scrivendo applicazioni software che richiamano le *API* (*Application Programming Interface*) e le *DLL* (*Dynamic Link Library*) fornite insieme al controllore.

E' possibile gestire la movimentazione per un vasto campo di applicazioni, che vanno dallo spostamento punto-punto fino all'interpolazione lineare/circolare; i comandi da inviare all'ESP 6000 possono essere sia ASCII che binari.

o Comandi ASCII

In questa configurazione il controllore mette a disposizione un buffer per la ricezione dei comandi: quando viene premuto il tasto *INVIO* (*Carriage Return, ASCII 13*), in seguito all'inserimento di una stringa di caratteri, questa viene interpretata come un comando ed eseguito. In caso di errore il corrispondente messaggio verrà salvato nel buffer degli errori.

La sintassi generale di un comando ASCII è: **xxAA**nn dove AA è la corrispondente sigla mnemonica, xx e nn gli eventuali valori opzionali.

Alcuni esempi:

1PA+30: porta l'asse 1 a 30 unità dalla posizione assoluta (Position Absolute)

2WS : aspetta che l'asse 2 si fermi (Wait Stop)

3PR-10: muove l'asse 3 di 10 unità in negativo (Position Relative)

I comandi vengono interpretati ed eseguiti dal controllore ed eseguiti una riga per volta, rispettando l'ordine di invio. Possono essere inviati contemporaneamente più comandi in una sola riga, purché separati con il punto e virgola [;], mentre per parametri multipli si fa uso di una virgola di separazione [,].

o **Comandi Binari**

I comandi binari sono adoperati per la programmazione in Visual C/C++, Visual Basic e LabVIEW tramite DLL, in questo caso si è aiutati nella ricerca di una data funzione dal nome stesso e dagli argomenti dei prototipi.

Alcuni esempi di funzione sono:

```
esp_enable_motor (long axis)
esp_get_error_string (char *ErrString, int *error, int *ServoTick)
esp_move_relative (long axis, double position)
esp_set_max_speed (long axis, float speed)
```

Nel complesso sono presenti oltre 100 comandi raggruppati nelle seguenti categorie: *Inizializzazione, Interfaccia per comandi ASCII, Selezione Generale, Funzioni per lo Stato del Sistema, Controllo di Moto e Posizione, Definizione di Traiettoria, Controllo di Flusso e Sequenza, Parametri per il Dispositivo, Programmazione, Filtri Digitali, Funzioni di Input Output.*

1.3 Controllo Retroazionato

Quando si parla di sistemi di movimentazione è importante soffermarsi sul tipo di controllo a catena chiusa che viene realizzato. L'architettura ESP implementa il controllo con un ciclo *PID (Proporzionale Integrale e Derivativo)*, un comune sistema di retroazione negativa impiegato nei sistemi di controllo automatici.

Il controllore acquisisce in ingresso un valore da un processo e lo confronta con un valore di riferimento. La differenza (o segnale di *errore*) viene quindi utilizzata per assestare alcuni dati in ingresso al processo per giungere alla sua stabilizzazione e ritornare al punto di lavoro desiderato.

Diversamente da un semplice controllore, il PID può assestare l'uscita di un sistema basandosi non solo sul segnale di errore, ma anche grazie alla sua "*storia*" (ovvero agendo con l'azione integrale del PID). I controllori PID non richiedono tecniche di progettazione

matematiche avanzate e sono di facile impostazione, diversamente da più complessi algoritmi di controllo basati sulla teoria del controllo ottimo e del controllo robusto. [18]

Le parti principali del controllore ESP sono l'interprete dei comandi, il generatore di traiettoria e il servo controllore, il quale si assicura che il motorino segua con buona approssimazione la traiettoria predefinita.

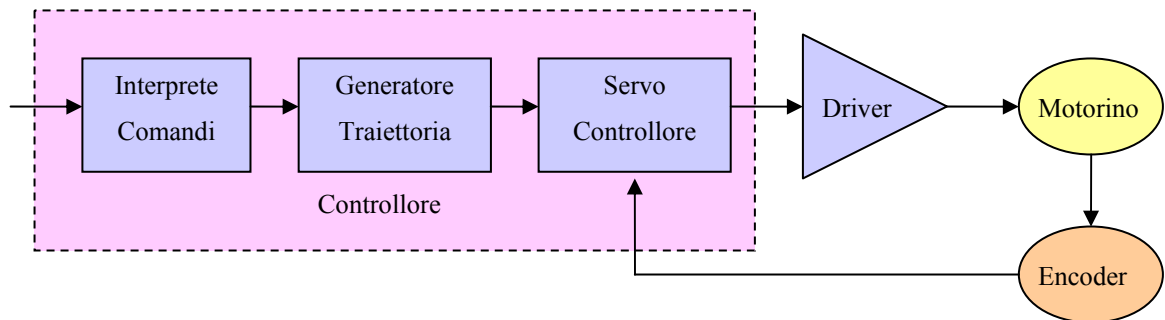


Figura 1-17: Schema a blocchi del controllore ESP.

L'encoder del motorino fornisce per ogni istante la posizione corrente

Per quel che concerne il *tuning* dei parametri, regolando i 3 guadagni del PID tramite regole empiriche è possibile ottenere controllori stabilizzanti veloci. La struttura del PID è essenzialmente un sistema lineare, composto da un parallelo tra un blocco proporzionale (un semplice gain), un blocco derivativo (un derivatore) e un blocco integratore. Esso è caratterizzato da una funzione di trasferimento del tipo:

$$f = K_p e + K_i \int e dt + K_d \frac{de}{dt}$$

dove e , K_p , K_i e K_d , sono rispettivamente: errore istantaneo, i termini proporzionale, integrale e derivativo.

Per correggere l'oscillazione degli assi si agisce principalmente diminuendo il fattore K_i , ma questo comporta uno squilibrio nella retroazione globale, rendendola troppo lieve. Aumentando il fattore K_p si riduce l'errore istantaneo, mentre K_d regola l'attenuazione in caso ci siano movimenti intorno la posizione di stop.

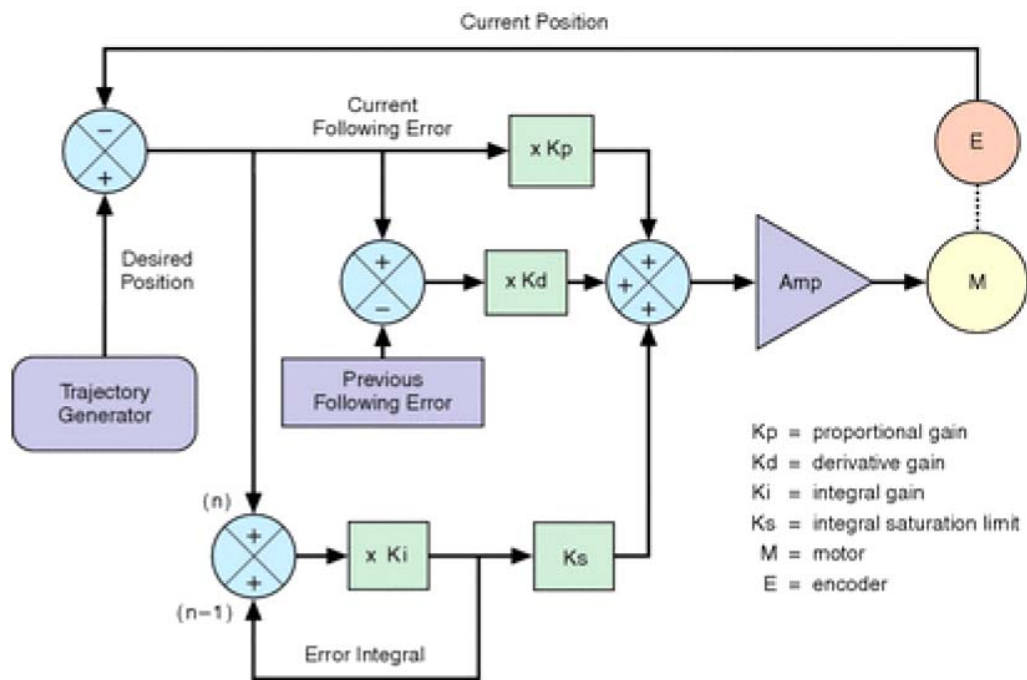


Figura 1-18: Schema a blocchi del loop PID interno al Servo Controllore

Il compito di un controllore PID è assicurarsi che il movimento dello strumento segua più fedelmente possibile la traiettoria ideale nel tempo; la deviazione dalla traiettoria ideale in un dato istante viene chiamata *errore istantaneo*.

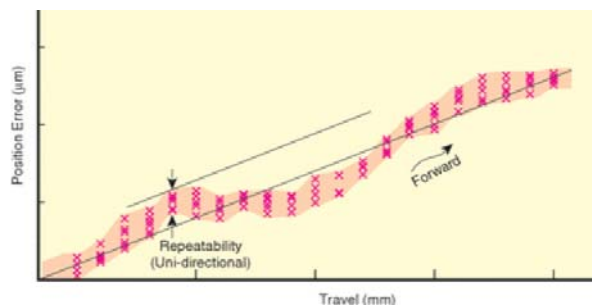


Figura 1-19: Errore istantaneo

Utilizzando un algoritmo di controllo PID è necessario che sia sempre presente un errore tra la traiettoria attuale e quella desiderata, in modo da generare un input di correzione. A causa di ciò deriva che sarà sempre presente un errore istantaneo non nullo. L'obiettivo di minimizzare l'errore istantaneo si raggiunge utilizzando un controllo di tipo *Feed Forward*.

Questo tipo di controllo agisce predicendo come il sistema si evolverà nel futuro, facendo correzioni basandosi su questa stima.

Tali correzioni sono generalmente implementate moltiplicando il valore della velocità desiderata per il fattore di velocità feed-forward V_f . La stessa tecnica può essere utilizzata per applicare un segnale di correzione all'accelerazione feed-forward (fattore A_f).

Questa correzione viene utilizzata per ridurre l'errore istantaneo medio durante i periodi di accelerazione e decelerazione che si hanno a inizio e fine moto; i cambiamenti di velocità repentini causano infatti grande accelerazione e di conseguenza grandi errori istantanei.

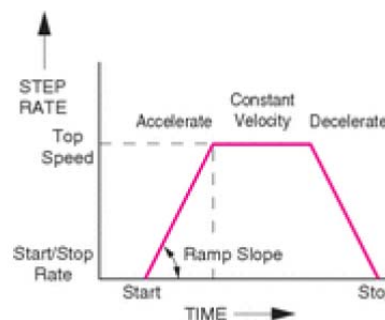


Figura 1-20: Il profilo velocità-tempo mostra come la velocità aumenta in modo lineare fino al valore desiderato, allo stesso modo del ritorno a zero; il grafico che ne risulta ha la forma di un trapezio.

I controllori avanzati permettono la modifica dei parametri di accelerazione/decelerazione in modo da raggiungere una configurazione ottimale.

Combinare la tecnica *Feed Forward* ad un PID, permette al controllo PID di correggere solo l'errore residuo proveniente del ciclo *Feed Forward*, migliorando così le prestazioni del sistema

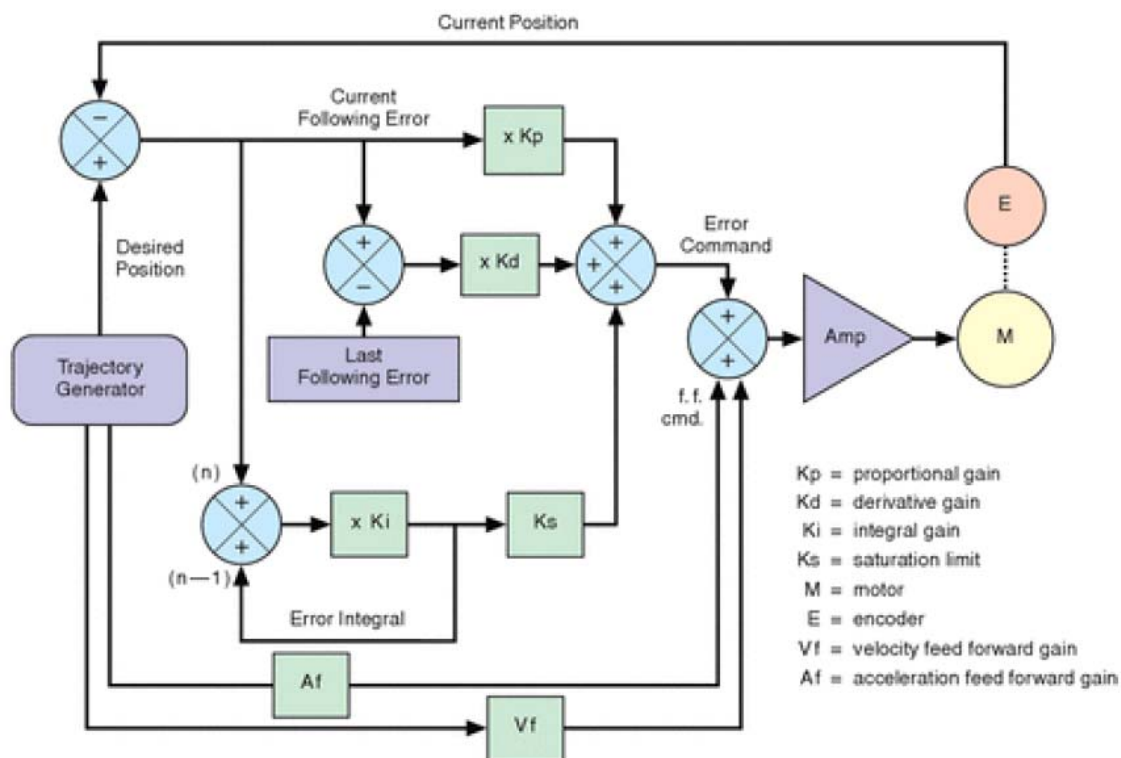


Figura 1-21: Schema a blocchi del loop PID con Feed Forward interno al Servo Controllore

L'ESP6000 riconosce per gli attuatori serie 850-G i valori di default da assegnare al controllore PID; in ogni caso, per raggiungere le migliori prestazioni, il sistema dovrebbe essere regolato per ogni applicazione specifica, tenendo conto del carico, dell'accelerazione e dell'orientamento degli assi. [16]

Tabella 1-2: Fattori controllore PID con Feed Forward e comprensione del loro valore

	Funzione	Valore troppo basso	Valore troppo alto
K_p	Determina la rigidità del ciclo	Ciclo troppo morbido con grandi errori istantanei	Ciclo troppo forte che causa oscillazioni
K_i	Riduce l'errore istantaneo durante movimentazioni lunghe e stop	La posizione di stop non viene raggiunta o mantenuta	Oscillazioni più ampie e a frequenza più bassa
K_d	Fattore di attenuazione, utilizzato per ridurre le oscillazioni	Crescono le oscillazioni non controllate causate dagli altri parametri	Oscillazioni più frequenti e grossi sbalzi di tensione sui motorini
V_{ff}	Riduce l'errore istantaneo durante il moto a velocità costante	Il dispositivo non raggiunge la traiettoria desiderata, l'errore istantaneo ha valore negativo durante la fase di velocità costante	La traiettoria desiderata viene anticipata, l'errore istantaneo è positivo durante la fase di velocità costante

A_{ff}	Riduce l'errore istantaneo durante le fasi di accelerazione e decelerazione	Il dispositivo non raggiunge la traiettoria desiderata, l'errore istantaneo ha valore negativo durante la fase di accelerazione	La traiettoria desiderata viene anticipata, l'errore istantaneo è positivo durante la fase di accelerazione
-----------------------	---	---	---

2 Rivelatori di particelle cariche

I metodi di identificazione comunemente usati per caratterizzare i prodotti delle reazioni nucleari si basano sull'utilizzo della tecnica ΔE - E , che fa uso di due rivelatori posti in successione lungo la traiettoria delle particelle (in una configurazione chiamata *telescopio*).

Il primo (ΔE) essendo abbastanza sottile viene facilmente attraversato dagli ioni incidenti, che rilasciano sul secondo rivelatore una quantità di energia dipendente dal loro numero atomico Z .

Questa tecnica consente di effettuare una identificazione dell'elemento corrispondente a ciascuno ione, nonché di misurarne l'energia cinetica, mediante una misura simultanea della perdita di energia su entrambi i rivelatori. Risulta alquanto evidente che in questa tecnica lo spessore del primo stadio rappresenta un aspetto cruciale.

L'utilizzo di rivelatori a gas di ionizzazione (*ionization gas chambers*) offre il vantaggio di poter utilizzare piccoli spessori a pressioni del gas molto basse, ma necessitano di una regolazione molto complessa e sono molto ingombranti.

La soluzione alternativa proviene dall'utilizzo di telescopi al silicio (*silicon telescopes*), ma produrre singoli rivelatori ΔE dello spessore di pochi micron con questa tecnica risulta molto costoso, data la loro estrema fragilità.

Una soluzione per risolvere il problema della fragilità, utilizzando tuttavia lo stesso tipo di rivelatore al silicio, è l'utilizzo di telescopi con la struttura ΔE direttamente impiantata nel substrato dello stadio E ; in questo modo si possono produrre stadi ΔE molto sottili (1-3 μm) piuttosto che stadi separati il cui spessore non potrebbe scendere al di sotto di 10 μm .

All'interno del capitolo viene presentata la famiglia di rivelatori di particelle cariche al silicio presa in esame per lo studio di caratterizzazione, documentando l'evoluzione dei vari prototipi prodotti e i principi fisici di funzionamento.

2.1 I prototipi progettati da INFN-LNS e prodotti da ST-Microelectronics

Da alcuni anni sono stati sviluppati dalla collaborazione tra *INFN* e *ST-Microelectronics* nuovi modelli di rivelatori al silicio, che realizzano un telescopio monolitico. Questi telescopi consistono di uno strato ΔE ultra sottile (1-2 μm) ottenuto tramite impiantazione ionica su di uno stadio E spesso 500 μm .

Le regioni p^+ sono state realizzate con un'implantazione a bassa energia di Boro, seguita da diffusione ad alta temperatura (1050 $^{\circ}\text{C}$) fino ad una profondità di 2 μm ; la regione p^+ che funge da elettrodo di massa comune per i due stadi, è stata ottenuta attraverso l'implantazione ad alta energia ($\approx 900 \text{ keV}$) di Boro su un bulk con resistività pari a 5000 $\Omega\cdot\text{cm}$. Successivamente sono stati creati i contatti elettrici su entrambi i lati.

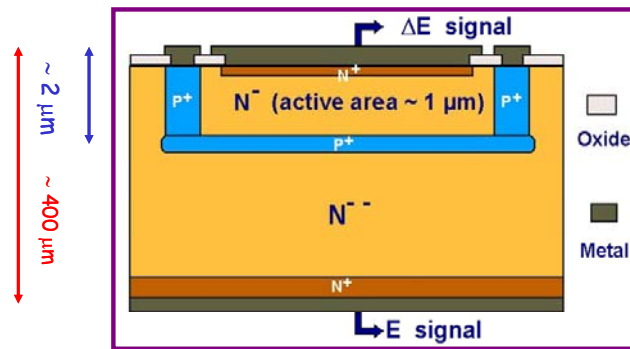


Figura 2-1: Sezione (non in scala) del substrato dopo il trattamento

Vengono di seguito riportate le rappresentazioni più significative dell'evoluzione dei prototipi prodotti durante il periodo di ricerche.



Figura 2-2: Primo prototipo.
Regione attiva 4x4 mm^2 [4]

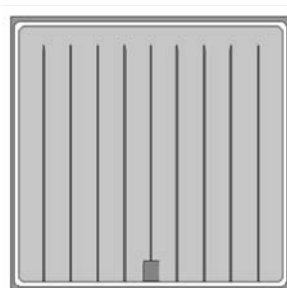


Figura 2-3: Regione attiva 20x20 mm^2 e anodo comune per velocizzare la raccolta delle cariche [5]

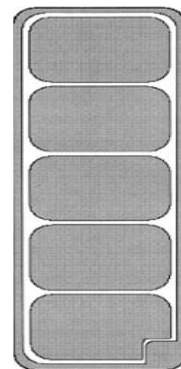


Figura 2-4: Regione attiva 15x4 mm^2 . 5 Strips indipendenti su uno stadio E comune [6]

L'ultimo modulo realizzato consiste in cinque strips indipendenti ΔE , impiantante nello stesso stadio E , che costituiscono l'elemento primario per il *multirivelatore MONTE* (*MONolithic TElescopes*), che è stato utilizzato intensivamente e con estrema efficacia in diversi esperimenti a bassa e media energia, sia con fasci radioattivi che stabili.

Questo tipo di configurazione consente di identificare la particella incidente in numero atomico (Z), di misurarne l'energia cinetica e la posizione di impatto con una precisione micrometrica.

Dai risultati dei test effettuati si è potuto notare che oltre ai segnali ΔE - E desiderati è fortemente presente un segnale di tipo induttivo in uscita sul canale ΔE , questo segnale è dovuto alla resistenza finita dell'elettrodo di massa comune impiantato che è dell'ordine di qualche migliaio di ohm.

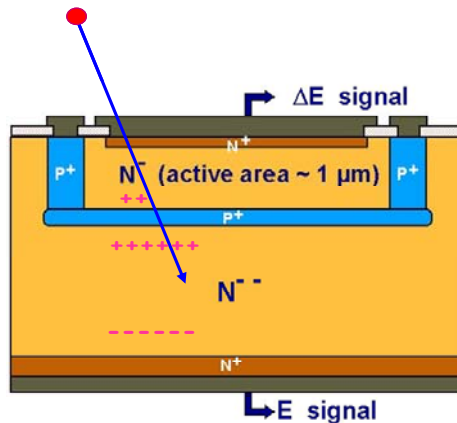


Figura 2-5: La carica creata nello stadio E, immagazzinata dall'elettrodo di massa, cambia il campo elettrico dello stadio ΔE e crea un segnale indotto molto veloce, più ampio del segnale reale

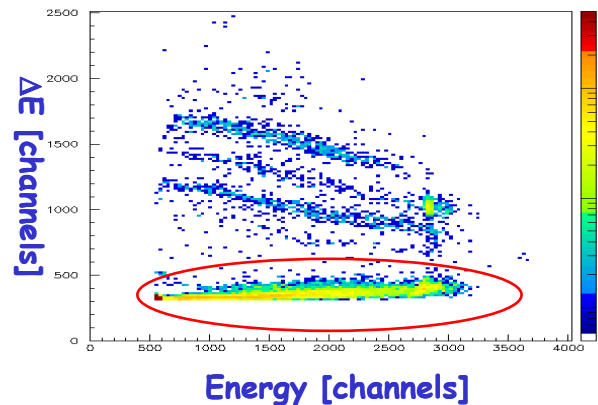


Figura 2-6: Il segnale indotto, cerchiato in figura, è chiaramente visibile

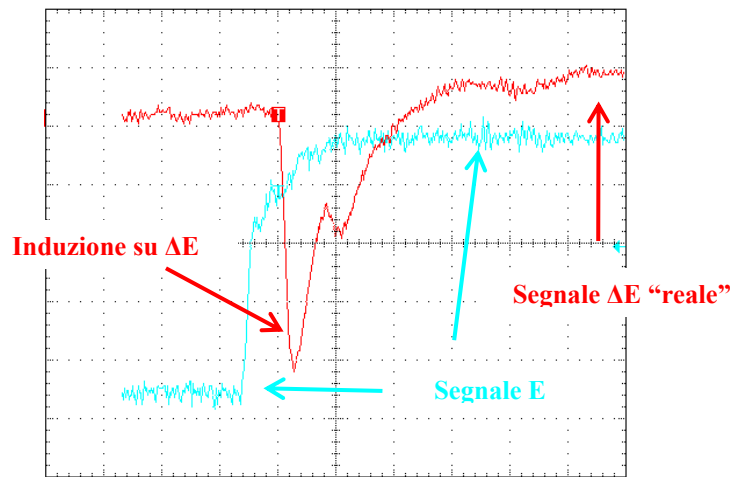


Figura 2-7: Andamento dei segnali ΔE -E all'oscilloscopio

Si è pensato dunque di misurare proprio l'ampiezza di questo segnale indotto, in modo da cercare di utilizzarlo come segnale contenente informazione.

Esso appare essere dipendente dalla posizione di incidenza sulla strip colpita, seguendo un andamento approssimativamente lineare, in funzione della distanza dal contatto di massa comune.

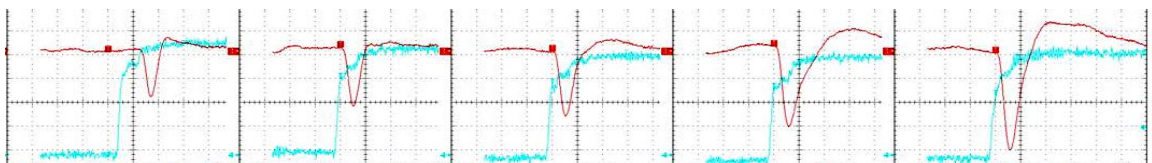


Figura 2-8: L'ampiezza del segnale induzione su una strip misurata in diversi punti di impatto

Studiando i dettagli di questo effetto è possibile utilizzarlo per costruire un telescopio monolitico sensibile alla posizione, dove il numero della strip colpita rappresenta la coordinata Y, mentre la posizione X viene determinata dal segnale indotto presente sulla strip stessa.

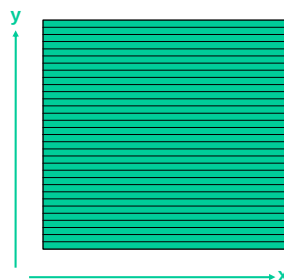


Figura 2-9: Schema geometrico di un rivelatore sensibile alla posizione.

2.2 Presentazione delle caratteristiche dei prototipi sottoposti a misurazione

Seguendo le osservazioni scaturite da questi studi è stato realizzato un nuovo telescopio monolitico a strip, di dimensioni $10 \times 10 \text{ mm}^2$, che consiste in 16 (o 32) strips verticali ΔE impiantate su uno stadio E comune. In questa struttura, la posizione X verrà fornita dalla strip colpita, mentre la posizione Y può essere ricavata dal segnale indotto lungo ogni strip.

Per testare la validità di questa nuova configurazione, sono stati realizzati 3 prototipi a geometria e parametri di produzione diversi (identificati rispettivamente dalle sigle: R328-1, R328-2, R328-3), con lo scopo di trovare la migliore configurazione per ricavare la posizione del segnale.

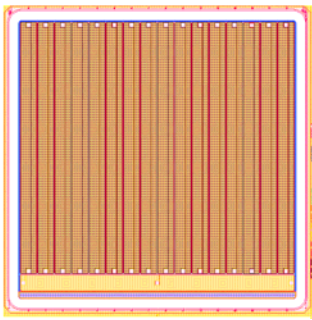


Figura 2-10: R328-1

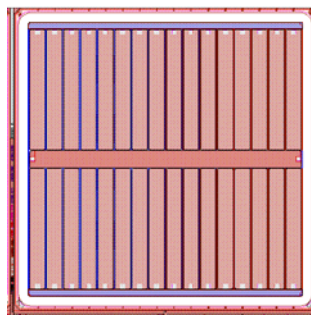


Figura 2-11: R328-2

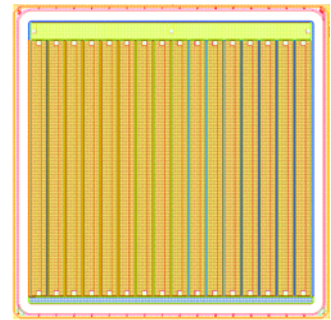


Figura 2-12: R328-3

Di seguito verranno riportate le caratteristiche del solo prototipo R328-1

- o Area totale: 100 mm^2
- o Area ΔE : 71 mm^2
- o 16 strip verticali, numerate da 0 a 15, di area: $0,536 \times 8,255 \text{ mm}^2$ (460 pF) di cui il 3,4 % metallizzata con Ti+Al
- o 1 strip orizzontale (massa comune) di area: $0,537 \times 9,012 \text{ mm}^2$ (500 pF) interamente metallizzata con Ti+Al

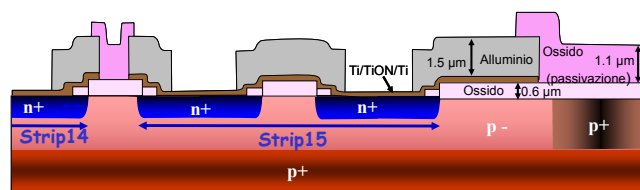


Figura 2-13: Sezione (non in scala) del rivelatore

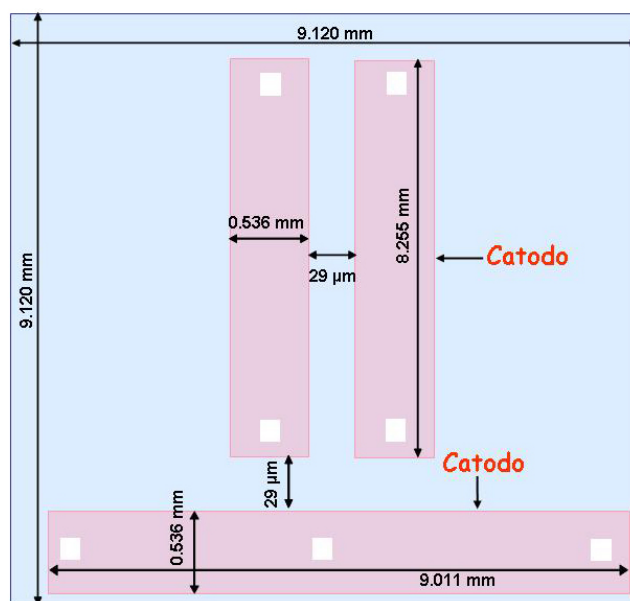


Figura 2-14: Dimensioni

I test sono orientati sullo studio del comportamento generale dei rivelatori: tensione di lavoro ottimale, corrente inversa, risoluzione, e una ricerca mirata all'uso come rivelatori sensibili alla posizione. Ogni strip sarà irradiata con un fascio di Carbonio in entrambe le direzioni X e Y in diversi punti, per ottenere un'accurata risposta in funzione della posizione.

Per caratterizzare la dipendenza del segnale indotto rispetto la posizione, è necessario studiare la sua ampiezza per gli urti delle particelle in punti differenti dentro la stessa strip.

Ci si propone di utilizzare un microfascio di ^{12}C , del diametro di $50\text{ }\mu\text{m}$, a 50 MeV (energia) e 5 nA (corrente), per irradiare una strip in 8 punti differenti lungo Y (con steps di $1000\text{ }\mu\text{m}$) e 3 differenti posizioni lungo l'asse X (con steps di $130\text{ }\mu\text{m}$), a posizione alternata lungo Y. Poiché ogni prototipo ha 4 regioni differenti con differenti parametri di produzione è necessario irradiare 4 strips per ogni prototipo. [7]

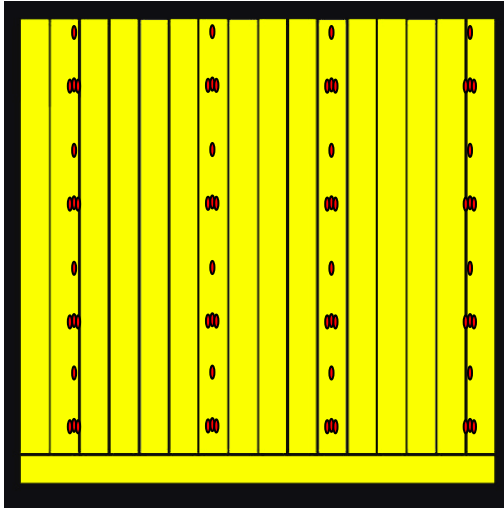


Figura 2-15: Punti in cui verranno effettuate le misure sulle strips

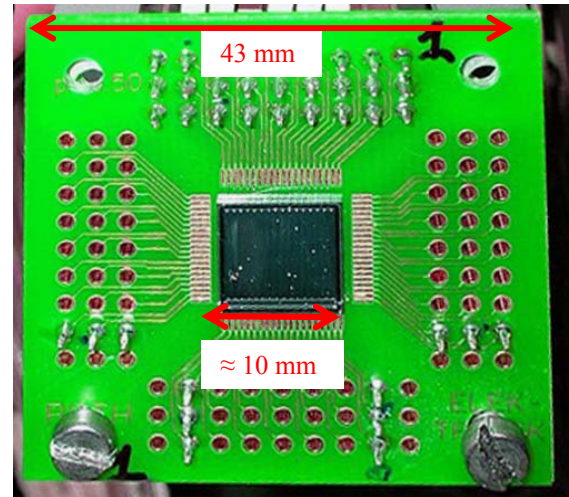


Figura 2-16: Il prototipo R328-1 montato sul PCB

3 Software

Il terzo capitolo presenta il LabVIEW, ambiente di programmazione visuale scelto per la realizzazione del software di gestione. Dopo una breve illustrazione delle potenzialità offerte al programmatore vengono descritti i vari componenti implementati nel programma, giungendo fino alla struttura a basso livello che rappresenta la collezione e il raggruppamento dei vari dati.

Quindi la descrizione del programma sviluppato, giustificando le scelte progettuali che hanno condotto ad operare verso una direzione, la presentazione del grafico che permette all'operatore di supervisionare il movimento come se fosse a contatto visivo con la struttura comandata da remoto.

3.1 Scrivere un programma in LabVIEW

I programmi di LabVIEW sono chiamati *VI* (*Virtual Instruments*), poiché nell'aspetto e nel funzionamento sono simili a strumenti fisici, come oscilloscopi e tester.

LabVIEW contiene un ampio set di strumenti per acquisire, analizzare, visualizzare e memorizzare i dati, oltre a strumenti che aiutano nella risoluzione dei problemi del codice.

In LabVIEW si può creare un'interfaccia utente, detta *Front Panel*, con alcuni controlli e indicatori. I controlli sono le manopole (*knob*), i pulsanti da premere (*push button*), i selettori (*dial*) e altri dispositivi di input. Gli indicatori sono i grafici, i led e altri displays. Una volta creata l'interfaccia utente, è possibile aggiungere il codice utilizzando subVI's e strutture che controllino gli oggetti del pannello frontale. Il *Block Diagram* contiene questo codice.

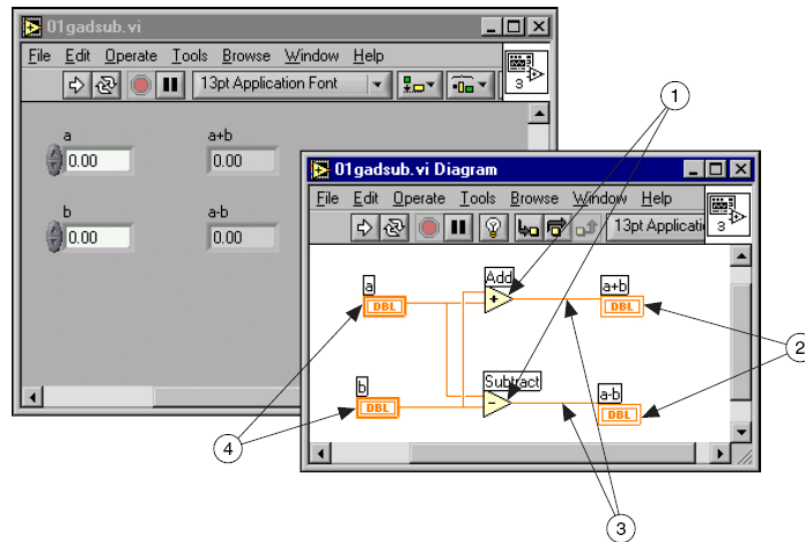


Figura 3.1: 1 Nodi, 2 Terminali degli indicatori, 3 Connessioni, 4 Terminali dei controlli

LabVIEW può essere utilizzato per comunicare con un hardware come quello per l'acquisizione dei dati e immagini, con i dispositivi di controllo del movimento e con strumenti GPIB, PXI, VXI, RS-232 e RS-485.

Il pannello frontale è l'interfaccia utente di un VI. Viene costruito utilizzando controlli e indicatori, che sono rispettivamente i terminali interattivi di input (*entrata*) e output (*uscita*) del VI. I controlli e gli indicatori simulano i dispositivi di input di uno strumento fisico, si trovano nella *Controls Palette* e forniscono i dati allo schema a blocchi del VI.

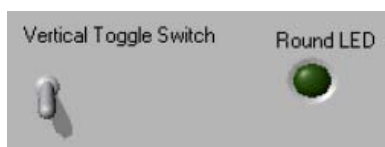


Figura 3.2: Interruttore Booleano e led

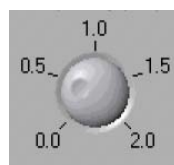


Figura 3.3: Manopola



Figura 3.4: Indicatore grafico semicircolare

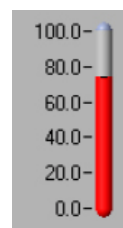


Figura 3.5: Indicatore lineare

Lo schema a blocchi contiene il codice sorgente grafico grazie al quale il VI può essere eseguito. Gli oggetti del pannello frontale appaiono come terminali sotto forma di icona nello schema a blocchi.

3.1.1 Tipi di dati

Man mano che si costruisce lo schema a blocchi per un VI, si può notare che i terminali per gli oggetti del pannello frontale appaiono di un colore diverso. Il colore e il simbolo di ciascun terminale indicano il tipo di dati del controllo o dell'indicatore. I colori indicano anche i tipi di dati delle connessioni (*wires*), degli input e degli output.

Il colore degli input e degli output sui VI indicano quali tipi di dati vengono accettati o restituiti. I terminali dei controlli hanno un bordo più spesso rispetto a quello dei terminali degli indicatori. Sui terminali del pannello frontale appaiono inoltre delle frecce, a indicare se il terminale è un controllo o un indicatore.

I tipi di dati indicano quali oggetti, input e output si possono connettere.

Tabella 3-1: Rappresentazione grafica delle connessioni

Tipo connessione	Scalare	Array 1D	Array 2D	Colore
Numerico Intero				Blu
Numerico Float				Arancione
Booleano				Verde
Stringa				Rosa

3.1.2 Arrays

Un *array* è un raggruppamento di elementi che sono dello stesso tipo. Un array ha una o più dimensioni e fino a 231 elementi per dimensione.

Si accede ad ogni elemento dell'array tramite il suo indice, che va da 0 a N-1, dove N indica il numero degli elementi dell'array.

Tabella 3-2: Array di 5 elementi numerici

Indice	0	1	2	3	4
	1.2	3.2	8.2	4.8	2.5

Per identificare un elemento di un array bi-dimensionale (2D) sono necessari due indici, uno per la riga e uno per la colonna, entrambi che partono da 0.

Tabella 3-3: Array 4x5 di caratteri

Indici	0	1	2	3	4
0	<i>a</i>	<i>j</i>	<i>r</i>	<i>H</i>	<i>X</i>
1	<i>u</i>	<i>e</i>	<i>S</i>	<i><</i>	<i>Q</i>
2	<i>b</i>	<i>%</i>	<i>4</i>	<i>f</i>	<i>F</i>
3	<i>£</i>	<i>\$</i>	<i>@</i>	<i>A</i>	<i>g</i>

Le funzioni numeriche di LabVIEW sono *polimorfiche*. Ciò vuol dire che i dati in ingresso a queste funzioni possono essere di tipo differente, scalari e arrays. Ad esempio si può sommare uno scalare ad un array o due arrays tra loro.

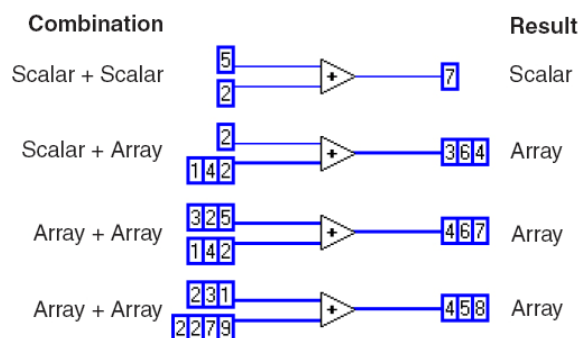


Figura 3.6: Funzioni numeriche polimorfiche

3.1.3 Clusters

Per raggruppare insieme informazioni di tipo diverso si usa la funzione *bundle* (raggruppa). Questa crea un tipo di dato che in LabVIEW viene chiamato *Cluster* (del tutto analogo alle Strutture dati di altri linguaggi come il C).

Un Cluster è una struttura dati che combina uno o più componenti in un nuovo tipo di dato. A differenza di un array, dove tutti i componenti sono dello stesso tipo, i componenti che compongono un cluster possono contenere differenti tipi di dati, come booleano, stringhe e dati numerici.

Nel Block Diagram, un connettore che trasporta i dati di un Cluster può essere visto come un fascio di connettori più piccoli, ogni “filo” rappresenta un differente elemento che compone il cluster.



Figura 3.7: Le connessioni dei clusters rappresentano fasci di tipi di dati diversi

Per accedere nuovamente ai dati immagazzinati in un cluster si deve procedere con l'operazione inversa, tramite la funzione *unbundle*.

3.1.4 Strutture Dati in LabVIEW

In LabVIEW le strutture dati comuni ai linguaggi di programmazione testuali vengono rappresentate graficamente nel Block Diagram con dei rettangoli, tutto ciò che è contenuto all'interno di queste strutture viene eseguito se la condizione che vi fa accedere è verificata.

o Ciclo *while*:

Simile al ciclo "Do" o al ciclo "Repeat-Until" dei linguaggi di programmazione testuali: un ciclo while esegue un sottodiagramma fin quando una condizione logica è verificata



Figura 3.8: While loop

o Ciclo *for*:

Ripete la parte del Block Diagram che racchiude per un numero determinato di volte, l'indice N che viene impostato dal programmatore, secondo il seguente modo:

```
{ for i = 0 to N-1
    execute diagram inside loop }
```

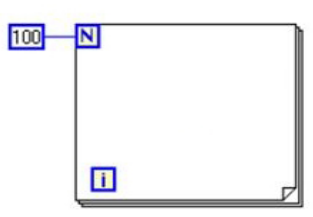


Figura 3.9: For loop

o *Case Structure:*

E' analogo alla condizione if ... then ... else o al costrutto "Case" dei linguaggi testuali. La struttura si presenta graficamente come un mazzo di carte, ognuna con il proprio sottodiagramma. Solo un caso è visibile per volta, quello che viene eseguito in funzione del valore presente nella condizione logica in ingresso.

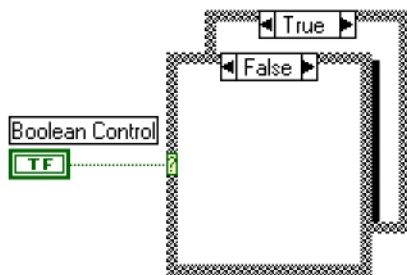


Figura 3.10: Case Structure analogo alla condizione if ... then ... else (True-False)

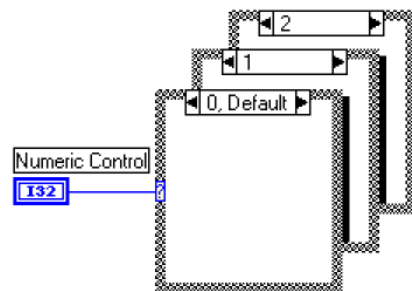


Figura 3.11: Case Structure analogo alla condizione Case ... (1, 2, ...)

o *Sequence Structure:*

Il bordo di tali strutture appare come la pellicola fotografica e indica che il codice contenuto nei diagrammi viene eseguito nella sequenza programmata.

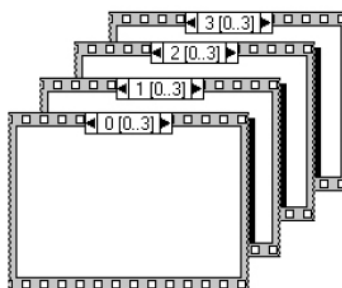


Figura 3.12: Le pagine della Sequence Structure appaiono come fotogrammi in successione

o *Shift Registers:*

Vengono utilizzati con cicli *While* e *For* per trasportare valori da una iterazione all'altra. Si presenta come una coppia di terminali posti sui lati verticali del bordo del ciclo.

Il terminale posto sulla destra immagazzina i dati al completamento dell'iterazione, questi dati sono trasferiti al terminale di sinistra dell'iterazione successiva. Possono contenere ogni tipo di dato, booleano, numerico, stringa e array.

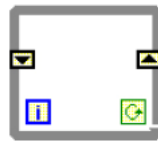


Figura 3.13: Lo Shift Register a destra invia il dato a quello di sinistra quando comincia un nuovo ciclo

o *Local Variables:*

Nel Front Panel è presente solo un terminale per ogni oggetto; quando è necessario accedervi o aggiornarne uno da un altro posto è comodo l'utilizzo delle Local Variables. Questi elementi permettono di passare i dati tra le varie strutture del codice a blocchi quando non è possibile connetterle in modo diretto e possono essere impostate in modalità *Read* e *Write*, quelle cioè che permettono la lettura o la scrittura dei dati che le compongono (si può notare la differenza dal bordo che le racchiude). Assumono lo stesso colore del dato che le compone.



Figura 3.14: Local Variables

3.1.5 Tools, Controls e Functions Palettes



Figura 3.15:
Tools Palette



Figura 16:
Controls Palette



Figura 17:
Functions Palette

- o La *Tools Palette*, accessibile sia dal Front Panel che dal Block Diagram, mette a disposizione: cursori di selezione, caselle di testo, rocchetto di filo (lo strumento utilizzato per connettere i vari blocchi), sonde (*probes*) e interruzioni (*breakpoints*) da utilizzare in fase di debug.
- o La *Controls Palette* è accessibile solo dal Front Panel. Mette a disposizione sia controlli e indicatori da utilizzare come terminali di input/output (pulsanti, manopole, led, sliders), sia grafici di tipo *Chart* o *XY* (per simulare ad esempio un oscilloscopio).
- o La *Functions Palette*, accessibile solo dal Block Diagram, contiene tutte le funzioni matematiche (aritmetiche, trigonometriche), logiche (algebra di Boole), di comparazione/creazione/modifica stringhe, e tutto quanto è necessario per processare i dati. E' inoltre possibile creare, modificare o accedere a valori indicizzati di arrays e clusters, oltre a scegliere le strutture cicliche (*while*, *for*, ...).

Dal pulsante in basso a destra si accede alle librerie personalizzate dell'utente, da dove, ad esempio, era possibile accedere ai subVI's forniti da Newport Corporation per il controllore ESP 6000.

3.2 VI's Newport per la gestione dei parametri.

Newport Corporation, oltre a sviluppare una serie di funzioni e comandi per programmare il controllore ESP6000, fornisce un set molto vasto di VI scritti per l'utilizzo del controllore in ambiente LabVIEW; si tratta essenzialmente di piccoli programmi che al loro interno utilizzano le funzioni descritte precedentemente come DLL, e permettono di gestire il flusso di dati in perfetta sintonia con il modello di programmazione di LabVIEW.

Ad esempio prendendo il VI *"Move to relative position"*, al suo interno viene utilizzata la funzione `esp_move_relative (long axis, double position)`, già presentata, e come la funzione prende in ingresso i valori relativi all'asse da movimentare e dello spostamento da comandare all'attuatore. Questi dati sono inoltre disponibili in uscita e la presenza del flag `execute` permette di abilitare/disabilitare questa funzione.

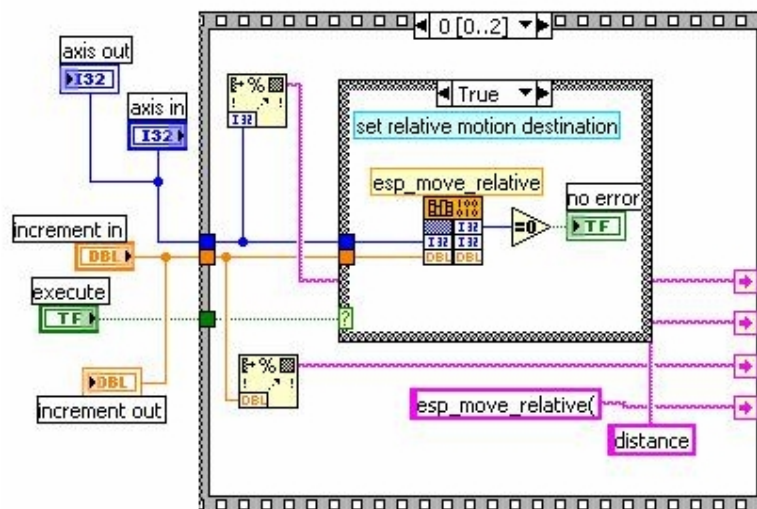


Figura 3.18: Block Diagram del subVI

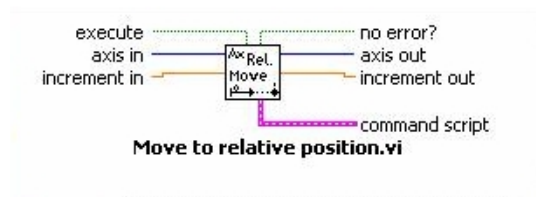


Figura 3.19: Icona che rappresenta il subVI nel Block Diagram

Risulta evidente che il lavoro di progettazione di uno strumento software eseguito con questo metodo totalmente grafico, senza scrivere neanche una riga di codice, permette al programmatore e all'utilizzatore finale una visione globale del sistema, seppur complesso, grazie proprio all'utilizzo di strutture annidate tra loro.

Inoltre, nella modalità di esecuzione alla ricerca di errori nel codice (*debug*), si può seguire il flusso di dati in modo rallentato, con la possibilità di seguire visualmente il migrare di un dato lungo il percorso di un connettore. [20]

3.3 Interfaccia Utente

Il programma sviluppato nasce dalla necessità di effettuare spostamenti micrometrici su uno o più assi giacenti sul piano di un rivelatore di particelle cariche da irraggiare con un fascio di carbonio (^{12}C).

Per la progettazione dell'interfaccia grafica, che permette all'operatore di gestire gli spostamenti, si è prestata grande attenzione affinché risultasse intuitiva nell'utilizzo ma, allo stesso tempo, ricca di tutte le funzionalità necessarie ad un completo controllo del sistema.

Per rispondere a questi requisiti è stato necessario conoscere a fondo le condizioni del sistema, le funzionalità di controllore e attuatori, le esigenze varie di precisione negli spostamenti. [3]

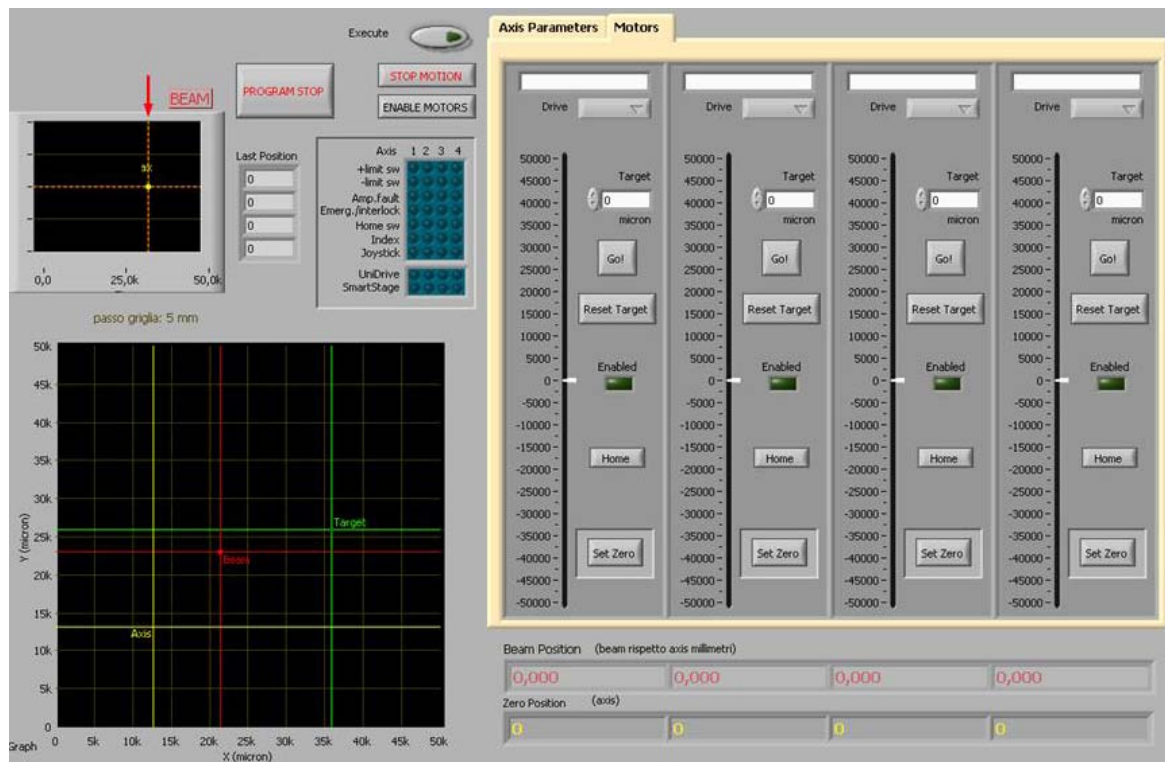


Figura 3.20: Interfaccia utente del programma sviluppato:

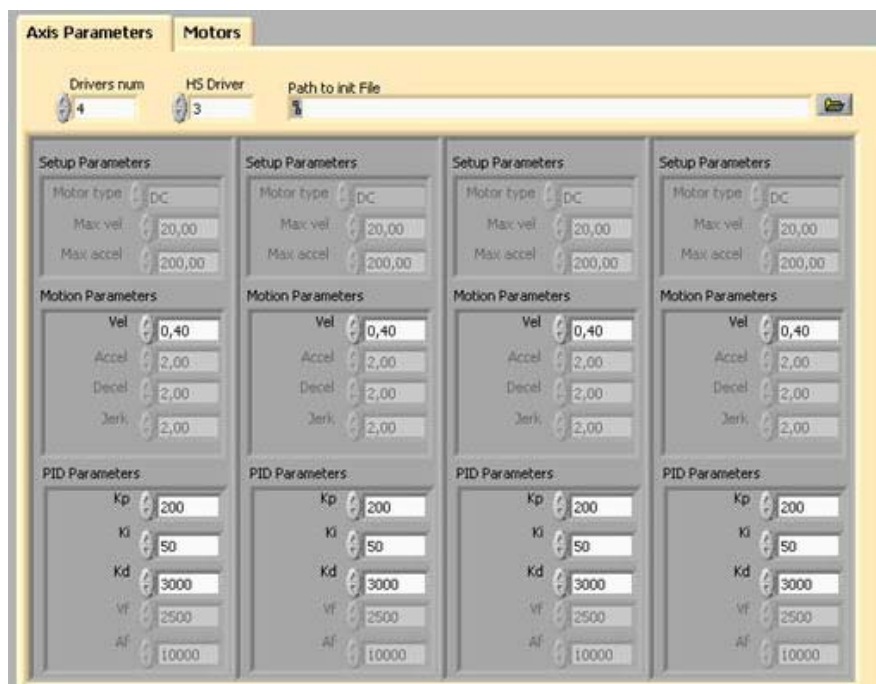
a sinistra sono presenti i grafici di visualizzazione,

a destra i comandi per il moto lungo gli assi

L'interfaccia utente si presenta con un *Tab Menu* sulla destra con due selettori, che consentono di accedere alle pagine "*Axis Parameters*" e "*Motors*".

Dalla pagina *Axis Parameters* è possibile impostare i valori di *Setup*, *PID*, e *Motion* per ogni driver dell'ESP6000. Nonostante siano stati utilizzati solamente tre attuatori, la pagina è provvista di quattro set di comandi, perché il controllore ESP6000 ha quattro drivers disponibili.

Si può comunque scegliere il numero dei drivers da utilizzare agendo sulla casella "*Drivers num*": se viene introdotto un numero inferiore a 4, il set di comandi in eccesso risulterà disabilitato.



**Figura 3.21: Il Tab Menu “Axis Parameters”
permette la regolazione dei valori del PID**

L’interfaccia è stata pensata, quindi, in modo da essere versatile nell’utilizzo in altre applicazioni oltre a quella sviluppata in particolare, anche se risulta ottimizzata per questa. A conferma di ciò è da notare infatti la presenza di una casella, denominata “*HS Driver*”, che identifica il driver su cui è installato l’attuatore 850-HS (High Speed).

Questo dato ricopre un’importanza notevole poiché attuatori lineari di diverso tipo hanno caratteristiche differenti, e quindi diversi valori da impostare per il controllore PID e diverso valore di velocità massima:

Tabella 3-4: Valori per il PID di default per gli attuatori lineari utilizzati

	850-G	850-HS
Kp	200	500
Ki	50	300
Kd	3000	1000
Velocità	0,4 mm/s	3 mm/s

Il numero impostato dentro questa casella (di default indica il driver numero 3) viene preso in considerazione nel momento in cui vengono inviati i valori al PID, se si tratta

dell'attuatore tipo High Speed vengono inviati i valori corrispondenti, in caso contrario quelli di default.

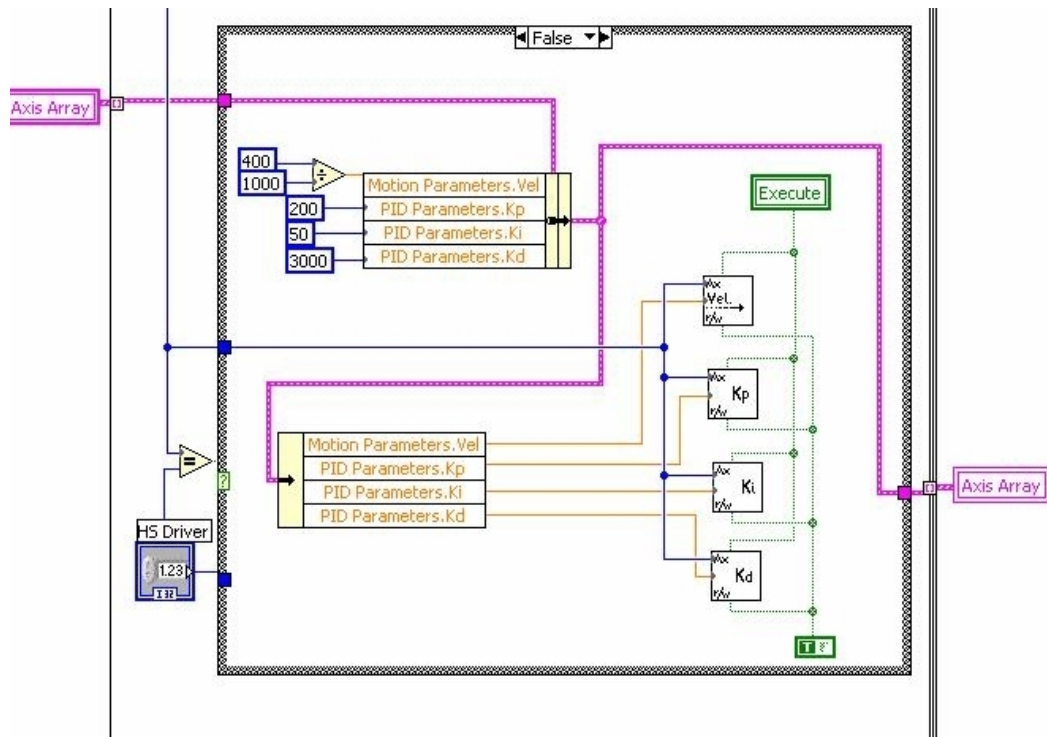


Figura 3.22: Axis Array, in ingresso ad un ciclo for, viene scomposto nei suoi quattro elementi e quindi viene aggiornato nei suoi valori tramite la funzione “Bundle by Name”

La pagina *Motors* contiene i comandi che permettono la movimentazione, quindi pulsanti e valori che l'utente andrà a impostare.

Sotto questo Tab Menu sono presenti due array numerici, “*Beam Position*” e “*Axis Position*”, che mostrano le posizioni in tempo reale degli assi e del punto preso come riferimento di origine per gli assi.

La parte a sinistra dello schermo contiene quattro pulsanti

- o “*Execute*”: tasto che avvia il programma; la condizione booleana *Execute* (True/False) viene controllata quasi ovunque dai subVIs del programma
- o “*Program Stop*”: Avvia la procedura di uscita dal programma
- o “*Stop Motion*”: Blocca il moto, se attivo, su tutti gli assi
- o “*Enable/Disabile Motors*”: Abilita/Disabilita i drivers dell'ESP6000

Sono presenti inoltre due arrays di bit di controllo e l'array “*Last Position*”.

I due grafici “*Beam*” e “*Target*” ricoprono un’importanza fondamentale nel complesso del programma. *Target* è un grafico XY che prende i valori corrispondenti ai driver 1 e 2, che pilotano i due attuatori lineari per gli spostamenti sul piano del rivelatore, perpendicolare alla linea di fascio Z. *Beam* invece rappresenta la movimentazione del collimatore posto di fronte al rivelatore (che effettua traslazioni lungo l’asse X), rispetto alla linea di fascio Z.

3.3.1 Strutture dati implementate

Per il programma sviluppato sono stati creati due grossi arrays: “*Motor Array*” e “*Axis Array*”.

Motor Array è un array monodimensionale composto da 4 clusters, i quali contengono i seguenti dati:

Tabella 3-5: Dati contenuti nei Clusters di Motor Array

	Nome	Tipo Dato
Cluster	<i>MotorName</i>	String
	<i>Drive</i>	Integer
	<i>Target</i>	Integer
	<i>SetAxisZero</i>	Boolean
	<i>Go</i>	Boolean
	<i>Enabled</i>	Boolean
	<i>Home</i>	Boolean
	<i>First Time</i>	Boolean
	<i>Reset Target</i>	Boolean

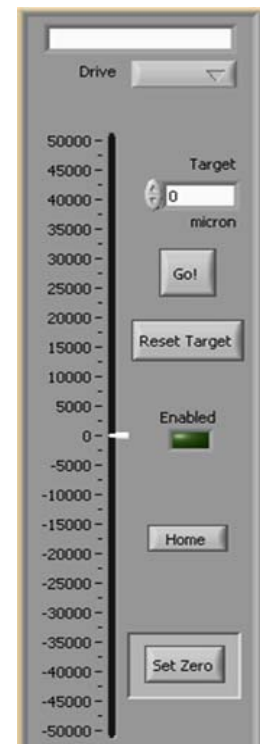


Figura 3.23: Visualizzazione del cluster

- o “*Motor Name*”: è il nome che viene assegnato al driver che andrà poi a comandare un attuatore
- o “*Drive*”: L’ESP6000 DCIB ha quattro drivers distinti, questo parametro indica quale driver viene scelto
- o “*Target*”: il valore numerico di questo campo viene inviato come spostamento desiderato in micron
- o “*Go*”: condizione booleana legata al pulsante che avvia la movimentazione
- o “*Reset Target*”: pulsante che aggiorna gli arrays utilizzati dai grafici
- o “*Enabled*”: la condizione se il driver è abilitato o no è visualizzata da un led
- o “*Home*”: questo pulsante comanda agli attuatori di tornare al punto preso come riferimento
- o “*Set Zero*”: questo pulsante imposta la posizione corrente come zero dell’asse di riferimento
- o “*First Time*”: questo dato non è visualizzato, indica se il driver del cluster è cambiato o no dall’ultima esecuzione

Axis Array è invece un array monodimensionale composto da 4 clusters di clusters, che a loro volta contengono i diversi dati da manipolare.

Tabella 3-6: Dati contenuti nei Clusters di Axis Array

		Nome	Tipo Dato
Main Cluster	Setup Parameters Cluster	<i>Motor type</i>	Integer
		<i>Max Vel</i>	Double
		<i>Max Accel</i>	Double
	Motion Parameters Cluster	<i>Vel</i>	Double
		<i>Accel</i>	Double
		<i>Decel</i>	Double
		<i>Jerk</i>	Double
	PID Parameters Cluster	<i>Kp</i>	Double
		<i>Ki</i>	Double
		<i>Kd</i>	Double
		<i>Af</i>	Double
		<i>Vf</i>	Double

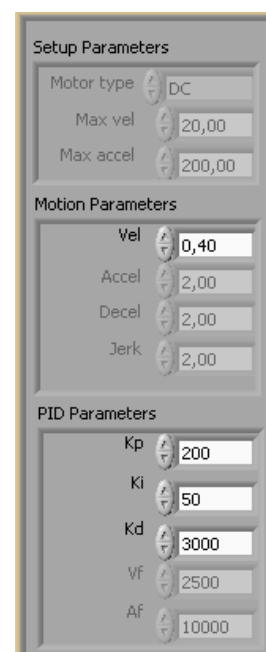


Figura 3.24: Visualizzazione del Cluster

Il cluster “*Setup Parameters*” contiene i seguenti dati:

- o “*Motor Type*”: un valore numerico che identifica le seguenti opzioni:

**Tabella 3-7: Tipo motorino e
valore numerico corrispondente**

0	No Motor
1	DC Motor
2	Stepper Motor

- o “*Max Vel*” e “*Max Accel*”: massima velocità e accelerazione selezionabile

Il cluster “*Motion Parameters*” contiene i seguenti dati:

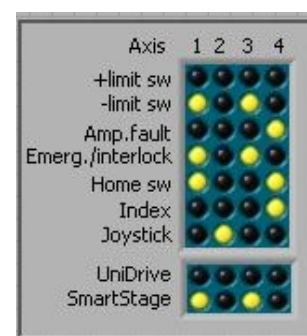
- o “*Vel*”, “*Accel*”, “*Decel*”, “*Jerk*”: i valori rappresentano rispettivamente Velocità, Accelerazione, Decelerazione e Spinta

Il cluster “*PID Parameters*” contiene i seguenti dati:

- o “*Kp*”, “*Ki*”, “*Kd*”: i valori numerici da inviare rispettivamente ai blocchi proporzionale, integrativo, derivativo del controllo retroazionato
- o “*Af*” e “*Vf*”: Acceleration forward e Velocity forward, altri valori impostabili per il PID

3.3.2 Arrays di bit

Sono due arrays che mostrano lo stato degli attuatori in tempo reale, ricoprendo un’importanza fondamentale quando non è possibile avere un riscontro visivo immediato della situazione corrente. Si sono rivelati molto utili nel mostrare quando gli attuatori avevano raggiunto il limite di fine corsa positivo o negativo.



**Figura 3.25: Visualizzazione
degli Arrays di bit**

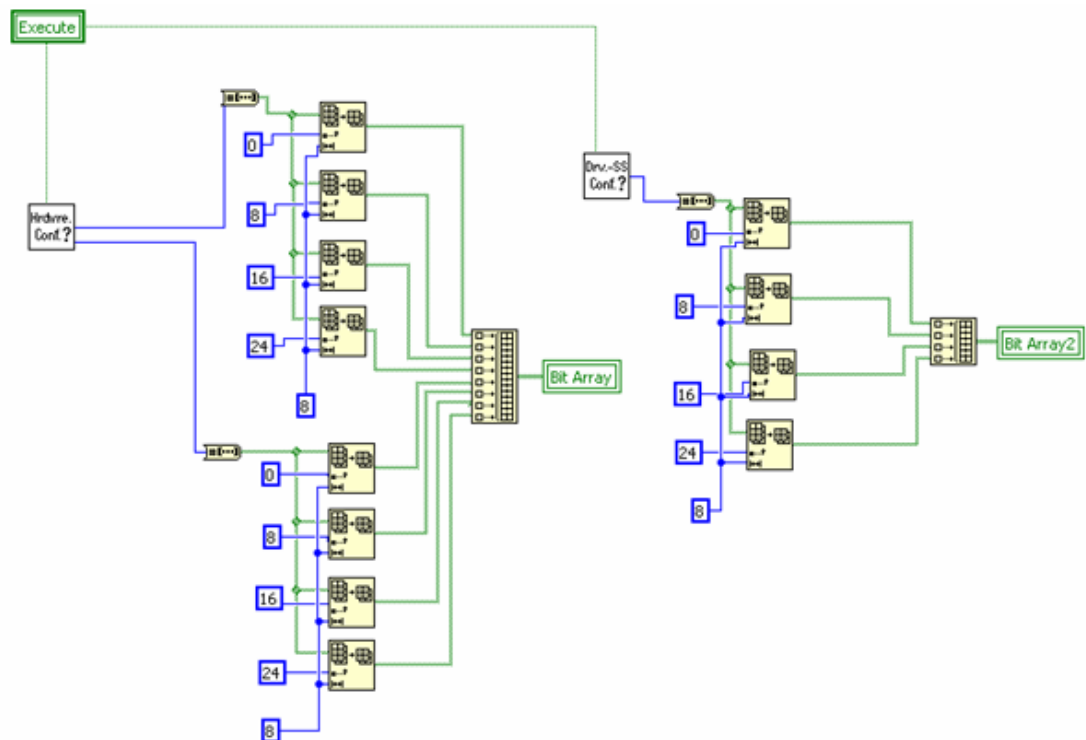


Figura 3.26: A sinistra Il subVI “Hardware Status” fornisce in uscita due valori di tipo integer a 32 bit che vengono successivamente convertiti in arrays di 32 valori booleani; questi valori, presi 8 bit per volta tramite la funzione “Array Subset”, andranno a essere concatenati con la funzione “Build Array” per formare “Bit Array”. A destra lo stesso procedimento: i dati che provengono dal subVI “System Configuration” andranno a costruire “Bit Array 2”.

3.3.3 Arrays numerici

Sono stati creati quattro arrays che contengono i dati numerici da cui verranno visualizzati la posizione attuale, quella relativa e quella del motorino rispetto all’encoder.

- o “Encoder Position”: viene costantemente aggiornato dal driver prendendo in ingresso i dati che provengono direttamente dagli encoders degli attuatori
- o “Zero Position”: registra il valore preso come riferimento per gli assi di movimentazione
- o “Actual Position”: mostra in tempo reale la posizione relativa sommando i valori degli arrays precedenti



Figura 3.27: Visualizzazione degli Arrays

- o *“Last Position”*: conserva le coordinate dell’ultimo punto raggiunto.

3.3.4 File di inizializzazione

In fase di avvio del programma viene chiesto il percorso da cui cercare il file di testo con i dati di inizializzazione, tale file contiene i valori che verranno passati ai clusters di *Motor Array* per identificare i diversi attuatori. Di seguito viene riportato un esempio di file di inizializzazione con il commento per ogni sezione.

Tabella 3-8: Il file di testo di inizializzazione

[Section 0] Motor Number 1=1 Motor Number 2=2 Motor Number 3=3 Motor Number 4=4	<i>Motor Number</i> è il valore che viene impostato nel campo <i>Drive</i> del cluster, è il valore che identifica il numero del driver dell’ESP6000
[Section 1] Motor Name 1="X (Left-Right) " Motor Name 2="Y (Up-Down) " Motor Name 3=Collimator Motor Name 4=None	<i>Motor Name</i> è la stringa che viene inserita nel corrispondente campo del cluster e viene visualizzata nella casella di testo corrispondente; serve a dare un nome ad ogni driver
[Section 2] Encoder Position 1=-12,1089 Encoder Position 2=-32,6986 Encoder Position 3=0,0000 Encoder Position 4=0,0000	Il valore di <i>Encoder Position</i> viene salvato nell’array <i>Encoder Position</i> , l’array che viene costantemente aggiornato dai valori di ritorno degli encoders degli attuatori
[Section 3] Zero Position 1=37,8911 Zero Position 2=17,3014 Zero Position 3=31,6945 Zero Position 4=50,0000	Il valore <i>Zero Position</i> viene salvato nell’array <i>Zero Position</i> , questo array contiene i valori numerici che permettono di visualizzare lo zero degli assi di riferimento.

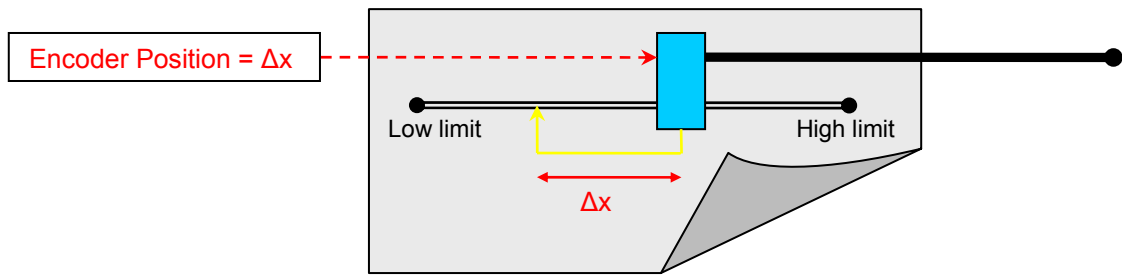


Figura 3.28: Quando l'encoder dell'attuatore riceve il valore numerico, letto dal file di inizializzazione, lo imposta come valore attuale; in questo modo viene determinato il punto che corrisponde allo 0 dell'asse

3.4 Descrizione del programma

Lo schema a blocchi del programma è contenuto all'interno di una Case Structure, alla quale si accede premendo il pulsante *“Execute”*.

Dentro vi è contenuta una Stacked Sequence Structure di quattro pagine, che scandisce le quattro fasi principali del programma:

- o Pagina 0: Inizializzazione del controllore, variabili e arrays
- o Pagina 1: Lettura del file d'inizializzazione e impostazione della velocità per gli attuatori
- o Pagina 2: Parte centrale del programma, contiene tutte le funzioni e le procedure per la movimentazione controllata e la gestione dei grafici
- o Pagina 3: Scrittura del file d'uscita (da utilizzare come inizializzazione al successivo rientro)

0.1.1 Pagina 0

Viene avviato il subVI *“Initialize”* che si occupa di aprire la comunicazione con il modulo ESP del PC e il modulo esterno DCIB.

Successivamente viene mostrata una casella di dialogo che mostra la versione del firmware del controllore, dove *Axis Array* viene scomposto nei suoi quattro elementi che

contengono i cluster e poi, tramite una scomposizione *unbundle*, vengono presi i valori di default e inviati alla procedura che si occupa di inizializzare i parametri per ogni driver.

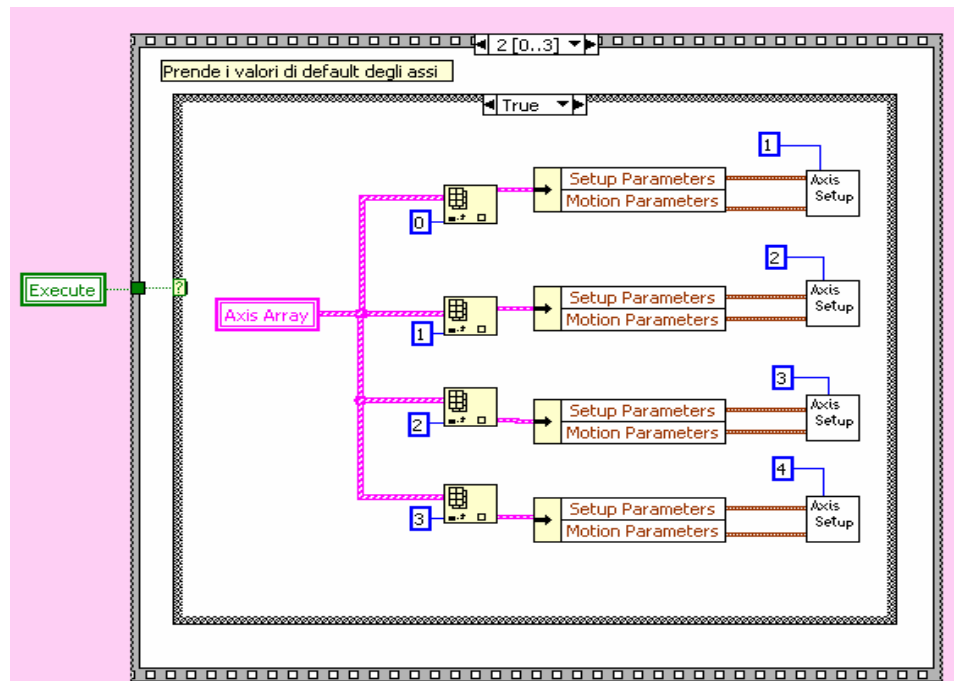


Figura 3.29: Axis Array viene prima scomposto nei quattro elementi che lo compongono tramite la funzione “Indexed Array”, poi tramite una funzione “Unbundle” vengono presi ed estratti i clusters Setup Parameters e Motion Parameters che contengono i dati in ingresso per i subVI “Axis Setup”

In realtà questi valori verranno aggiornati costantemente seguendo i valori impostati nel tab Menu “*Axis Parameters*”, come spiegato in precedenza.

Gli attuatori utilizzati permettono infatti di regolare in modo fine i parametri da inviare al controllore PID in modo da massimizzare le prestazioni delle traiettorie e quindi del moto; tali aggiustamenti risultano necessari per compensare effetti di degenerazione dovuti a pressione, umidità e carico a cui sono sottoposti gli attuatori.

Come conclusione di questa Stacked Sequence si ha la configurazione del limite hardware tramite il subVI “*Hard-limit configuration*”.

Vengono inoltre creati gli arrays *Actual*, *Zero*, *Encoder*, *Last Position* e inizializzati a zero per ogni elemento.

3.4.1 Pagina 1

Viene effettuato il parsing (analisi) del file di inizializzazione tramite le funzioni di comparazioni stringhe, passando come valori chiave “Section 0, 1, 2, 3”, e i valori vengono inseriti all’interno dei campi corrispondenti di **Motor Array** e degli arrays **Encoder**, **Last** e **Zero Position**.

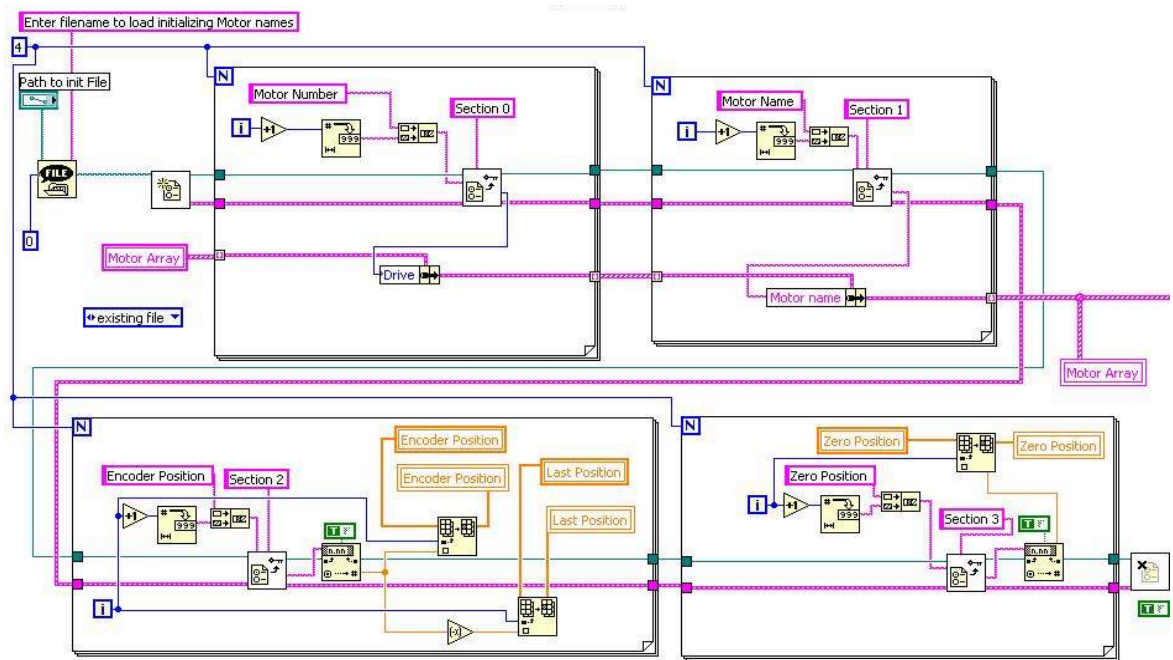


Figura 3.30: La lettura del file di inizializzazione avviene in varie fasi

3.4.2 Pagina 2

All’interno di questa pagina, che contiene il cuore del programma, è presente una Stacked Sequence Structure di 10 pagine.

- o Pagina 0: All’interno di una case sequence viene controllato se il numero del driver assegnato al cluster di **Motor Array** è cambiato rispetto all’ultima visita e, in caso positivo, viene aggiornato il valore relativo alla posizione dell’encoder dell’attuatore collegato. Un’altra case sequence si occupa di aggiornare i valori indirizzati al controllore PID.

- o Pagina 1: Vengono interrogati gli encoders degli attuatori, i valori di ritorno andranno a popolare l'array *Encoder Position*. Allo stesso tempo questi valori vengono negati nel segno e andranno a popolare l'array *Beam Position*, array fondamentale per la visualizzazione del grafico.
- o Pagina 2: Viene popolato l'array *Actual Position*, i valori in esso inseriti sono il risultato del valore di ritorno dell'encoder (*Encoder Position*) sottratto a quello imposto come riferimento dell'asse (*Zero Position*).
- o Pagina 3: In questa pagina viene controllato, all'interno di un ciclo for che permette di esplorare lo stato per ogni elemento di *Motor Array*, lo stato dei pulsanti "Go!", "Home" e "Reset Target". Nel caso in cui essi siano premuti vengono avviate le procedure corrispondenti: "Go!" avvia la procedura di movimento relativo, mandando come valore ciò che proviene dal campo *Target*; "Home" avvia la procedura di movimento assoluto, comandando all'attuatore di ritornare alla sua posizione di zero, e aggiorna al valore zero il campo corrispondente nell'array *Last Position*; "Reset Target" aggiorna il campo nell'array *Last Position* aggiungendovi il valore immesso nel campo *Target*, per poi porre il campo *Target* a zero.
- o Pagina 4: Vengono opportunamente combinati i valori, prelevati dagli array numerici, che andranno a essere visualizzati nei grafici; tale combinazione verrà spiegata in modo più esauriente in un successivo paragrafo.
- o Pagina 5: Il pulsante "Stop Motion", all'ingresso di un ciclo for per agire sui quattro drivers, si occupa di interrompere il moto su tutti gli assi, avviando il subVI "Stop Motion".
- o Pagina 6: Vengono popolati gli arrays di bit di controllo, andando a interrogare lo stato dell'hardware e la configurazione del sistema tramite i subVI preposti. I valori in uscita, di tipo integer, vengono convertiti in array di valori di tipo booleano, e poi rappresentati graficamente con delle tabelle a led.
- o Pagina 7: Quando viene premuto il pulsante "SetAxisZero" viene aggiornato il valore presente all'interno dell'array *Zero Position* secondo la seguente sequenza: al valore di *Zero Position* viene sottratto il valore di *Encoder Position*; viene imposto all'encoder dell'attuatore di considerare la posizione attuale come zero; viene azzerato il valore nel campo *Target*.
- o Pagina 8: Viene interrogato lo stato dei drivers attraverso la chiamata diretta a DLL `esp_get_motor_onoff_status`; il valore di ritorno va ad aggiornare il campo

enabled del *Motor Array* per ogni attuatore, mostrando così con un led luminoso lo stato attuale dei drivers.

- o Pagina 9: Stop 100 ms

3.4.3 Pagina 3

A questa pagina si accede solo dopo che è stato premuto il pulsante “*Program Stop*”, che permette l’uscita del ciclo while al cui interno era contenuta la Stacked Sequence Structure descritta nel paragrafo precedente.

Il file di testo in uscita viene assemblato nelle sue quattro sezioni andando a scrivere i valori che vengono letti dai campi “*Drive*”, “*Motor Name*”, “*Encoder Position*” e “*Zero Position*”. Tale file permetterà, al successivo riavvio del programma, di non perdere i valori attuali di posizione degli attuatori, permette quindi di memorizzare la situazione globale per tutto il sistema.

Il pulsante “*Execute*” viene impostato al valore *False*, così da inibire ogni utilizzo dei subVI presenti nel programma.

3.5 Perché un grafico?

La scelta progettuale di realizzare un grafico nasce dall’evidente vantaggio che si acquisisce nell’avere un sistema di monitoraggio in tempo reale, semplice ed intuitivo, che permette di osservare la struttura che viene sottoposta a movimentazione dalla consolle.

Se il sistema di piastre e attuatori fosse visibile all’operatore, questa funzionalità non ricoprirebbe un requisito importante, ma avendo a che fare con un sistema che verrà posto all’interno di una camera a vuoto, lontana dall’operatore e per di più in condizioni di radioattività, il supporto grafico permette di controllare sia le fasi di movimento, sia la posizione del campione. Inoltre risulta necessario, dato che si ha a che fare con spostamenti nell’ordine delle centinaia di micron, un punto di osservazione che ingrandisca la visuale sugli spostamenti in modo da permettere di procedere con estrema accuratezza.

Volendo simulare il movimento del fascio di particelle sulla superficie del campione, con un grafico risulta più semplice adottare un sistema di riferimento solidale con l'oggetto stesso. In questo modo si vedrà muovere un cursore (che rappresenta il fascio) all'interno dell'area del grafico.

Come dimensioni dell'area visibile si è pensato di utilizzare il valore corrispondente all'escursione massima degli attuatori, 50 mm; in questo modo si può facilmente stimare, considerando la distanza del cursore dai bordi del grafico, quanto si è vicini ai limiti di corsa inferiore o superiore degli attuatori. Un reticolo di riferimento, con intervallo di divisione pari a 5 mm, è stato inserito come supporto visivo.

L'utilizzo di un sistema di riferimento solidale con l'oggetto da sottoporre a movimentazione impone dei cambiamenti di segno per tutti i valori numerici coinvolti nelle procedure di movimentazione. L'array **Beam Position** contiene i valori, letti dagli encoders degli attuatori, già contenuti in **Encoder Position** ma cambiati di segno; in quanto uno spostamento del cursore sull'asse X verso destra (positivo) corrisponde ad uno spostamento verso sinistra (negativo) dell'attuatore e quindi della piastra.

Il valore numerico contenuto nel campo *Target*, inviato in seguito alla commutazione del pulsante “Go!” per avviare la procedura di movimento relativo, viene cambiato pertanto di segno.

Il valore desiderato per lo spostamento viene visualizzato in tempo reale dall'omonimo cursore “*Target*” (verde) che, posto sotto il cursore “*Beam*” (rosso), si muove della quantità desiderata. In questo modo l'immissione dati risulta del tutto trasparente all'utente che non deve preoccuparsi di comandare uno spostamento con segno positivo o negativo: quando si muove lo slider di una quantità positiva nella scala graduata, si avrà uno spostamento positivo del cursore (che corrisponderà ad uno spostamento in senso contrario delle piastre).

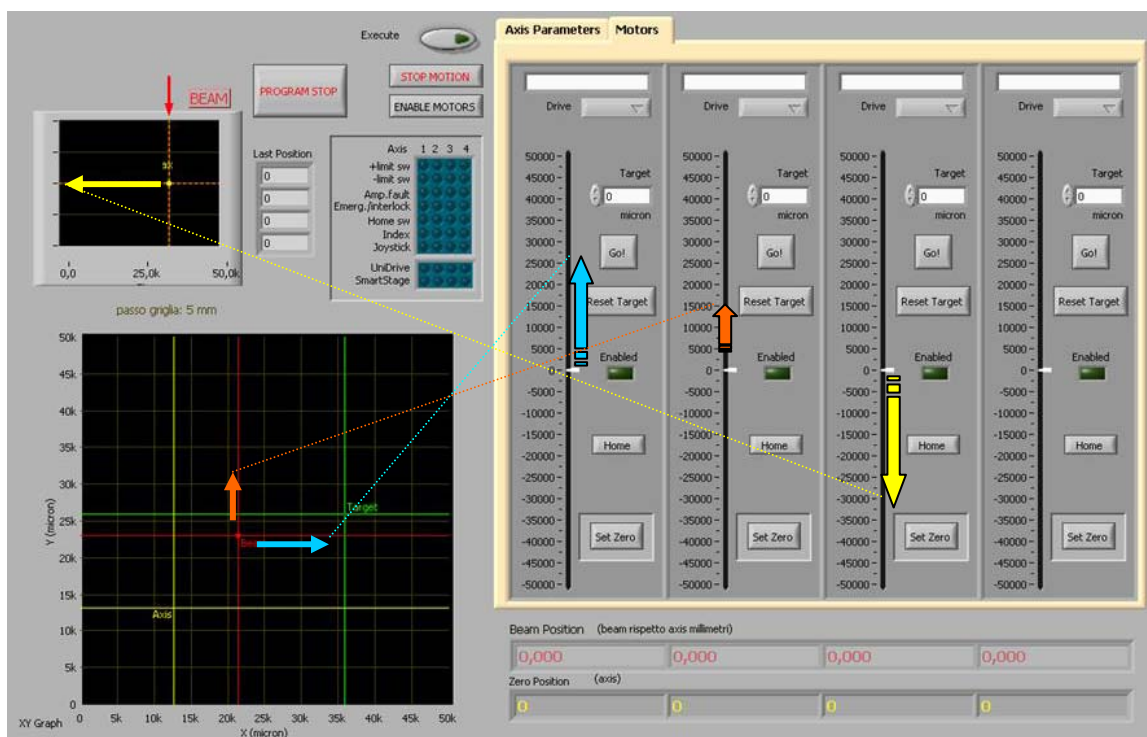


Figura 3.31: Le frecce indicano che moto produrrà uno spostamento dello slider corrispondente

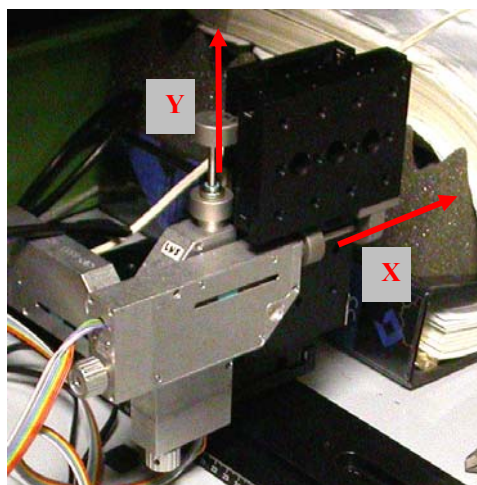


Figura 3.32: Attuatori montati con orientamento negativo del moto. Ad uno spostamento positivo delle piastre corrisponde uno spostamento negativo del rivelatore rispetto il fascio di particelle

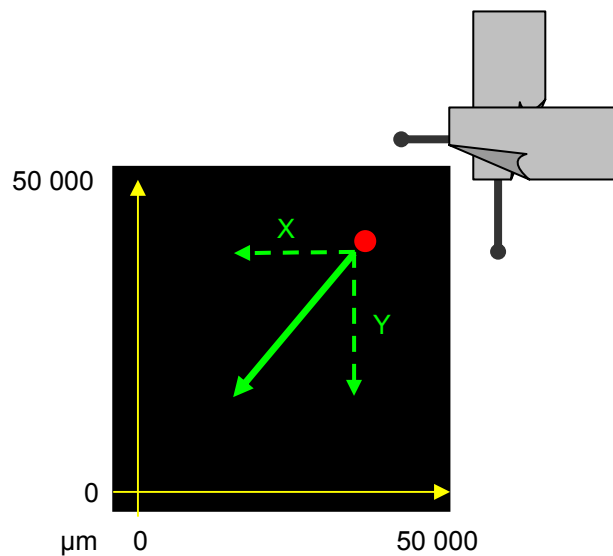


Figura 3.33: Utilizzando questo sistema di riferimento relativo, il punto di coordinate (50 000; 50 000) sul grafico corrisponde al limite inferiore di corsa degli attuatori, mentre l'origine degli assi (0; 0) è il punto corrispondente alla massima escursione raggiungibile

3.6 Coordinate e Spostamenti Relativi

Quando viene premuto il pulsante “Go!” si assiste al movimento, visibile grazie ai valori di ritorno degli encoders, del cursore *Beam* che si andrà a riposizionare sopra *Target*.

Ogni punto desiderato si raggiunge quindi muovendosi per steps (utilizzando il subVI “Move to relative position” che si occupa di questa funzione).

Mentre il cursore si muove, le coordinate visualizzate nell’array *Beam Position* sono in relazione al punto scelto come origine del sistema di riferimento *Axis* (cursore giallo); l’array *Zero Position* invece mostra la coordinata dell’origine degli assi XY rispetto l’area visibile del grafico.

E’ possibile scegliere come riferimento per gli assi qualsiasi punto all’interno della corsa degli attuatori, è sufficiente premere il pulsante “Set Zero” per aggiornare l’array *Zero Position* e di conseguenza il grafico.

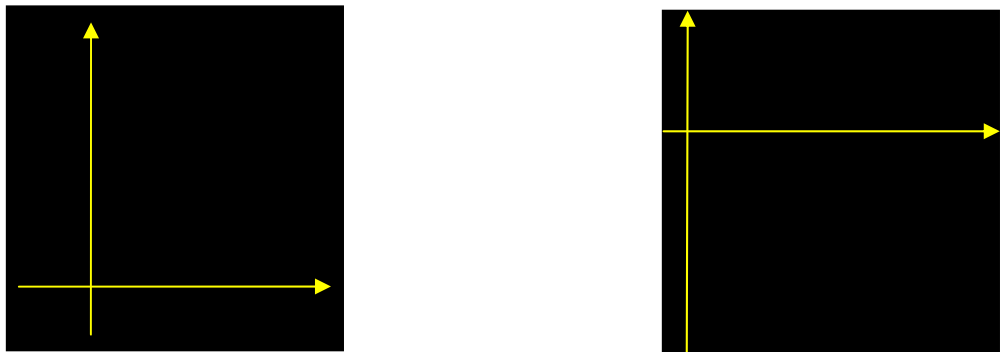


Figura 3.34: Il sistema di riferimento degli assi, indicato con le linee gialle nel grafico, può essere posizionato in qualunque posto nell’area visibile (che rappresenta il range di movimento dei due attuatori)

3.7 Arrays statici e dinamici per la visualizzazione dei puntatori

Tabella 3-9: Arrays dati e tipo di dati contenuti

<i>Zero Position</i>	Contiene la coordinata dello 0 di riferimento per ogni asse, nel range di corsa degli attuatori. Su questo valore viene ancorato il cursore giallo Axis	<i>statico</i>
----------------------	---	----------------

Actual Position	Legge direttamente dagli encoders i valori attuali durante gli spostamenti	<i>dinamico</i>
Target Position	Prende il valore impostato immesso con il controllo slider e lo rende disponibile per eseguire lo spostamento desiderato	<i>statico</i>
Last position	Registra l'ultima coordinata raggiunta	<i>statico</i>

3.8 I pulsanti di Motor Array

Tabella 3-10: Pulsanti di Motor Array e azione eseguita

Axis Zero	Aggiorna l'array <i>Zero Position</i> , impostando la posizione attuale come nuovo zero di riferimento per l'asse
Go!	Passa il valore contenuto nel campo <i>Target</i> alla procedura che si occupa degli spostamenti relativi
Reset Target	Aggiorna l'array <i>Last Position</i> , utilizzato dai cursori sul grafico, e imposta nuovamente 0 come valore iniziale nel campo <i>Target</i>
Home	Muove il fascio verso il punto scelto come riferimento per l'asse; avvia la procedura di spostamento verso una coordinata assoluta, mandando come valore 0

3.9 I tre cursori: Axis, Beam e Target

I valori degli arrays *Actual*, *Zero* e *Last position*, prelevati tenuto conto dell'indice che ne identifica il numero dell'asse, vengono inviati sui canali X e Y del grafico, solo dopo però essere stati raggruppati in un array di passaggio per la conversione in dati dinamici. Il flusso di questi dati dinamici, in ingresso al grafico XY, viene opportunamente processato dal VI Express "Build XY Graph".

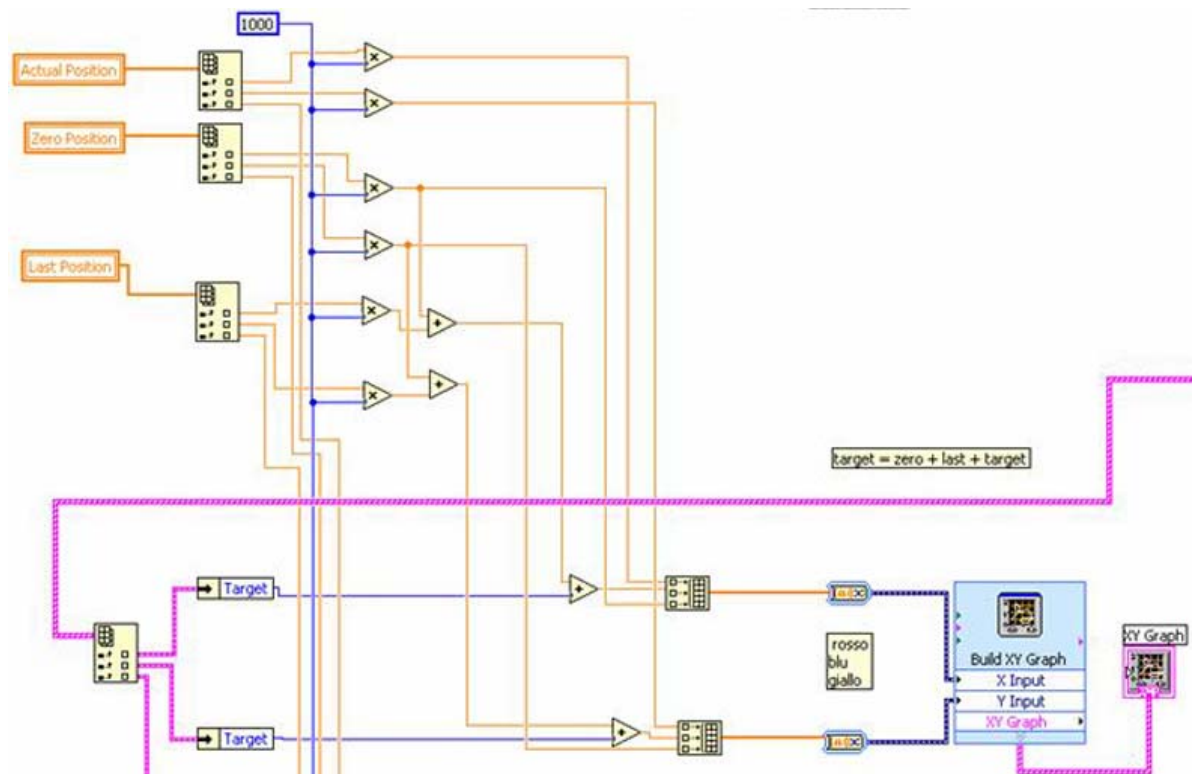


Figura 3.35: I valori mostrati dai cursori del grafico sono prelevati direttamente dagli arrays e combinati in modo opportuno

Nota: Gli encoders degli attuatori utilizzano il millimetro come unità di misura, in tutto il programma vengono utilizzati valori in micron. E' necessario quindi, ogni volta che i valori vengono prelevati dagli arrays, effettuare una conversione di misura (moltiplicare per 1000 il valore numerico)

I cursori del grafico quindi rappresenteranno valori diversi:

Tabella 3-11: Colore e dati rappresentati dai cursori

Beam	<i>Actual Position</i>	Rosso
Target	<i>Zero Position + Last Position + Target</i>	Verde
Axis	<i>Zero Position</i>	Giallo

Agendo sulla casella proprietà del grafico, sul Front Panel, è stato possibile quindi aggiungere i tre cursori “Axis”, “Beam” e “Target”, che prendono i valori rispettivamente dal primo, secondo e terzo elemento degli array, costruiti per contenere tali dati dinamici.

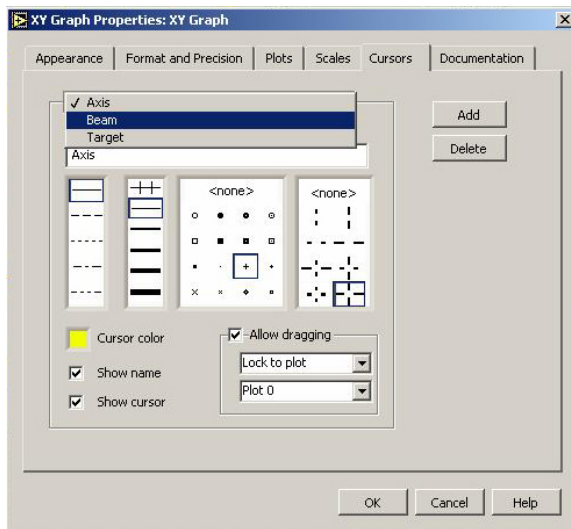


Figura 3.36: Dalle proprietà del grafico è possibile aggiungere i cursori da visualizzare

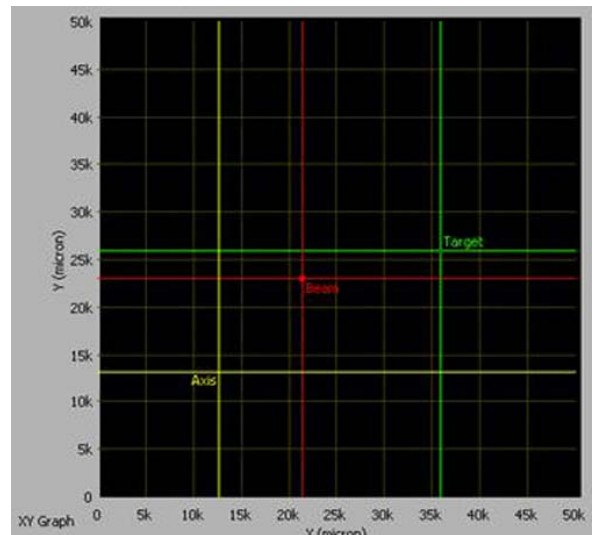


Figura 3.37: Risultato ottenuto, tre cursori indipendenti

3.10 Schema del rivelatore da inserire sotto il grafico

Nel Front Panel di LabVIEW è possibile inserire, oltre a comandi e oggetti per la manipolazione dati, immagini prelevate da files esterni (di tipo .jpg, .bmp, ecc); si è quindi pensato di inserire all'interno del grafico un'immagine che riproducesse la struttura del rivelatore da sottoporre a caratterizzazione, in modo da avere in tempo reale l'indicazione visiva del punto che si sta irraggiando.

Tuttavia, data la necessità di avere un grafico ben definito, non è opportuno inserire files a mappa di bit (raster), perché nel ridimensionamento perderebbero definizione. Convienne, piuttosto, inserire un'immagine di tipo vettoriale (dove le linee e le curve sono rappresentate da funzioni matematiche), in modo da mantenere la definizione quando viene modificato nelle dimensioni.

L'immagine del campione è stata quindi disegnata con un programma CAD con le giuste dimensioni e convertita in file .wmf (Windows MetaFile), supportato da LabVIEW.

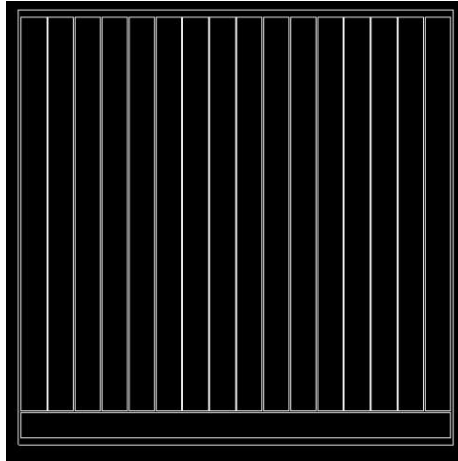


Figura 3.38: Rappresentazione grafica del rivelatore

Per inserire il disegno di riferimento come sfondo del grafico è stato necessario agire nuovamente sulle sue proprietà.

Nonostante sia possibile cambiare il colore dello sfondo (il nero è il colore di default), non è permesso inserire un'immagine da file esterno come fondo stesso. Si è proceduto quindi rendendo trasparente lo sfondo del grafico, si è creato un quadrato nero di area pari allo sfondo (non più visibile) e l'immagine del campione è stata inserita in mezzo (utilizzando le funzioni di ordinamento oggetti presenti sul Front Panel), creando così un grafico a livelli sovrapposti.

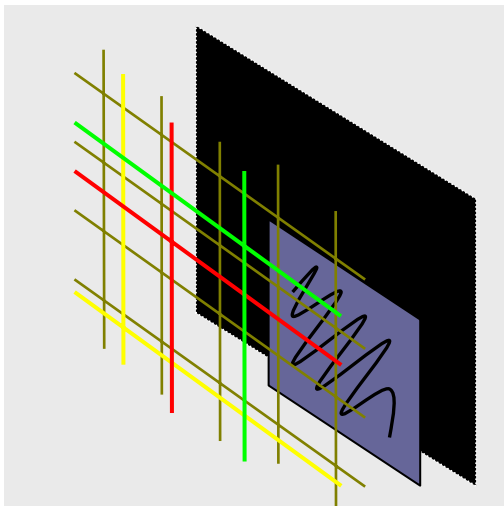


Figura 3.39: Rappresentazione grafica della struttura a livelli sovrapposti

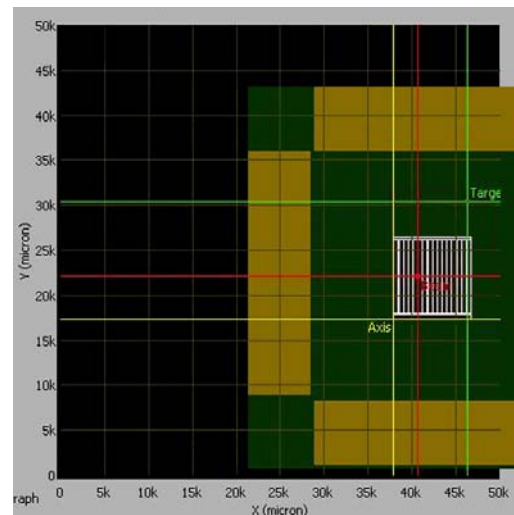


Figura 3.40: Per il risultato finale è stato preso pure in considerazione l'ingombro del PCB

3.11 Zoom

Per molti oggetti che vengono inseriti nel Block Diagram è possibile agire sui “*Property Node*”, che permettono il settaggio di alcuni parametri direttamente da comandi posti sul Front Panel.

Il property node del tipo di grafico XY mette a disposizione ad esempio i campi “*Scale.Minimum*” e “*Scale.Maximum*” per i due assi visualizzati, che permettono di agire in modo dinamico sui valori numerici che verranno rappresentati sulle scale ascissa e ordinata.

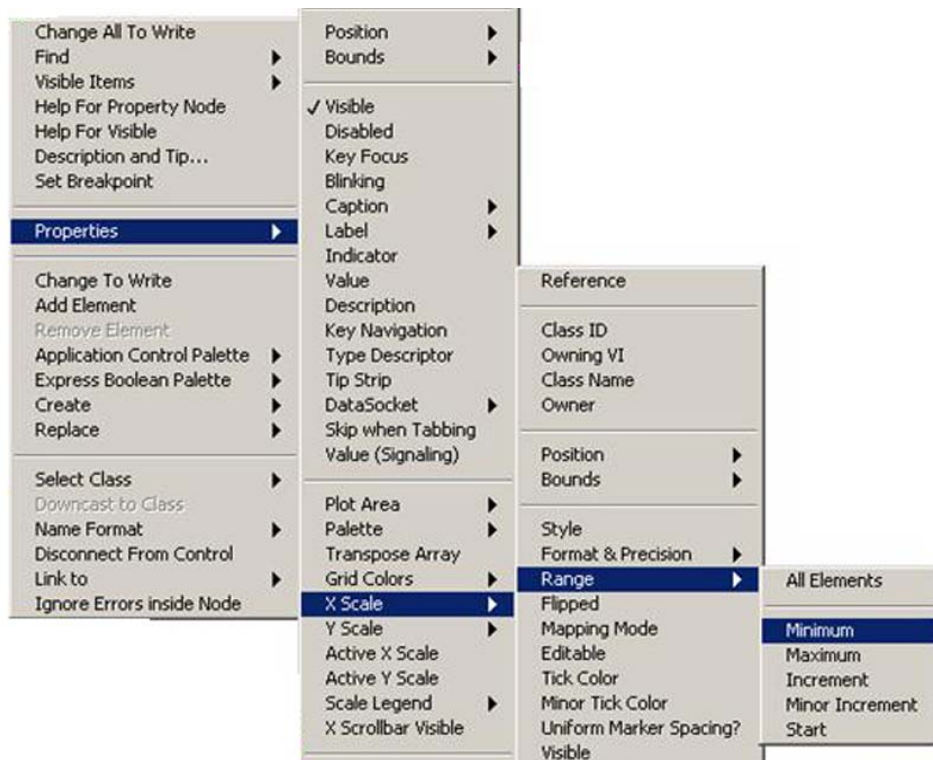


Figura 3.41: Alcuni dei possibili controlli che è possibile raggiungere dal Property Node di un grafico XY, in figura “XScale.Minimum”

Sarebbe possibile quindi creare sul front panel uno slider per variare in modo dinamico la dimensione dell’area visualizzata all’interno del grafico, insomma uno zoom.

Ciò renderebbe al sistema un vantaggio significativo in quanto sarebbe possibile visualizzare da un punto d’osservazione lontano i movimenti macroscopici per spostamenti consistenti (nell’ordine dei millimetri o dei centimetri), mentre si potrebbe rimpicciolire l’area di grafico visualizzata (dilatando quindi i valori sugli assi) nei momenti in cui sarebbe necessaria una risoluzione maggiore (per spostamenti di frazioni di millimetro).

Si è pensato quindi di implementare una soluzione creando un altro programma, del tutto identico a quello originale, in cui però il Property node del grafico viene manipolato in modo da visualizzare un'area di 12x12 mm² (invece di 50x50 mm²), dato che la dimensione del campione è inferiore a 10x10 mm².

Dalla proporzione $\frac{zoom_{fact}}{100} = \frac{50}{12}$ segue che $zoom_{fact} = 100 \cdot \frac{50}{12} = 416$, quindi si è raggiunto un fattore di zoom pari a 416 % rispetto la visualizzazione precedente.

Il punto da cui partire nella visualizzazione sugli assi X e Y non può però essere preso a caso; si è scelto di utilizzare come punto di ancoraggio il valore presente nell'array *Zero Position*, dato che le coordinate dei punti sono espresse prendendo come punto di riferimento il cursore giallo nel grafico. Avendo a disposizione 12 mm e considerando che nell'immagine il lato del campione misura meno di 10 mm, verrà quindi visualizzata la porzione di grafico compresa tra 1 mm prima e 11 mm dopo gli assi di riferimento, la cui intersezione verrà posta nel punto corrispondente all'angolo in basso a sinistra dell'immagine del rivelatore.

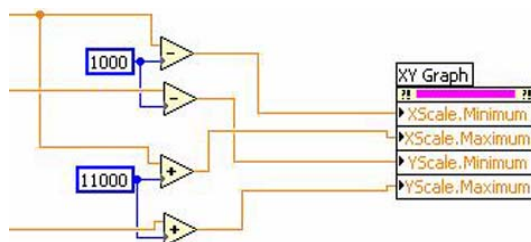


Figura 3.42: Al valore di ogni asse della coordinata di riferimento, prelevato dall'array *Zero Position*, vengono sottratti e sommati i valori scelti come limite inferiore e superiore di visualizzazione (da 1 mm prima a 11 mm dopo)

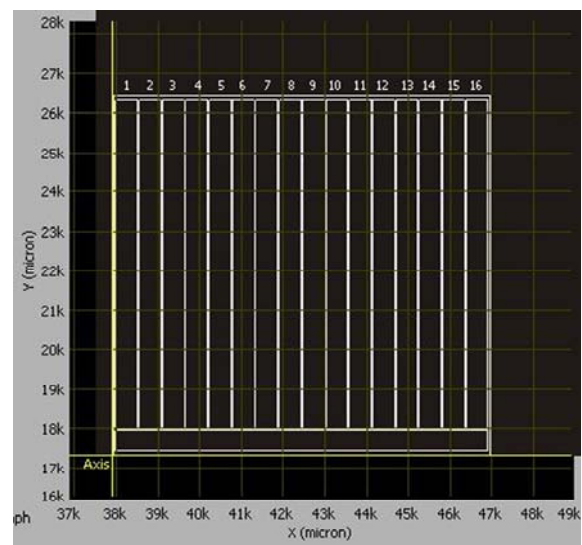


Figura 3.43: Risultato finale

Ogni volta che verrà premuto il pulsante “Set Zero” sull'asse X o Y verrà aggiornato l'array *Zero Position* e si noterà il cambiamento dei valori di coordinata nell'asse graduato corrispondente.

Per rendere questo secondo programma più adatto alla movimentazione micrometrica, e alle dimensioni dell'area visibile, è stato inoltre modificato il valore massimo di escursione comandabile tramite gli sliders del **Motor Array**, riducendolo da 50000 μm (50 mm) a 10000 μm (10 mm). Il passo di divisione del grafico nella versione zoom è stato ridotto da 5 mm a 1 mm.

Sarebbe risultato interessante creare anche uno zoom dinamico, che ridimensionasse l'area visibile in modo interattivo muovendo un cursore, come mostrato in figura. Si sarebbe potuto ad esempio "ancorare" il grafico al cursore **"Axis"** (che rappresenta gli assi), come mostrato in figura, oppure seguire il cursore **"Beam"** nel suo moto verso il cursore **"Target"**.

Allo stesso modo sarebbe stato possibile variare dinamicamente il passo della griglia di riferimento e mostrare tale valore all'utente, sempre agendo sui property node (basandosi sul valore di zoom impostato).

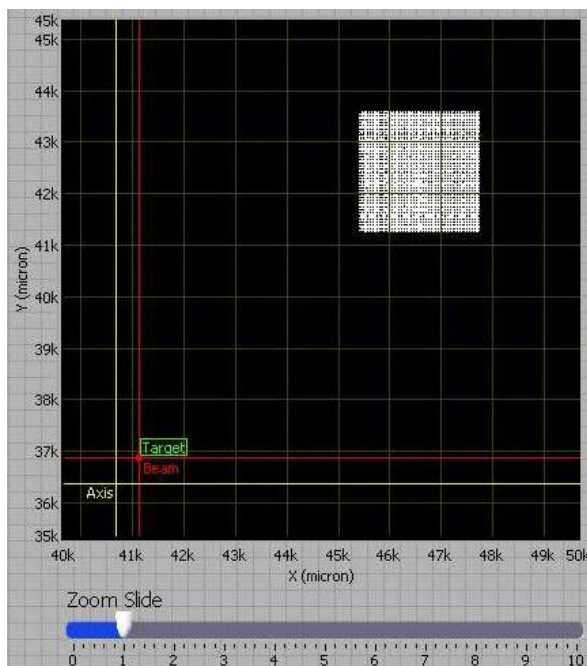


Figura 3.44: Slider Zoom

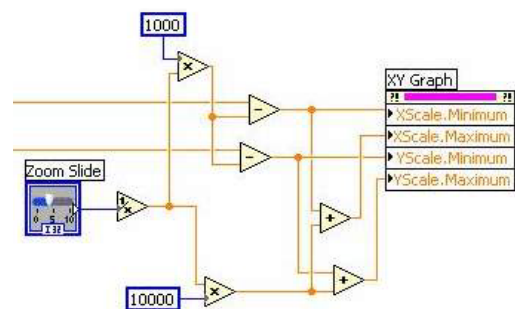


Figura 3.45: Moltiplicando il reciproco del fattore di zoom per i valori fissi impostati si ridimensiona la quantità di area visibile che, quindi, diminuisce per valori maggiori di 1 ma a causa di ciò appare più grande.

In questo caso il grafico è ancorato ai valori di **Axis Array**, il cursore giallo rimane immobile durante i ridimensionamenti.

Gli oggetti immagine che vengono inseriti nel Front Panel sono però sprovvisti di Property Node, le uniche operazioni possibili su di essi sono il ridimensionamento manuale e lo spostamento, risulta quindi impossibile implementare questa soluzione.

3.12 Grafico del collimatore

Il grafico che mostra la posizione attuale del microcollimatore finale è molto più semplice nella realizzazione, si tratta fondamentalmente di una finestra su cui viene rappresentata la posizione attuale del collimatore tramite un cursore a forma di croce, che si muove seguendo le indicazioni impostate dal cluster “Motor Array” relativo all’attuatore corrispondente.

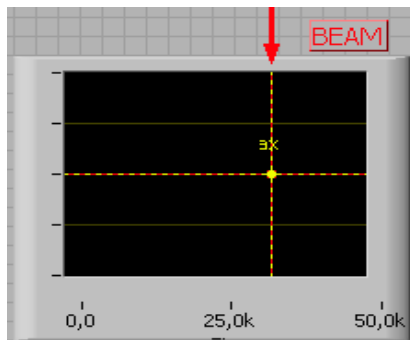


Figura 3.46: Collimatore allineato alla linea di fascio

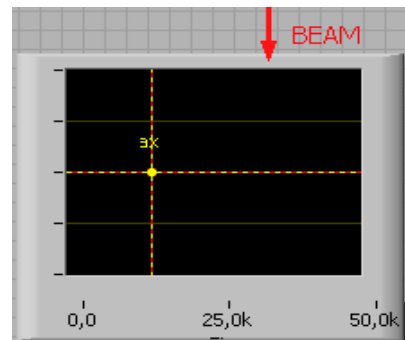


Figura 3.47: Collimatore fuori dalla linea di fascio

4 Allineamento

In questo capitolo viene fatta una panoramica sull'importanza della precisione del sistema messo in opera, evidenziando il significato dei termini ripetibilità e risoluzione di un sistema. Vengono proposti i risultati di misure statistiche effettuate per mettere alla prova la risoluzione del sistema in uso.

Viene inoltre riportato, in modo minuzioso, il metodo di allineamento prima ottico poi strumentale utilizzato per mettere in linea con il fascio di particelle la struttura modulare utilizzata.

4.1 Errori, Risoluzione del Sistema, Misura di tali entità

Per ripetibilità degli attuatori (o del sistema di movimentazione, ossia motore + piastra) si intende *“Attitudine di uno strumento a fornire valori di lettura poco differenti tra loro, in letture consecutive eseguite indipendentemente sullo stesso misurando con procedimento unificato, dallo stesso operatore nelle stesse condizioni per le grandezze d’influenza”*. [23]



Figura 4.1: Accuratezza e ripetibilità

La ripetibilità dichiarata per gli attuatori che ho adoperato è nell'ordine dei micron; se si volesse pertanto verificare l'esattezza del dato sarebbe necessario un sistema di misura con accuratezza nanometrica (i dati in questione sono il risultato di un'indagine statistica condotta con interferometri laser ad alta precisione).

Non avendo a disposizione uno strumento così preciso mi sono limitato a prendere confidenza con il sistema composto da piastre e attuatori e ho dato il via ad una serie di misure con lo strumento di precisione a disposizione: un calibro a cursore con risoluzione 50 μm .

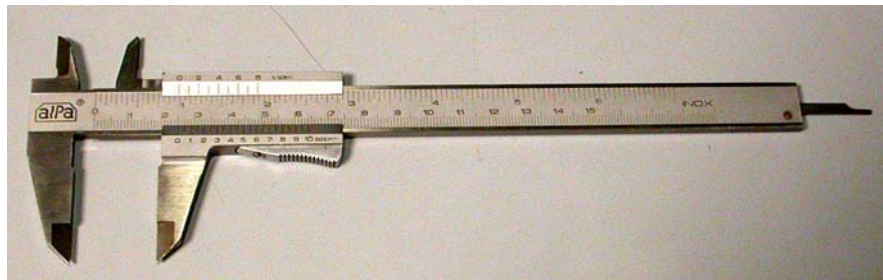


Figura 4.2: Calibro a cursore

La bassa risoluzione dello strumento, a confronto delle grandezze da misurare, ha imposto una collezione di dati statistici in modo da poter migliorare considerevolmente la risoluzione globale di questo sistema di misura diretta.

Nella maggioranza dei casi, la migliore stima dei valori di aspettazione di una grandezza q che varia casualmente, è la *media aritmetica* o *valore medio* $\tilde{q}$ delle n osservazioni:

$$\tilde{q} = \frac{1}{n} \sum_{k=1}^n q_k$$

dove q_k sono le n osservazioni indipendenti.

Le singole osservazioni q_k differiscono a causa di variazioni casuali delle grandezze d'influenza. La *varianza sperimentale* delle osservazioni è data da:

$$s^2(q_k) = \frac{1}{n-1} \sum_{k=1}^n (q_k - \tilde{q})^2$$

Questa stima della varianza e la sua radice quadrata positiva $s(q_k)$, denominata *scarto tipo sperimentale*, caratterizzano la variabilità dei valori osservati q_k o la dispersione intorno alla media $\tilde{q}$. [24]

Sono state effettuate misure statistiche per spostamenti lineari macroscopici di 5 e 3 mm con gli attuatori tipo 850-G (standard) e 850-HS (high speed). I risultati ottenuti hanno permesso di apprezzare l'affidabilità del sistema per misure di tipo millimetriche e l'andamento statistico delle prestazioni dei diversi attuatori, messe a confronto.

Tabella 4-1: Risultati delle prestazioni messi a confronto

5 mm	850-G (Standard)	850-HS (High Speed)
<i>Valore medio di lettura</i>	4,993 mm	4,998 mm
<i>Varianza sperimentale</i>	428 μm^2	119 μm^2
Scarto tipo	20,68415 μm	10,928 μm
Errore %	0,413683	0,218556
3 mm	850-G (Standard)	850-HS (High Speed)
<i>Valore medio di lettura</i>	2,999 mm	2,998 mm
<i>Varianza sperimentale</i>	191 μm^2	110 μm^2
Scarto tipo	13,83221 μm	10,502 μm
Errore %	0,461074	0,350075

Un'analisi dei risultati ottenuti evidenzia un comportamento per gli attuatori serie 850-G meno accurato rispetto l'attuatore serie 850-HS per spostamenti dell'ordine dei millimetri, infatti nel range di misure analizzato risultano meno precisi; tuttavia la tendenza che si può osservare è di una riduzione graduale dell'errore di posizionamento al ridursi della distanza.

I dati raccolti permettono di fare inoltre alcune considerazioni sulla ripetibilità del sistema: avere ad esempio un errore percentuale di 0,5 % per spostamenti di 500 μm vuol dire lavorare con un sistema che introduce un errore di posizionamento pari a 2,5 μm ; i dati raccolti pur derivando da una serie di misure grossolane si avvicinano molto alle caratteristiche descritte dalla casa di produzione per movimenti micrometrici.

I seguenti grafici mostrano i risultati ottenuti eseguendo 100 misure per uno spostamento di 5 mm con un attuatore serie 850-G.

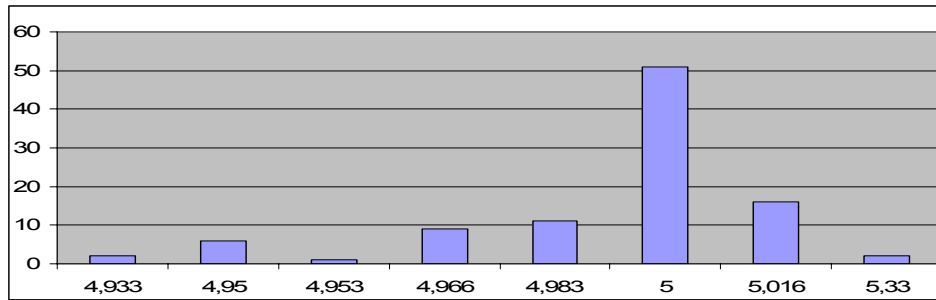


Figura 4.3: Letture statistiche del misurando

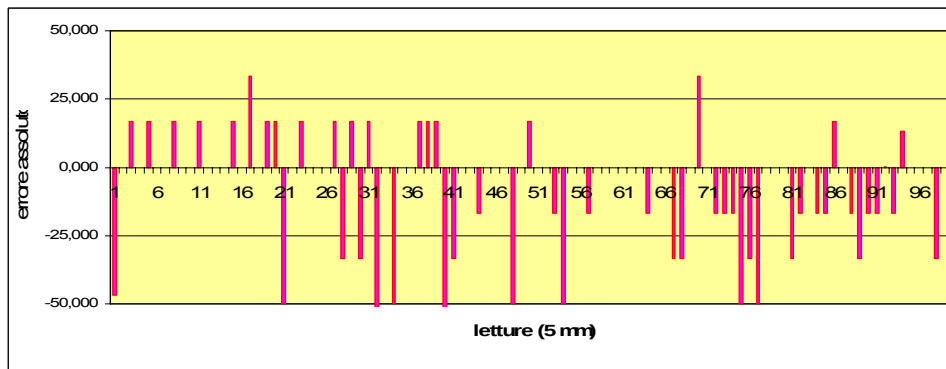


Figura 4.4: Errore assoluto dato dalla differenza tra il valore letto e il valore ideale

$$Err_{abs} = q_k - q$$

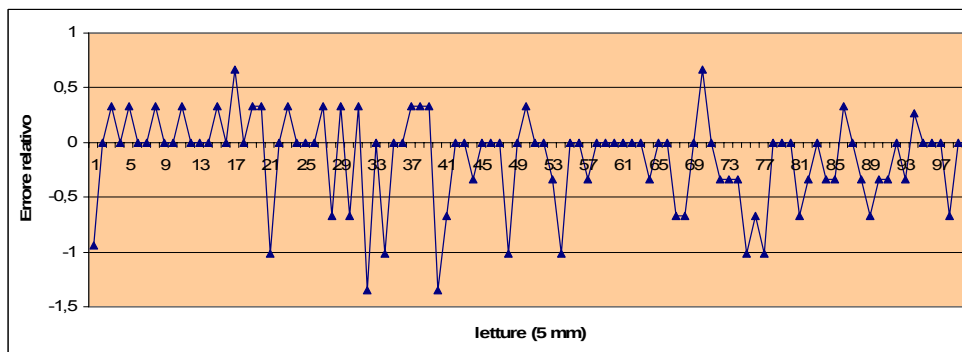


Figura 4.5: Errore relativo dato dal rapporto tra errore assoluto e il valore letto in %

$$Err_{rel} = \frac{Err_{abs}}{q_k} * 100$$

4.2 Gradi di libertà del sistema

Nel campo ingegneristico e aeronautico i gradi di libertà descrivono la flessibilità del movimento. Un sistema che ha completa libertà di movimento ha sei gradi di libertà: tre modi di traslazione (lungo le tre dimensioni X, Y, Z) e tre modi di rotazione (la capacità di cambiare angolo lungo i tre assi perpendicolari).

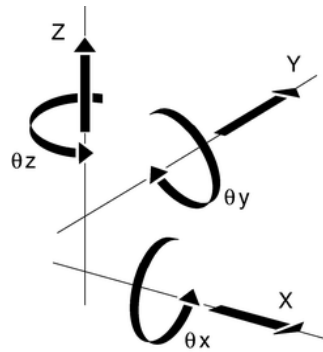


Figura 4.6: Gradi di libertà

Il sistema a disposizione è in grado di agire su cinque gradi di libertà (si possono montare tre piastre di traslazione in configurazione XYZ e scegliere lungo che asse è possibile sacrificare la rotazione). Non sarà necessario tuttavia, per il lavoro in questione, muoversi lungo l'asse parallelo al fascio di particelle; la rotazione lungo questo stesso asse potrà inoltre essere corretta analiticamente facendo uso delle coordinate XY sul piano ad esso perpendicolare. Si è deciso quindi di utilizzare 2 attuatori per la traslazione XY del campione sul piano trasverso alla linea di fascio, un attuttore per il movimento di un collimatore e la piastra "Tip and Rotation" con due micrometri manuali per l'aggiustamento di eventuali angoli di rotazione del campione.

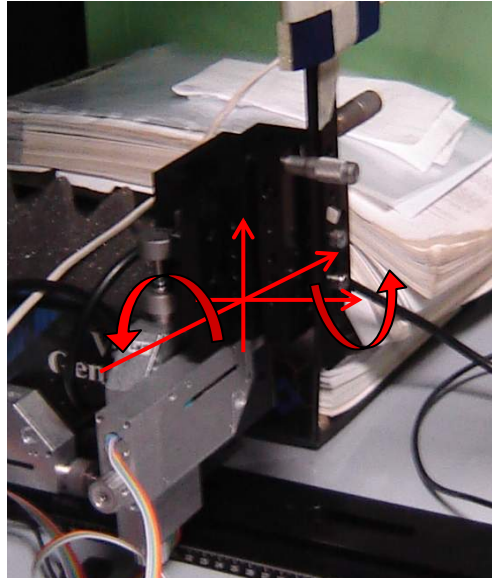


Figura 4.7

4.3 Allineamento del sistema di movimentazione al fascio

Per rispondere in maniera efficiente alla necessità di precisione del sistema, è doveroso operare in modo da raggiungere un allineamento quanto più possibile accurato. Se si vuole quindi prevedere di muovere un oggetto in maniera perfettamente orizzontale o verticale rispetto ad un sistema di riferimento, è necessario che vengano presi alcuni accorgimenti fondamentali.

Un buon inizio è accertarsi che le piastre meccaniche siano saldamente fissate tra loro agendo in maniera decisa su viti e bulloni, è utile inoltre fare largo uso di rondelle per assicurare una maggiore superficie di contatto e quindi conferire solidità alla struttura.

Le piastre sono provviste di fori filettati misura M6 o fori passanti con diametro Φ 6 mm distanziati tra loro di 25 mm, in alcuni casi è stato necessario utilizzare fori adiacenti e ciò ha comportato l'introduzione di errori sistematici a cascata per ogni stage montato sul precedente. Nonostante si possa prestare attenzione al montaggio accurato dei componenti, è inevitabile che l'errore si accumuli sempre di più ogni volta che alla struttura viene aggiunto un componente.

Bisogna inoltre prendere in considerazione che la struttura verrà poi posta sopra un piano che non sarà perfettamente parallelo e quindi ogni tipo di tentativo di raggiungere un

allineamento perfetto sotto il millimetro con strumenti esclusivamente meccanici è poco realistico.

Risulta invece più comodo montare le piastre nel miglior modo possibile e andare ad apprezzare di quanto il sistema si discosta dall'allineamento ideale per poi procedere con le correzioni necessarie.

4.3.1 Simulazione con collimatore spesso e fascio laser

Come prima esperienza di messa in opera di un sistema completo di piastre di movimentazione (montate in struttura complessa X-Y-Z) e attuatori, si è proceduto con l'assemblaggio su banco ottico di un sistema per simulare le condizioni reali del sistema.

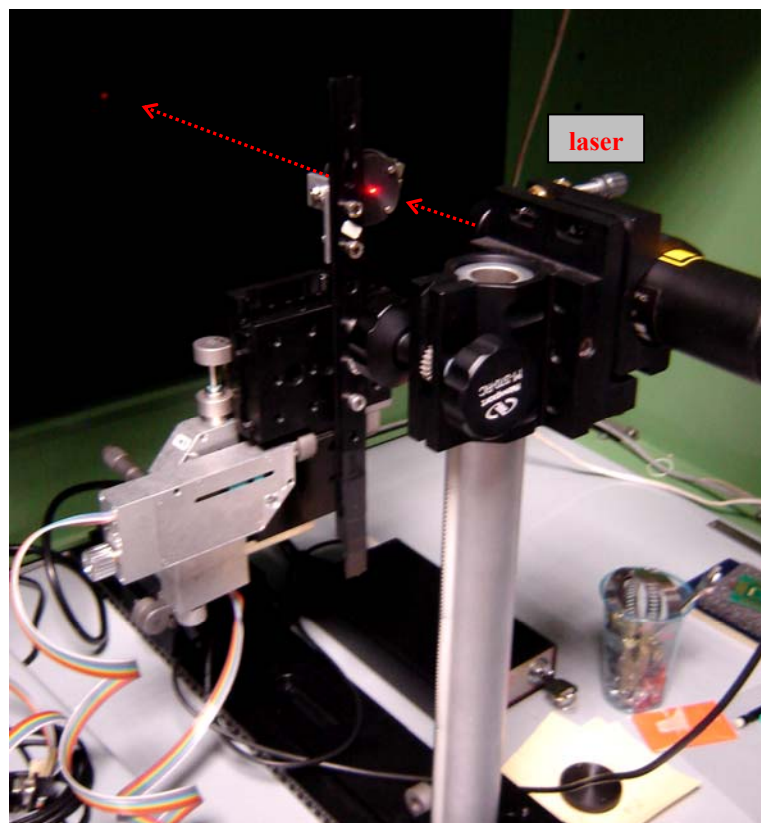


Figura 4.8: Il fascio di luce laser attraversa il collimatore in PVC e colpisce lo schermo

Le piastre traslazionali sono state assemblate tra loro in modo da poter permettere libertà di movimento lungo i 3 assi perpendicolari X-Y-Z; ad ogni piastra è stato inserito

un attuatore lineare comandato tramite il software su PC ed è stata inoltre montata la piastra di correzione angolare per correggere eventuali angoli di deviazione.

Per simulare il fascio di particelle è stato montato un laser da laboratorio su un'asta lungo il binario che sosteneva tutta la struttura.

Solidale alle piastre di movimentazione è stato poi montato un disco in PVC nero con foro centrale di 800 μm ; questa configurazione ha permesso di poter simulare il movimento di un collimatore lungo la linea di fascio.

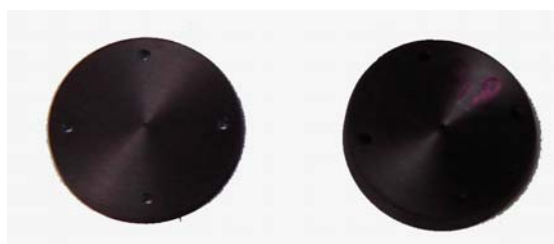
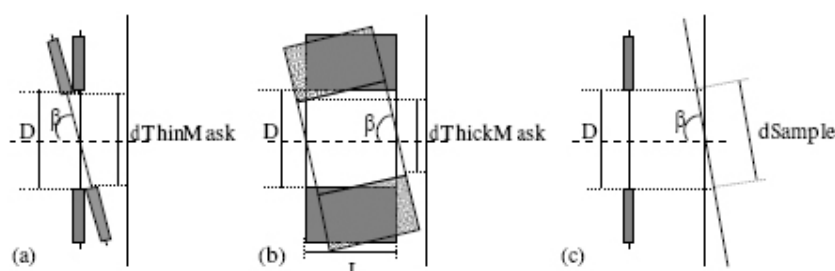


Figura 4.9: collimatori in PVC

Il diametro del foro, piccolo rispetto lo spessore del disco in PVC (10 mm), ha permesso di apprezzare in maniera immediata come l'allineamento del sistema ricopre un'importanza fondamentale ai fini della riuscita dell'esperimento. Nel caso particolare di un collimatore non perfettamente allineato con il fascio, si ha una distorsione della forma del fascio stesso in uscita (da circolare a ellittico); nel caso peggiore il fascio rimane intrappolato all'interno del collimatore e non raggiunge lo schermo di osservazione.



**Figura 4.10: a) un collimatore sottile, anche se inclinato, non provoca rilevanti riduzioni;
b) un collimatore spesso, se inclinato, causa alterazione alla geometria del fascio;
c) se il collimatore è allineato ma il campione è inclinato si ha lo stesso effetto del caso a.**

Raggiunta la condizione di perfetto allineamento, mostrata in figura, si è proceduto quindi imponendo al sistema una serie di movimenti ripetuti, in modo da poter apprezzare quanto questi potessero influire sulla ripetibilità del sistema stesso.

Per questa prima esperienza ci si è affidati ad un metodo di misura esclusivamente ottico, senza ricorrere all'utilizzo di strumenti più raffinati come ad esempio un sistema di acquisizione video; tuttavia l'esperimento ha permesso di apprezzare direttamente i disagi che un non perfetto allineamento del sistema globale produce.

L'utilizzo di una videocamera è stato invece necessario ai fini dell'esperienza riportata in seguito.

4.3.2 Simulazione con pattern cartaceo e visualizzazione su monitor

In questa esperienza si è proceduto applicando direttamente sulle piastre un pattern di precisione, in modo da poter apprezzare quanto gli scostamenti tra la posizione impostata e quella realmente raggiunta dagli attuatori fosse affetta da errore.

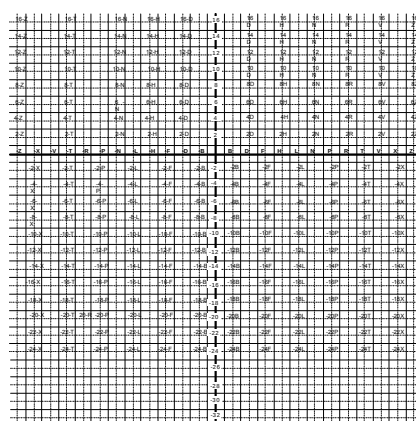


Figura 4.11: Immagine pattern. La distanza tra le linee è 1 mm

In questo caso il pattern, stampato su carta, è stato applicato direttamente sulle piastre e la stessa immagine è stata inserita nell'interfaccia grafica del programma (come spiegato nel precedente capitolo), dopo essere stata opportunamente ridimensionata.

Sul foglio è stato poi proiettato il fascio laser e il punto di incidenza è stato osservato su monitor tramite una telecamera orientata su di esso. L'utilizzo della telecamera si è reso necessario in quanto la luminosità e le dimensioni del fascio laser non permettevano un'osservazione diretta dello spot in uscita sul pattern, nonostante l'utilizzo di speciali occhiali protettivi da laser; la telecamera ha permesso invece di osservare la zona di incidenza con un'ottimale rapporto di ingrandimento, e la chiusura del diaframma ha permesso di ridurre il fenomeno di abbagliamento.

Tutto ciò ha permesso di operare in condizioni di sicurezza. La struttura composta da piastre e laser è stata inoltre coperta con un panno scuro e ciò ha permesso di simulare la movimentazione del sistema come se fosse posto in un luogo lontano dall'operatore.

Imponendo spostamenti di qualche centimetro, lungo gli assi X-Y è stato possibile apprezzare il non perfetto allineamento del pattern rispetto il piano ottico e si è potuto correggere l'angolo di deviazione agendo sul micrometro posto sulla piastra rotazionale.

Essendo la distanza tra le varie linee orizzontali e verticali di 1 mm è stato possibile verificare gli spostamenti per distanze multiple di 500 μm , in modo da poter riconoscere il punto di arrivo del fascio su una linea, un'intersezione o il punto centrale dei quadrati.

4.3.3 Collimazione del fascio in tre fasi: 2mm, 1mm, 50 μm

Il fascio giunge alla camera con un diametro che viene ridotto in fasi successive: il lungo collimatore denominato “*cannocchiale*” posto all'ingresso della camera (1 m) ha fori di entrata e uscita da 2 mm, il precollimatore (10 cm) ha fori da 1 mm e il collimatore finale (sottile lamina di Al con foro da 50x100 μm^2), che è posto a pochi mm dalla superficie del rivelatore.



Figura 4.12: Il collimatore finale montato sul supporto

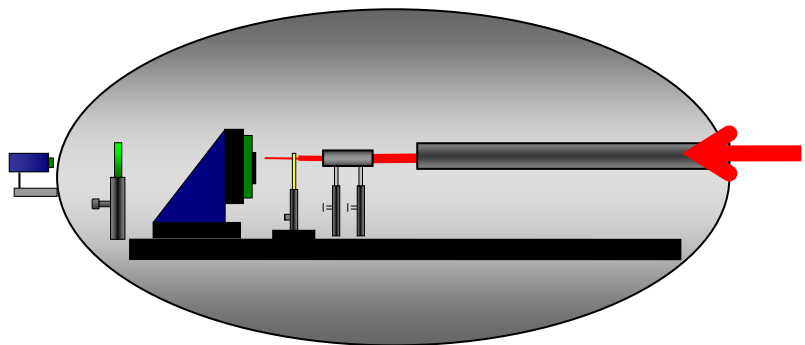
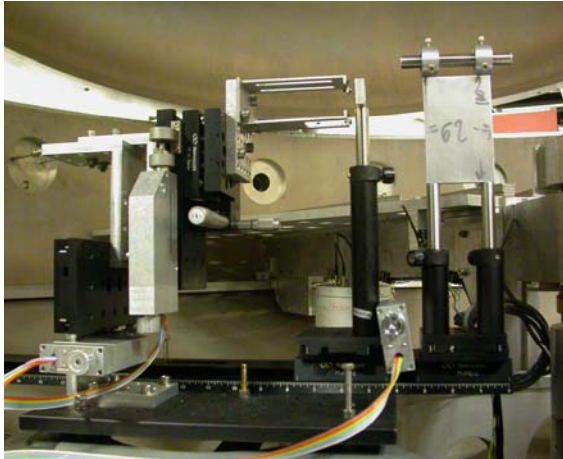


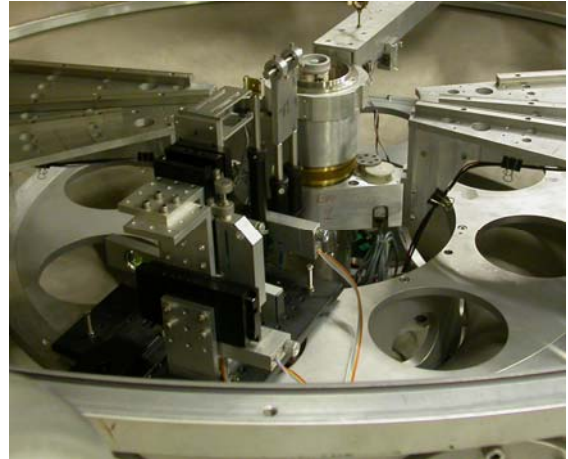
Figura 4.13: Posizionamento della struttura all'interno della camera e riduzione del fascio in tre fasi

4.4 Montaggio del sistema finale, soluzione adottata

Il sistema composto da piastre e attuatori è stato assemblato in modo da poter essere inserito all'interno della camera a vuoto; la configurazione delle piastre, diversa dalle soluzioni finora adottate, nasce dalla necessità di disporre di un sistema molto compatto, con ingombro in altezza ridotto.



**Figura 4.14: Vista della struttura
dal piano di appoggio**



**Figura 4.15: Vista della struttura
dell'esterno della camera**

Tutto il sistema è stato assemblato su una base quadrata, la *breadboard*, e successivamente posto sul piano goniometrico della camera stessa. Sulla breadboard sono stati fissati quattro viti per la regolazione dell'inclinazione e un asse filettato con il quale si è proceduto al fissaggio sul piano goniometrico.

La soluzione adottata provvede alla sistemazione sul binario del precollimatore e del collimatore finale, le piastre invece sono state fissate a partire da un punto della breadboard decentrato rispetto il binario, a sua volta in linea con l'asse del fascio.

Questa soluzione si è resa necessaria in quanto il montaggio direttamente sul binario avrebbe comportato un ingombro in altezza superiore rispetto a quello consentito.

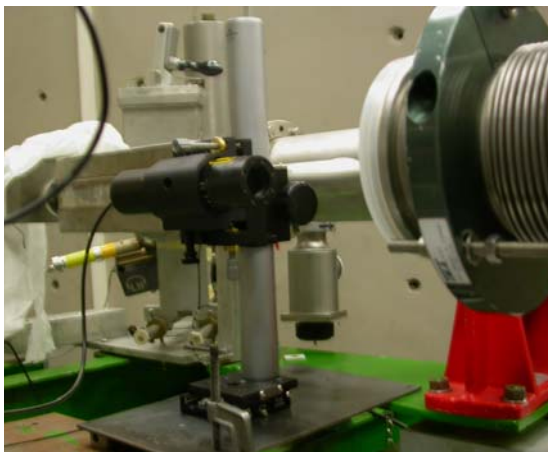
4.4.1 Allineamento ottico del sistema

Dopo aver posto il sistema di piastre e attuatori all'interno della camera, si è proceduto con l'allineamento ottico alla linea di fascio. A tal scopo è stato utilizzato un

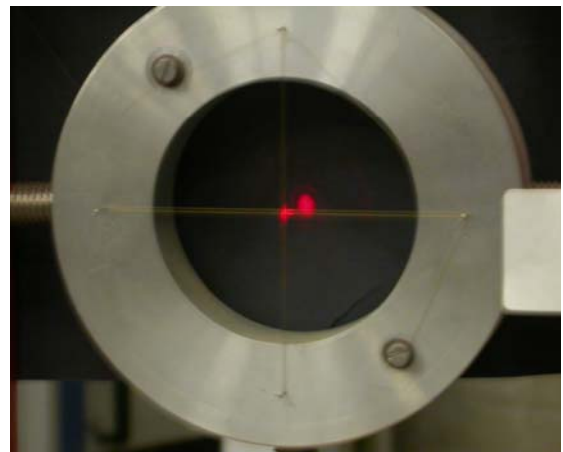
laser, posto sulla linea di fascio in un punto in cui era aperto un manicotto, e un *teodolite* (strumento ottico a cannocchiale utilizzato per la misurazione di angoli). Il laser, posto a monte della camera a vuoto, proietta un fascio luminoso che attraversando la camera va successivamente a incontrare a valle il teodolite.

Per identificare una linea nello spazio sono necessari due punti, nel nostro caso si è scelto come primo punto una crocetta posta lungo la linea di fascio poco distante dal teodolite, per trovare il secondo si è proceduto tramite approssimazioni successive, muovendo il laser di frazioni di angolo grazie alle viti di regolazione fine, puntando il laser verso la crocetta del teodolite e osservando il foro di uscita del laser (spento) e i collimatori dal teodolite stesso.

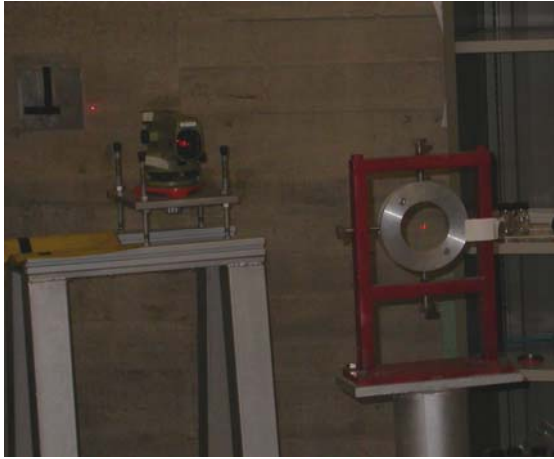
L'allineamento effettuato ha dato come risultato una retta allineata per tre punti: sorgente luminosa, crocetta e centro di focalizzazione del teodolite.



**Figura 4.16: Il laser inserito
nella linea di fascio**



**Figura 4.17: Proiezione del fascio
laser sulla crocetta**



**Figura 4.18: Il fascio laser allineato
attraversa il teodolite**



**Figura 4.19: Il fascio laser attraversa
le estremità del precollimatore**

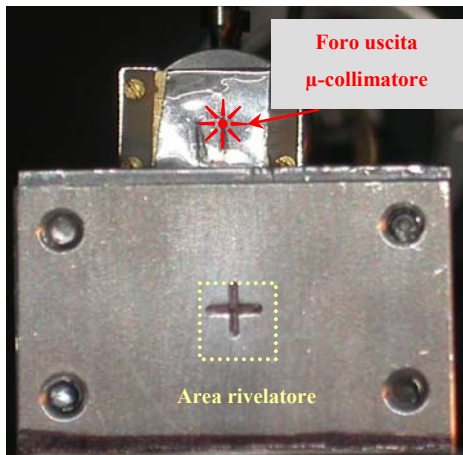
Simulata la linea di fascio grazie al laser si è dunque allineato il precollimatore, composto da un tubo cilindrico in alluminio lungo 10 cm alle cui estremità erano stati praticati due fori dal diametro di 1 mm.

4.4.2 Allineamento del prototipo

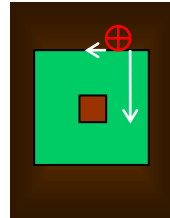
Dopo aver effettuato l'allineamento del sistema di piastre ed attuatori alla linea di fascio, si è proceduto con l'allineamento della staffa su cui è stato montato il rivelatore. Questo procedimento si è reso necessario in quanto ci si è dovuti assicurare che l'orientamento orizzontale e verticale del campione fosse quanto più possibile parallelo alla direzione degli assi di movimento degli attuatori per gli spostamenti sulle direzioni X e Y.

Inoltre, per inserire il prototipo sotto il fascio di particelle, era pure necessario assicurarsi che i movimenti degli attuatori fossero quanto più possibile ridotti, per non stressare in modo eccessivo le molle antagoniste delle piastre.

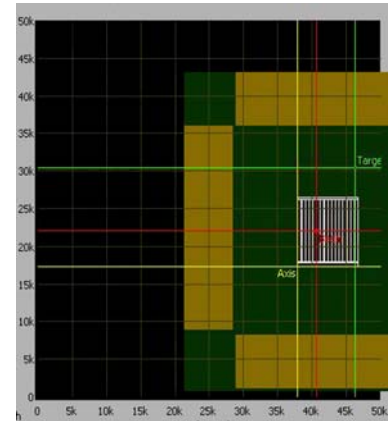
Per fare questo si è osservato dal teodolite (nel verso opposto a quello del fascio) l'ingombro della staffa di sostegno, si è dunque trovata una posizione che limitasse al minimo indispensabile la corsa degli attuatori.



**Figura 4.20: Staffa di supporto
vista dal teodolite.
La croce indica il centro
del rivelatore**

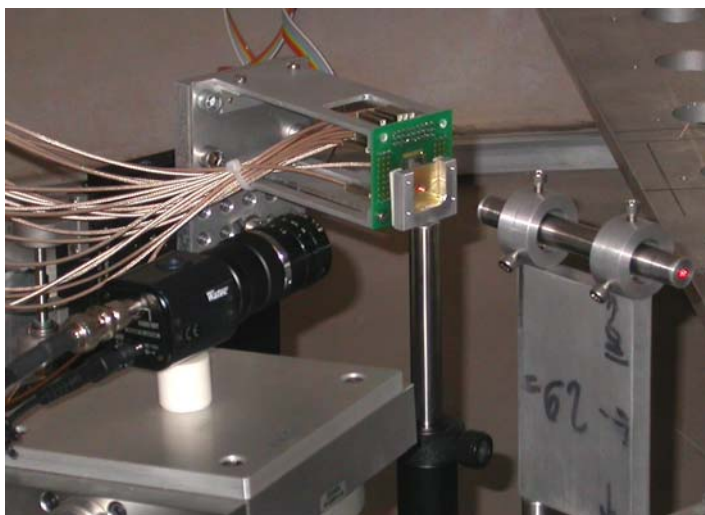


**Figura 4.21: Posizione
del fascio con attuatori
a fine corsa negativo
(vista frontale)**

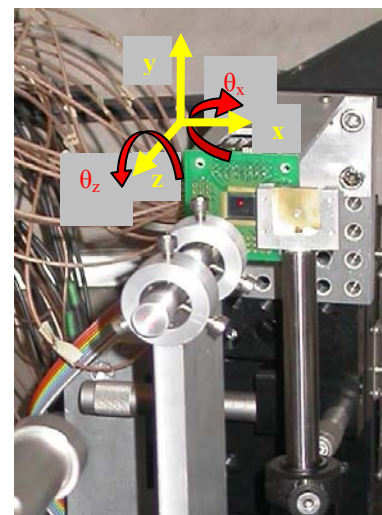


**Figura 4.22: Escursione max
per gli attuatori ottenuta:
per X 13 mm, per Y 33 mm**

Successivamente si è predisposta una telecamera orientata sul campione che inquadrasse la zona attiva del rivelatore colpita dal fascio laser incidente. Imponendo al sistema traslazioni lunghe quanto il bordo del campione stesso, si è osservato tramite la telecamera come il fascio rimanesse all'interno del bordo per tutto il movimento imposto, in caso contrario, si sarebbe corretto l'angolo di inclinazione agendo direttamente sui micrometri della piastra di rotazione Tip & Rotation.



**Figura 4.23: La videocamera, montata su
una piastra micrometrica, inquadra il punto
di incidenza dello spot laser sul bordo del rivelatore**



**Figura 4.24: Deviazioni d'angolo che
è possibile correggere
con i micrometri**

La procedura di allineamento ottico, effettuata con il laser, ha permesso di ottenere un allineamento del prototipo entro un errore accettabile, considerando infatti 1 mm il diametro dello spot del laser in uscita dal precollimatore, si può considerare la quantità pari alla sua metà (500 μm) come massima risoluzione osservabile.

Essendo riusciti ad allineare sulla stessa linea ideale i punti corrispondenti a inizio e fine bordo, lontani tra loro 1 cm, si può affermare che l'angolo di deviazione di cui è affetto il sistema dopo l'allineamento ottico è $\theta_z \leq \theta_{z\text{Max}} = \text{Arctg}\left(\frac{500}{10000}\right) = 2,86^\circ$.

Tale valore sarà successivamente verificato per via sperimentale.

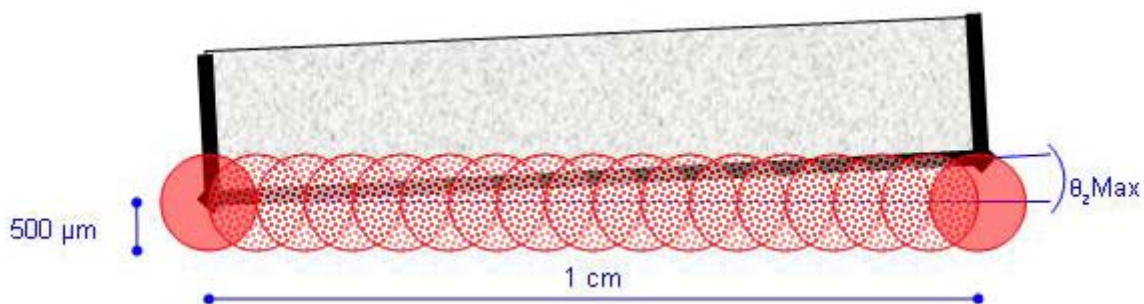


Figura 4.25

4.4.3 Creazione del file di inizializzazione

Trovato il giusto allineamento del prototipo si è proceduto con la creazione del file di inizializzazione per assegnare i valori corretti agli encoders degli attuatori, utilizzando la seguente procedura per gli assi X e Y:

- o Dalla posizione corrente si è premuto il pulsante “Set Zero” per azzerare il valore dell'encoder e della coordinata attuale
- o Si è raggiunto il limite di corsa negativo e si è letto il valore dell'encoder (-12,1089 per X e -32,6986 per Y); tali valori verranno salvati nel file di inizializzazione nella sezione “Encoder Position”
- o Sempre nel file di inizializzazione sono stati introdotti nella sezione “Zero Position” i complementari (in valore assoluto) a 50 (dimensioni del grafico) dei valori degli encoders letti (37,8911 per X e 17,3014 per Y), che saranno visualizzati tramite il cursore “Axis” come assi di riferimento

4.4.4 Diagnostica fascio

In fondo alla camera a vuoto, lungo la linea di fascio, è stato inoltre montato su un apposito supporto un cristallo a fibre ottiche scintillanti (*Scintillating Fiber Optic Plate, SFOP*) utilizzato come schermo su cui fare incidere il fascio di particelle, per poterne osservare dimensione e geometria.

Gli *SFOP* sono costruiti mediante l'assemblaggio di un fascio di fibre scintillanti di vetro drogato al Terbio, disposte in geometria esagonale e poi tagliate longitudinalmente; con questa struttura la luce prodotta all'interno di ciascuna fibra per effetto di scintillazione (sotto l'effetto di un rilascio di energia dovuto ad una particella incidente) rimane in buona parte intrappolata nella stessa e viene trasmessa alle estremità.

Nella struttura sono presenti a intervalli regolari alcune fibre di colore nero, dette *EMA (Extra Mural Absorber)*, che assorbono la luce che non è stata intrappolata dalle fibre adiacenti, con la conseguenza che si ha un netto miglioramento della risoluzione spaziale rispetto ad uno schermo di tipo convenzionale (CsI oppure NaI, Allumina, ecc.).

Lo *SFOP* a nostra disposizione ha geometria quadrata, di dimensioni 25x25x2 mm³. Le dimensioni della singola fibra sono di 10 µm di diametro per 2 mm di lunghezza. [8]

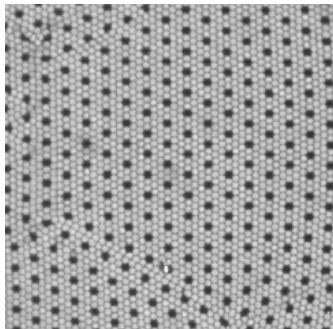


Figura 4.26: Osservando lo SFOP al microscopio è possibile riconoscere la struttura a nido d'ape

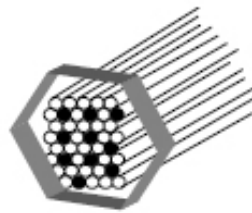


Figura 4.27: "Slice" trasversale di un fascio di fibre ottiche



Figura 4.28: Fascio laser incidente sullo SFOP

Una telecamera posta fuori dalla camera a vuoto, orientata sullo *SFOP* attraverso un cristallo di quarzo inserito in un oblò, ha permesso di acquisire su monitor il profilo del fascio incidente sul prototipo da caratterizzare, permettendo inoltre di agire sulla posizione

del microcollimatore finale per massimizzare la quantità di luce prodotta in uscita e trovare quindi il punto esatto di allineamento.

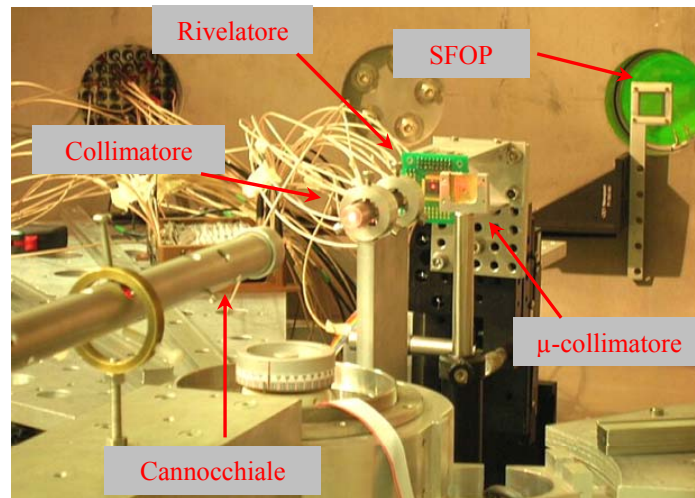


Figura 4.29: Setup completo del sistema

4.5 Rototraslazione del sistema di riferimento cartesiano

4.5.1 Richiami generali

A partire da un qualsiasi sistema è possibile effettuare un cambiamento dei punti di riferimento, basta conoscere le relazioni per cui il nuovo sistema di riferimento si lega al precedente preso in considerazione. Un esempio tipico e molto comune è il cambiamento di coordinate da un sistema di tipo cartesiano ortogonale ad uno di tipo polare.

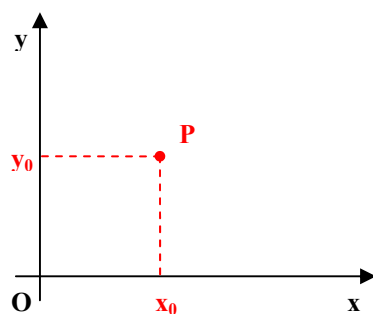


Figura 4.30: Riferimento Cartesiano.

Il punto P è individuato dalle coordinate (x_0, y_0) , che rappresentano le distanze dall'origine O delle proiezioni di P sui due assi

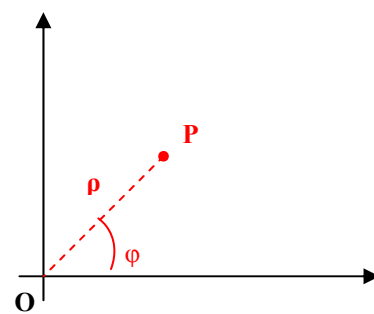


Figura 4.31: Riferimento Polare.

Il punto P è individuato dalla coppia di valori (ρ, φ) , rispettivamente la distanza di P da O e l'angolo sotteso dal segmento OP con l'asse polare.

Per passare dal sistema cartesiano xOy al sistema polare, o viceversa, si usano le seguenti relazioni:

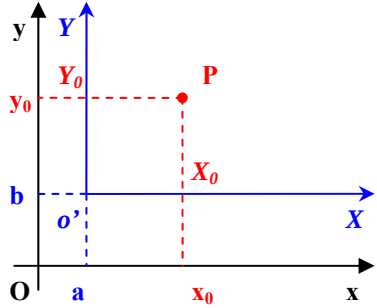
Tabella 4-2: Relazioni di conversione

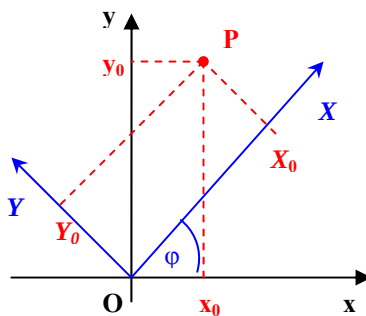
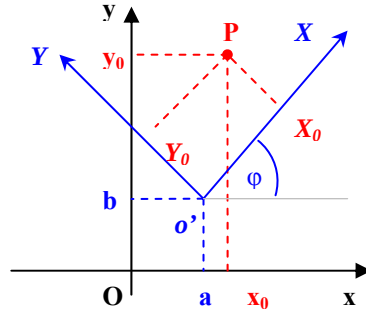
Polare $\rightarrow$ Cartesiano	Cartesiano $\rightarrow$ Polare
$\begin{cases} x_0 = \rho \cdot \cos \varphi \\ y_0 = \rho \cdot \sin \varphi \end{cases}$	$\begin{cases} \rho = \sqrt{x_0^2 + y_0^2} \\ \varphi = \arctg \frac{y_0}{x_0} \end{cases}$

Non è raro, nel campo ingegneristico in genere, avere a che fare con varie forme di conversioni da un sistema di riferimento ad uno più adatto; ciò permette di riconoscere univocamente le coordinate di un dato punto nel piano o nello spazio, utilizzando però coordinate più idonee all'applicazione da sviluppare.

Dato un sistema di riferimento assoluto, è probabile che non si adatti alla situazione che deve rappresentare, può accadere infatti che esso sia traslato, ruotato di un certo angolo o entrambi.

Tabella 4-3: Formule di trasformazione

Formule per la traslazione degli assi: Le coordinate del generico punto P sono: $P(x_0, y_0)$ nel sistema di assi ortogonali xOy, e $P(X_0, Y_0)$ nel sistema di assi paralleli concordi X_0Y_0. Se l'origine del nuovo sistema X_0Y_0 ha, rispetto al primo, le coordinate $o'(a, b)$, valgono le relazioni: $\begin{cases} x_0 = a + X_0 \\ y_0 = b + Y_0 \end{cases}$	 Figura 4.32: Traslazione
---	--

Formule per la rotazione degli assi Le coordinate del generico punto P sono: $P(x_0, y_0)$ nel sistema di assi ortogonali xOy, e $P(X_0, Y_0)$ riferite al sistema $XO'Y$ in cui gli assi sono ruotati di un angolo φ, e le origini O e o' coincidono. Le relazioni: $\begin{cases} x_0 = X_0 \cdot \cos \varphi - Y_0 \cdot \sin \varphi \\ y_0 = X_0 \cdot \sin \varphi + Y_0 \cdot \cos \varphi \end{cases}$ consentono di passare da un sistema di riferimento xOy al sistema $XO'Y$ ruotato di un angolo φ rispetto al precedente.	 Figura 4.33: Rotazione
Formule per la rototraslazione degli assi Questo movimento risulta composto da una traslazione ed una rotazione: $\begin{cases} x_0 = a + X_0 \cdot \cos \varphi - Y_0 \cdot \sin \varphi \\ y_0 = b + X_0 \cdot \sin \varphi + Y_0 \cdot \cos \varphi \end{cases}$	 Figura 4.34: Rototraslazione

4.5.2 Funzioni di allineamento adottate durante il turno di misura

Durante il turno di misura effettuato sono stati sottoposti a caratterizzazione i prototipi R328-1 e R328-3, i quali (non considerando una lieve differenza nel layout geometrico dovuta alla presenza del contatto di massa in posizioni diverse) presentano uguale struttura geometrica: 16 strip verticali e una strip orizzontale.

Il prototipo R328-2, con layout di 32 strip verticali (due gruppi da 16) e una strip orizzontale in mezzo, non è stato sottoposto a misure.

4.5.3 Letture dei conteggi per allineare il primo rivelatore

Dopo aver chiuso la camera a vuoto e fatto giungere in essa il fascio di particelle, si è proceduto con il controllo strumentale per apprezzare l'entità dell'errore di allineamento effettuato con la procedura descritta precedentemente.

Ponendo l'origine degli assi di riferimento sull'angolo in basso a sinistra dello schema del rivelatore, e considerando le dimensioni delle strip all'interno dell'area attiva, si ottiene una mappa di coordinate X-Y per le 16 strip verticali, raccolta nella tabella seguente. La strip orizzontale non è stata presa in considerazione in quanto non sarebbe stata sottoposta a caratterizzazione.

Tabella 4-4: Mappa coordinate X-Y del rivelatore.

Tutti i valori sono espressi in μm

Strip	X inizio	X centro	X fine	Y inizio	Y fine
1	495	763	1031	1155	9410
2	1060	1328	1596	1155	9410
3	1625	1893	2161	1155	9410
4	2190	2458	2726	1155	9410
5	2755	3023	3291	1155	9410
6	3320	3588	3856	1155	9410
7	3885	4153	4421	1155	9410
8	4450	4718	4986	1155	9410
9	5015	5283	5551	1155	9410
10	5580	5848	6116	1155	9410
11	6145	6413	6681	1155	9410
12	6710	6978	7246	1155	9410
13	7275	7543	7811	1155	9410
14	7840	8108	8376	1155	9410
15	8405	8673	8941	1155	9410
16	8970	9238	9506	1155	9410

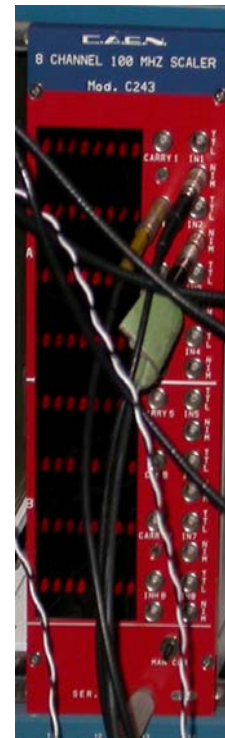


Figura 4.35: Il contatore CAEN C243 è un modulo CAMAC con 8 contatori indipendenti a 8 cifre, ogni contatore accetta input di tipo TTL o NIM e può lavorare alla frequenza massima di 100 MHz

Ci si è posti poi sulla strip numerata come 10, che secondo i calcoli avrebbe dovuto trovarsi tra le coordinate X:(5580, 6116), il cui centro strip presunto si trova alla coordinata X:(5848).

Si è poi tornati indietro alla coordinata X:(5248), esterna alla strip 10, e muovendosi a passi di 100 μm si è cominciato ad acquisire tramite un contatore il numero di particelle rivelate dalla strip 10, per ogni coordinata impostata.

Per ogni punto sono state effettuate più misure, ad una counting rate nell'ordine delle centinaia di particelle al secondo, in modo da poter considerare una media ponderata invece di una lettura singola.

Procedendo in questo modo si è giunti alla coordinata X:(6548); è stato dunque esplorato un segmento di 1300 μm , lungo più del doppio dello spessore di una strip (536 μm), e si sono visualizzati in un grafico i valori corrispondenti per ogni misura.

Dal grafico si è potuto osservare che il numero conteggi è molto basso nelle posizioni esterne alla strip, si ha poi un rapido incremento con un punto di massimo, poi una diminuzione sempre più accentuata per posizioni sempre più distanti dal centro della strip. Leggendo la coordinata del punto di massimo assoluto del grafico, si è visto X:(5948) risulta essere 100 μm più a destra della coordinata preventivamente calcolata.

Si è proceduto con la stessa misurazione sulla strip 10 per un'altra ordinata 3548 μm più in alto della precedente e si è giunti all'individuazione del centro strip alla coordinata X:(5960), solo 12 μm più a destra; effettuando i calcoli l'angolo di deviazione risulta pari a $0,19^\circ$ e si è considerato questo errore di allineamento come trascurabile.

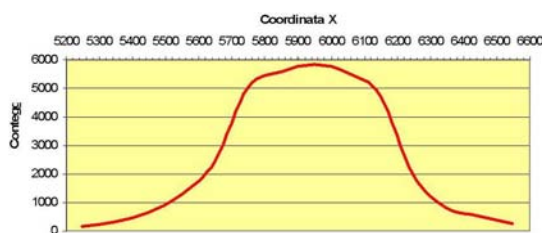


Figura 4.36

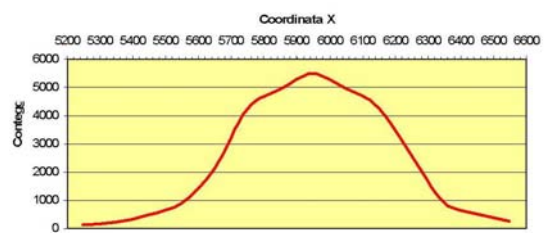


Figura 4.37

Dopo aver apprezzato lo spostamento sull'asse X si è proceduto nel rilevare quello sull'asse Y, ponendosi al centro della strip 10 e analizzando il numero conteggi lungo le zone di inizio e fine strip. Conoscendo la dimensione verticale della strip, 8255 μm , si è trovato il punto esatto di inizio e fine, sovrapponendo una barra di tale misura al grafico che è stato costruito con i conteggi rilevati.

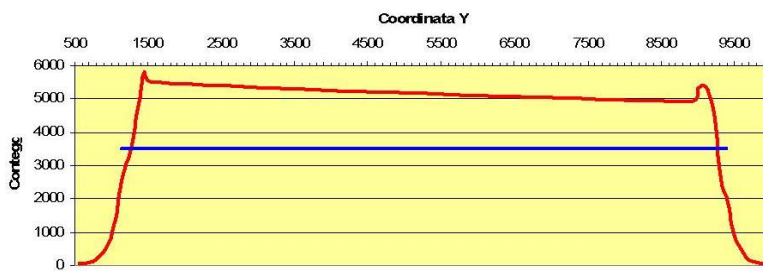


Figura 4.38

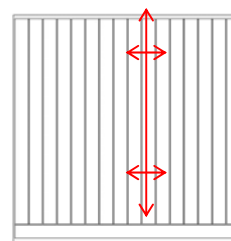


Figura 4.39: Esplorazioni sulla strip 10

Si è giunti ad accertare che la strip, grazie ai dati raccolti, comincia e finisce alle coordinate Y:(1135, 9390), invece di Y:(1155, 9410) come previsto.

Tabella 4-5

	Coordinate Previste	Coordinate Reali	Shift μm
X:(inizio-fine)	5580 - 6116	5680 - 6216	+ 100
Y:(inizio-fine)	1155 - 9410	1135 - 9390	- 20

Apprezzati tali valori si è proceduto con le correzioni sul sistema di riferimento, si è dunque modificato il file di inizializzazione degli attuatori, in modo da imporre agli encoders degli attuatori X e Y un offset di -100 e +20 μm .

La corretta individuazione delle coordinate dei punti del rivelatore ha permesso inoltre di poter ridimensionare in modo accurato l'immagine del layout che è stata inserita nell'interfaccia di comando per gli attuatori; posizionandosi infatti con i cursori nelle coordinate mostrate nella tabella infatti è stato possibile far combaciare la coordinata impostata con la zona del rivelatore che la rappresentava.

4.5.4 Allineamento strumentale del secondo rivelatore

Dopo aver effettuato le misure previste sul primo rivelatore, si è aperta la camera a vuoto per sostituire nella struttura di movimentazione il secondo rivelatore da sottoporre a caratterizzazione.

In questo caso non è stato possibile effettuare nuovamente l'allineamento ottico del rivelatore con il fascio, in quanto ciò avrebbe reso necessario aprire nuovamente la linea di trasmissione, inserire il laser e tutta la procedura di allineamento laser già descritta precedentemente.

Tuttavia un metodo di allineamento sarebbe certamente stato necessario, in quanto lo smontaggio della staffa di supporto finale per posizionare il nuovo rivelatore e il successivo riassettaggio avrebbe sicuramente introdotto un errore non prevedibile; una diversa forza impressa alle viti di fissaggio avrebbe sicuramente introdotto uno spostamento e un angolo di deviazione.

Per risolvere questo problema si è dunque proceduto con la misura di ricerca dell'errore di posizionamento, analogamente a come è stato fatto per il primo rivelatore, misurando i conteggi su una strip a due ordinate diverse per trovarne il centro.

Questa procedura ha permesso di apprezzare pure l'angolo di deviazione del rivelatore rispetto al precedente, in quanto i due centri strip trovati differivano di una quantità Δx pari a $38 \mu\text{m}$.

Considerando che lo spostamento in verticale imposto Δy è stato di $3610 \mu\text{m}$ si è giunti alla conclusione che l'angolo di deviazione $\alpha = \text{Arctg}\left(\frac{\Delta x}{\Delta y}\right)$, pari a $0,6^\circ$.

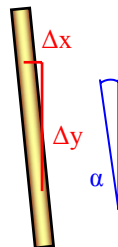


Figura 4.40

Tabella 4-6

	Shift
X	$38 \mu\text{m}$
Y	$12 \mu\text{m}$
α	$0,6^\circ$

Imponendo questi valori sperimentali al sistema, grazie alle formule di rototraslazione, si è proceduto con il calcolare le coordinate assolute al sistema.

5 Misure effettuate

Vengono di seguito presentati i risultati del turno di misura effettuato, soffermandosi sulla validità dei dati acquisiti e mostrando come il sistema ha risposto in modo efficiente ai termini di accuratezza richiesti.

5.1 Tabella posizioni

Come già anticipato nel Capitolo 2, i rivelatori utilizzati durante la misura sono stati i modelli R328-1 e R328-3, con uguale layout geometrico di 16 strips verticali e una orizzontale (ad eccezione del contatto di massa comune che nel modello R328-1 è posto in alto, lontano dalla strip orizzontale).

Per ogni prototipo è stata eseguita l'acquisizione dati in 16 differenti posizioni, su quattro strips diverse.

Inizialmente è stato osservato il profilo generale del segnale indotto su una strip; nelle figure seguenti sono riportati i segnali prodotti dall'oscilloscopio, acquisiti dalla strip 2 quando vengono irraggiate strips diverse.

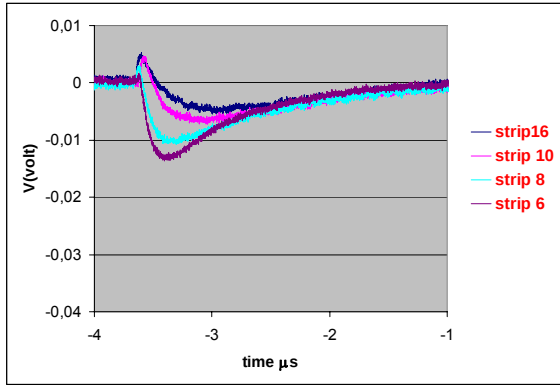


Figura 5.1: Induzione sulla strip 2 quando vengono colpite le strips 16, 10, 8 e 6

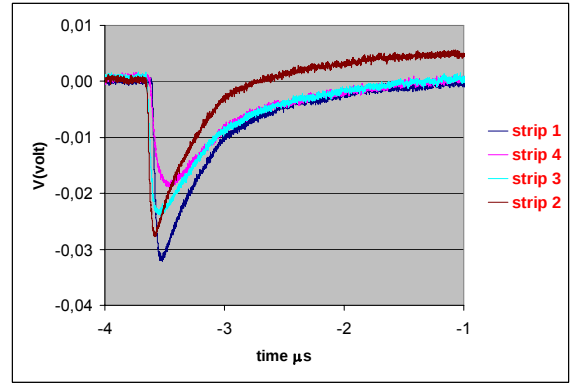


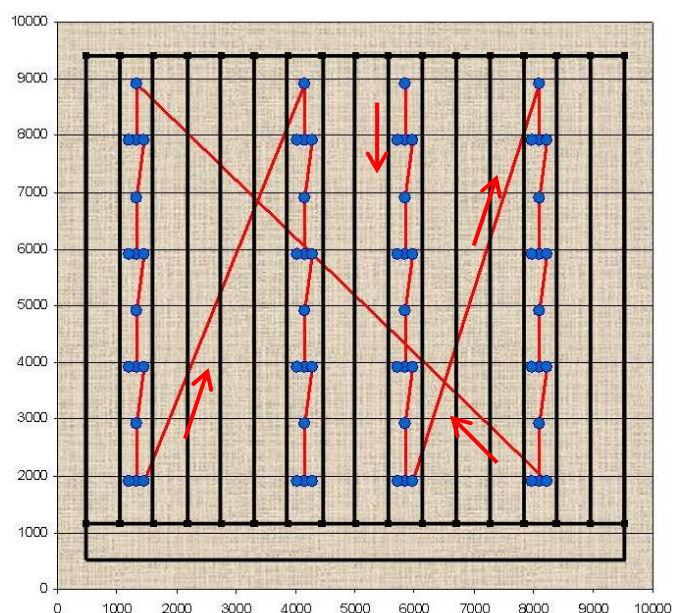
Figura 5.2: Induzione sulla strip 2 quando vengono colpite le strips 4, 3, 2 e 1

Quando lo stadio E viene colpito da una particella, si genera un segnale indotto su tutte le altre strips; la caratteristica del segnale indotto cambia in funzione della distanza della strip irraggiata. Quando viene colpita una strip lontana l'induzione mostra una crescita rapida positiva (*rise time* nell'ordine dei μs) seguita da un segnale negativo che ritorna a zero lentamente.

Irraggiando strips sempre più vicine questa cresta positiva decresce in ampiezza, fino a scomparire quando si colpisce la strip 6; il segnale indotto dunque cambia nella forma in funzione della distanza del punto di impatto delle particelle cariche. Avvicinandosi alla strip osservata si nota una crescita del segnale indotto e il *rise time* del ritorno a zero è di circa $0,5 \mu s$. Quando si colpisce la strip osservata (strip 2) la crescita del segnale ΔE reale copre parzialmente il segnale indotto e il ritorno a zero è più veloce.

Questo andamento può essere compreso in generale pensando che le lacune generate dall'interazione di cariche all'interno dello stadio E , e raccolte dall'elettrodo di massa, producono un incremento locale del potenziale elettrico che perturba i campi elettrici di tutte le strips.

Dopo questi controlli preliminari sul comportamento del segnale indotto dello stadio ΔE , sono state irraggiate 4 strips in 16 punti: 8 sono stati scelti a centro della strip in modo da esplorare l'andamento per tutta la sua lunghezza, ogni punto distante 1 mm dal precedente. E' stato inoltre controllato, 4 volte per ogni strip, il comportamento dell'induzione lungo la coordinata orizzontale, colpendo due ulteriori punti a $\pm 130 \mu m$ dal centro strip.



**Figura 5.3: Nel corso del primo turno di misura
sono state esplorate rispettivamente le strips numero 10, 14, 2 e 7;
il grafico rappresenta le coordinate dei punti
e il tracciato dei movimenti relativi effettuati**

Le misure dell'induzione sono state effettuate per alcune strips, cambiando i valori degli stadi ΔE e E rispettivamente da 5 a 15 volt e da 70 a 200 volt.

Come previsto il valore dell'induzione decresce all'avvicinarsi al contatto di massa (in alto nel prototipo R328-1), si osserva inoltre un andamento quadratico in funzione della posizione; si può ottenere una precisa misurazione della posizione utilizzando i segnali indotti provenienti da strips non colpite.

Questa dipendenza in realtà non è molto semplice da trovare, in quanto dipende sia alle coordinate lungo l'asse X che lungo l'asse Y.

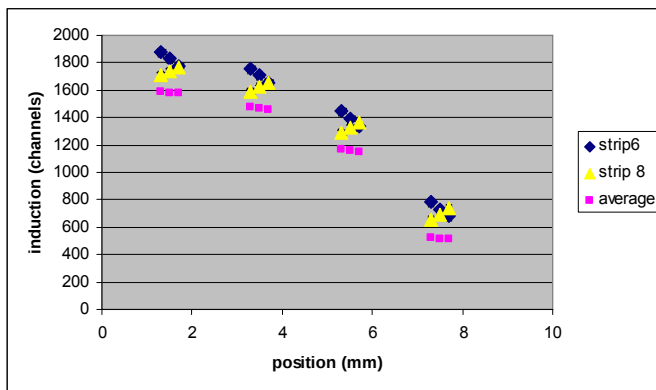


Figura 5.4: Induzione misurata sulle strips 6 e 8 durante l'irraggiamento della strip 7, e media degli eventi, riferita ai 4 punti in cui sono stati effettuati steps di $\pm 130 \mu\text{m}$ dal centro

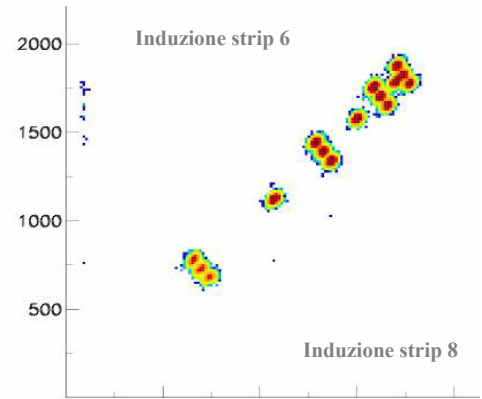


Figura 5.5: Combinazione induzioni delle strips 6 e 8 quando viene irraggiata la 7, si può notare una dipendenza non lineare in funzione della posizione; gli assi indicano le coordinate in μm

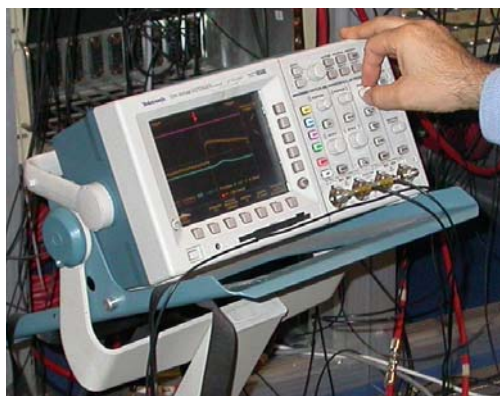
In Figura 5.4 si può osservare una dipendenza importante dell'induzione in funzione degli spostamenti di $\pm 130 \mu\text{m}$ lungo X e si notano due comportamenti opposti da parte delle strips adiacenti, tuttavia la media degli eventi (traslata in verticale di 200 unità per permetterne la visione nel grafico) è pressoché piatta.

Questo comportamento si può comprendere in modo semplice: l'induzione è prodotta dalla nube di lacune raccolte dal contatto di massa quando le particelle vengono rivelate nello stadio E. Questa nube influenza in modo rilevante le strips vicine, e muovendosi dalla strip 7 verso l'adiacente a sinistra (strip 6) l'induzione su questa crescerà ma allo stesso tempo diminuirà sulla strip 8; la media delle induzioni è quasi insensibile a questi spostamenti orizzontali. [1]

5.2 Elettronica

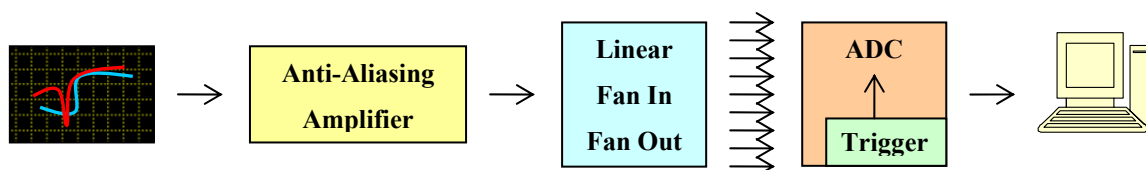
Le misure presentate nel paragrafo precedente sono il risultato dei dati acquisiti per ogni punto irraggiato nelle strips dei rivelatori. In fase di misura era infatti possibile riprodurre l'andamento dei segnali direttamente dall'oscilloscopio, ma non era possibile

analizzare in modo efficace la correlazione dei vari segnali, che ha portato successivamente ai risultati mostrati.



**Figura 5.6: Oscilloscopio digitale
utilizzato per la visione dei segnali**

Il sistema di acquisizione dati è composto essenzialmente da due *ADC* (CAEN V1719 e SIS 3301) che inviano i dati attraverso una connessione in fibra ottica ad un PC, che immagazzina i dati in files differenziandoli per canale.



**Figura 5.7: Schema a blocchi semplificato
del percorso acquisizione e salvataggio dati**

Poiché i segnali ΔE - E provenienti dalle strips hanno una dinamica molto veloce e sono soggetti a interferenze lungo il loro cammino, è necessario utilizzare un sistema che rigetti il rumore con estrema efficienza.

Essenzialmente il modulo “*Anti-Aliasing Amplifier*” funge da filtro passa-basso, tagliando le componenti spettrali del rumore bianco sopra una certa frequenza (attenuando di circa 24 dB per ottava, a 30 MHz si ha un’attenuazione di -3 dB); il segnale passa poi attraverso due moduli “*Linear Fan In – Fan Out*” (CAEN N625 e Le Croy 428F) che duplicano il segnale in diverse uscite: ciò è comodo in quanto ogni uscita può essere acquisita in modo indipendente da un canale diverso degli ADC.

Il modulo “*Fan In – Fan Out*” permette inoltre di regolare l’offset del segnale da inviare agli ADC: questi dispositivi permettono infatti di convertire i segnali che rientrano

nella loro dinamica (5 V per il SIS 3301, 1 V per il CAEN V1719) in digitale, ma non sono in grado di acquisire segnali con andamento sia positivo che negativo, per il segnale ΔE (che ha un picco negativo dovuto all'induzione e poi una componente positiva) risulta quindi necessario operare in qualche modo per non perdere la parte del segnale positiva (gli ADC in questione registrano solo segnali negativi), per la parte negativa non ci sono problemi.

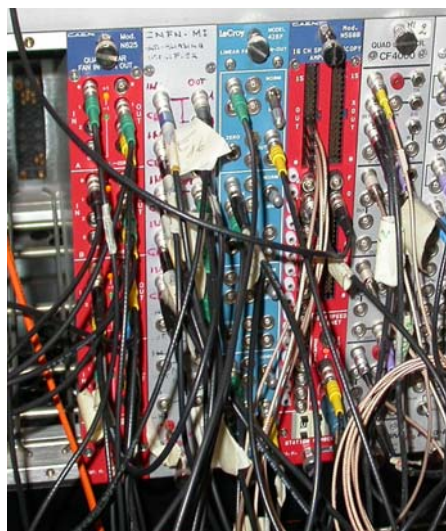
Poiché è stato misurato che il segnale ΔE ha come massimo positivo un valore pari a 100 mV, si è deciso di impostare come offset un valore pari a 300 mV, in modo da far rientrare tutto il segnale ΔE nella dinamica degli ADC.

E' possibile inoltre, agendo su jumpers posti all'interno dei moduli, far sì che i segnali in uscita risultino invertiti di polarità, procedimento necessario quando si vuole utilizzare come evento per la generazione degli impulsi di trigger un segnale positivo come ΔE .

Gli ADC sono stati impostati per acquisire dati all'interno di una finestra temporale pari a 1 μ s; impostando come valore di "*pre-trigger*" un valore di 500 ns è possibile registrare dati cominciando da un tempo antecedente all'evento di trigger, in quanto tutti i canali hanno dei buffer. E' possibile quindi, utilizzando un discriminatore di ampiezza, scegliere se impostare l'impulso del trigger degli ADC sul segnale indotto ΔE o sul segnale ΔE reale: in particolare se si imposta su quest'ultimo sarà certo che l'evento che l'ha generato è reale, se si imposta sul segnale indotto si corre il rischio che l'evento sia originato dal segnale indotto dovuto alle strips adiacenti, anche se con degli accorgimenti è possibile distinguerlo perché avrebbe diversa ampiezza da quello della strip stessa.



**Figura 5.8: Gli ADC inseriti nel crate VME
(bus industriale per moduli a microprocessore)**



**Figura 5.9: Moduli NIM
(altro bus industriale) e loro
interconnessioni**

5.3 Altro turno di misura

Lo stesso sistema di posizionamento micrometrico è stato utilizzato nell'ambito di un altro esperimento per la misura della sezione d'urto di reazione totale per un fascio di Litio ad energia vicino alla barriera di Coulomb. [2]

Le esigenze del caso hanno portato questa volta ad un assemblaggio del sistema composto da piastre ed attuatori completamente diverso.

In questa occasione risultava necessario per il rivelatore essere posto il più lontano possibile dal foro di uscita del cannocchiale, com'è possibile osservare dalle immagini si è arrivati a raggiungere quasi il bordo della camera da vuoto.

La nuova configurazione, con baricentro della struttura di piastre e attuatori esterno alla breadboard, è stata fissata su un braccio motorizzato della camera e orientata perpendicolarmente alla linea di fascio, utilizzando come riferimento le linee radiali tracciate sulla superficie del braccio.

Anche in questa occasione l'attuatore 850-HS è stato utilizzato per comandare i movimenti del collimatore, questa volta costituito da un disco di Tantalio, posto a metà del tragitto di volo delle particelle dirette al rivelatore.

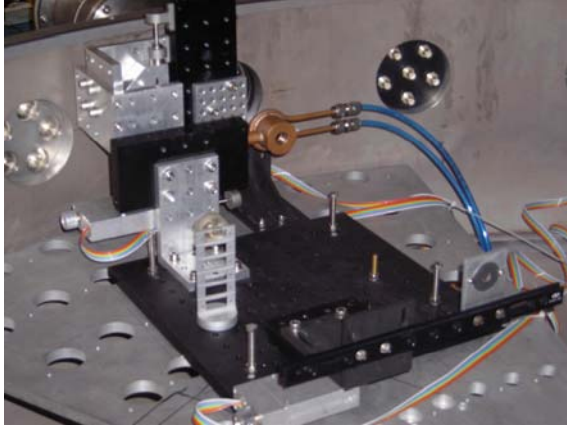


Figura 5.10

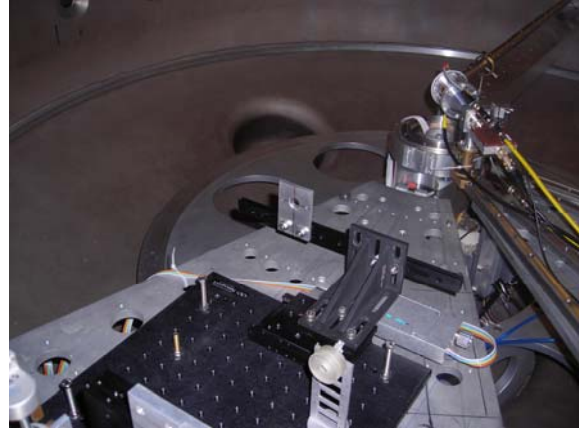


Figura 5.11

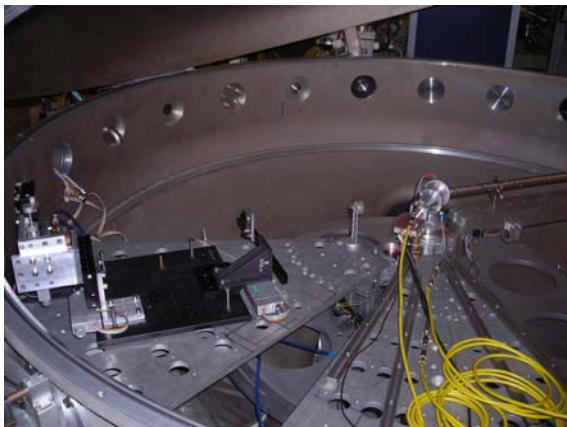


Figura 5.12

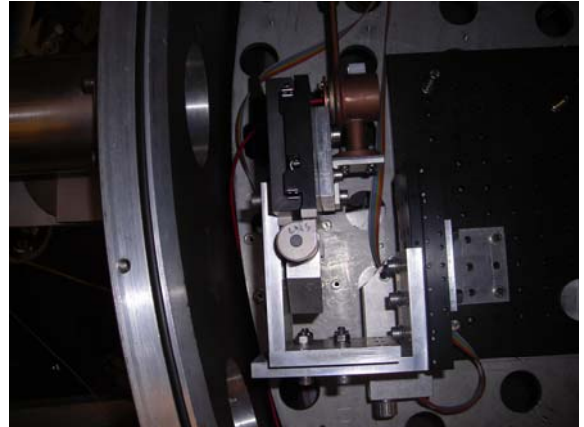


Figura 5.13

6 Perfezionamenti

In questo capitolo si propongono dei perfezionamenti al programma sviluppato, che viene naturale prendere in considerazione per ottenere un prodotto valido come strumento di microposizionamento adatto ad essere utilizzato in altre situazioni diverse da quella originale.

L'ultimo paragrafo è dedicato a considerazioni su come l'uso di un sistema real-time migliorerebbe le prestazioni dell'apparato proposto in questo lavoro di tesi.

6.1 Controllo allineamento e proposta correzioni automatiche

6.1.1 Allineamento di un piano per 3 punti

L'algoritmo di allineamento che verrà esposto è stato progettato per allineare una matrice 5x5 di rivelatori ottici, con un fascio laser opportunamente calibrato e collimato, con dimensione dello spot inferiore all'area di rivelazione pari a un disco di diametro 50 μm ; la distanza tra aree attive adiacenti è pari a 200 μm .

Si supponga inoltre di avere un riscontro strumentale (oscilloscopio, rilevatore di conteggi) per ogni rivelatore della matrice, in modo da poter affermare quando il fascio è centrato all'interno di una area attiva.

E	○	○	○	○	○
D	○	○	○	○	○
C	○	○	○	○	○
B	○	○	○	○	○
A	○	○	○	○	○
	1	2	3	4	5

Figura 6.1: Matrice 5x5.

Chiamando le righe A, B, C, D, E e le colonne 1, 2, 3, 4, 5 è possibile identificare ogni elemento della matrice con la coppia di coordinate riga-colonna, in questo modo il punto in basso a sinistra sarà identificato dalle coordinate A-1 e il punto in alto a destra E-5.

Per allineare un piano nello spazio è necessario identificare almeno tre punti quindi, dato A-1 come punto di partenza, bisogna andare alla ricerca di altri due. I punti che meglio si prestano allo scopo sono A-5 e E-1, rispettivamente il più lontano sulla stessa riga e il più lontano sulla stessa colonna.

1. Supponiamo di aver posizionato il fascio luminoso all'interno dell'area attiva del punto A-1 e di aver preso questa coordinata come riferimento per tutte le altre; vado alla ricerca di A-5 spostandomi 800 μm (200 μm x 4) a destra lungo l'asse X (in realtà il campione si sposta di 800 μm a sinistra perché il fascio è fisso ma per semplicità di comprensione dell'algoritmo prendiamo un sistema di riferimento solidale con il campione)

A-1: (0,0)
move X 800 +

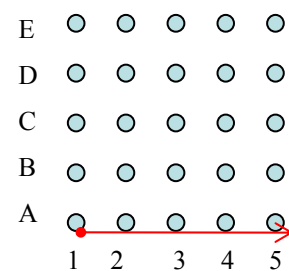


Figura 6.2

2. Giunti al punto di coordinate (0, 800) dovremmo essere all'interno del rivelatore A-5: se i dati strumentali lo confermano allora vuol dire che il campione è perfettamente orizzontale e le coordinate dei punti della stessa riga sono univocamente determinate

Tabella 6-1

A	0, 0	0, 200	0, 400	0, 600	0, 800
	1	2	3	4	5

3. Se i dati mostrano che siamo giunti ad un punto lontano dalla zona attiva di rivelazione allora è necessaria una ricerca a piccoli steps ad arco di compasso (centrato su **A-1** e di ampiezza 800 μm) intorno quella posizione, ad esempio 50 μm in alto e a sinistra calcolando la coordinata corrispondente con il Teorema di Pitagora.

move Y 50 +

move X Δx -

dove

$$\Delta x = 800 - \sqrt{800^2 - 50^2}$$

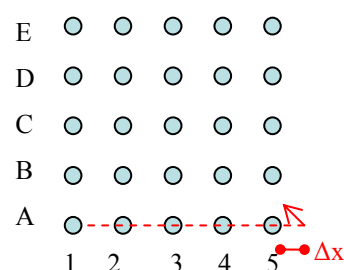


Figura 6.3

Si procede così per iterazioni successive, scorrendo ordinate positive e negative, fin quando si trova il centro della zona attiva.

Trovate le coordinate (**x**, **y**) l'intera riga è determinata univocamente

Tabella 6-2

A	0, 0	x /4, y /4	x /2, y /2	3 x /4, 3 y /4	x , y
	1	2	3	4	5

4. Se l'asse individuato dalla riga A non è perpendicolare alla linea di fascio la traslazione imposta di 800 μm risulterebbe maggiore di quella sufficiente al raggiungimento del punto fissato, è necessario quindi ripartire dal punto (3) a coordinate (0,800), ridurre gradualmente l'ascissa a piccoli intervalli e continuare con la ricerca a compasso.

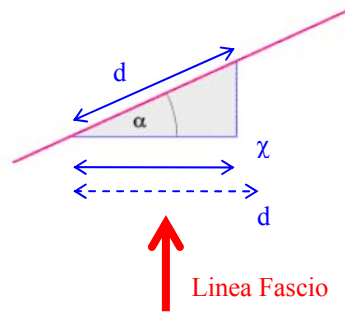


Figura 6.4

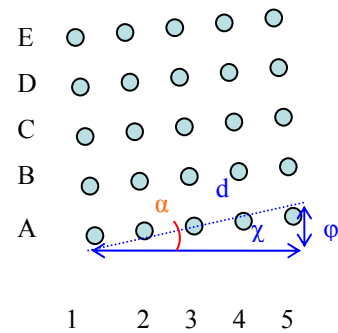


Figura 6.5

Trovato il punto **A-5** di coordinate (χ, φ) è allora possibile apprezzare l'angolo di inclinazione α , perché $\chi = d \cdot \cos \alpha \Rightarrow \alpha = \arccos\left(\frac{\chi}{d}\right)$, correggerlo e poi ripartire dal punto (1).

5. Trovate le coordinate della riga **A** ci si muove alla ricerca delle coordinate del punto **E-5** partendo da **A-1**. Si procede poi lungo la colonna 1 in modo analogo a quanto fatto per i punti (1, 2, 3, 4), lungo la riga **A**, sicchè trovate le coordinate di **E-1**: $(\underline{x}, \underline{y})$ le coordinate dell'intera matrice sono univocamente determinate.

Tabella 6-3

E	$\underline{x}, \underline{y}$	...	...	...	$\underline{x}+\underline{x}, \underline{y}+\underline{y}$
D	$3\underline{x}/4, 3\underline{y}/4$	...	...	...	...
C	$\underline{x}/2, \underline{y}/2$	...	...	...	...
B	$\underline{x}/4, \underline{y}/4$	...	...	...	...
A	0, 0	$\underline{x}/4, \underline{y}/4$	$\underline{x}/2, \underline{y}/2$	$3\underline{x}/4, 3\underline{y}/4$	$\underline{x}, \underline{y}$
	1	2	3	4	5

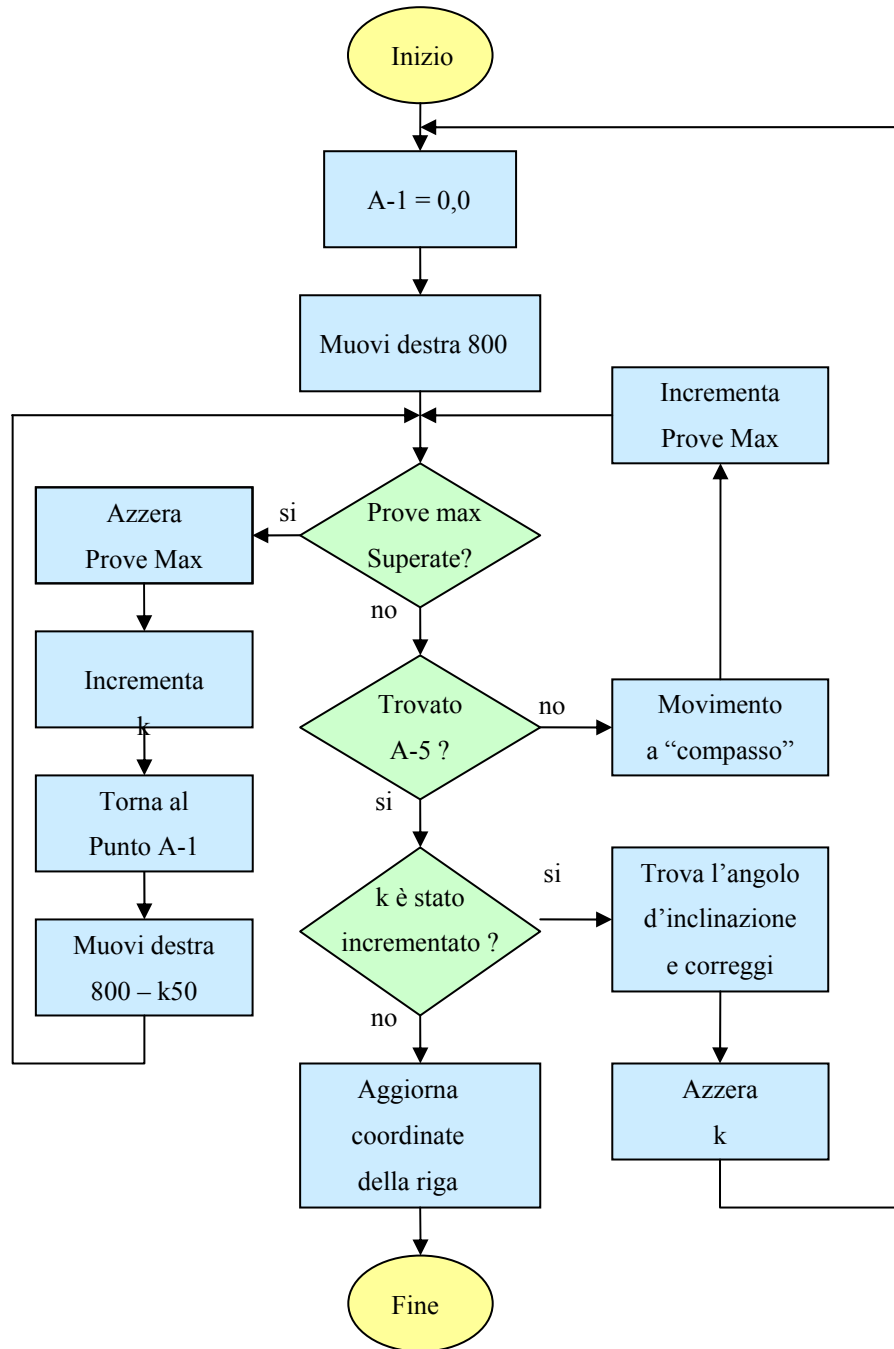


Figura 6.6: La figura mostra lo schema a blocchi semplificato delle procedure per l'allineamento su una riga descritte ai punti (1, 2, 3, 4).

In realtà questo algoritmo non è stato utilizzato, in quanto la rotazione meccanica effettuata con gli attuatori sul piano X-Y, perpendicolare all'asse del fascio Z, si è rivelata molto più precisa di un eventuale correzione meccanica per mezzo dei micrometri della piastra di rotazione "Tip & Rotation", inoltre la condizione di perpendicolarità del piano X-Y (rispetto l'asse Z, richiesta per una corretta acquisizione dei dati) non era restrittiva.

6.1.2 *Raster*

Il programma sviluppato nasce dall'esigenza di comandare movimenti punto-punto per gli attuatori lineari, eventualmente muovendosi su più assi contemporaneamente, ma in ogni caso prevede lo spostamento da una posizione statica, in cui sono state effettuate misure ad un'altra in cui verranno eseguite. Non prevede la possibilità di muovere un rivelatore sotto il fascio durante un turno di acquisizione dati.

Allo stato attuale una necessità del genere imporrebbe all'operatore di inserire il valore desiderato di spostamento, ogni volta che l'attuatore corrispondente ha terminato il movimento precedente. Questa condizione non risulta di certo ottimale in quanto i tempi non regolari di permanenza del fascio in un determinato punto (a causa di accelerazione, decelerazione, ritardo nell'avvio del moto, ecc.) potrebbero portare a valori di misura inconcludenti e perfino danneggiare il dispositivo.

Risulta naturale pensare a evoluzioni del programma che inglobino al loro interno funzioni di questo tipo, permettendo ad esempio di irraggiare il rivelatore lungo tutte le strips, muovendosi per tutta la loro lunghezza, semplicemente impostando un percorso predefinito. Oppure si potrebbe esplorare una singola strip seguendo un andamento tipo onda quadra all'interno dell'area attiva, in modo da poter studiare la tendenza del rivelatore sottoposto ad una situazione dinamica, con il fascio in movimento (sarebbe opportuno sincronizzare i dati acquisiti con le posizioni attuali del sistema di movimentazione).

Tra i vari SubVI's messi a disposizione da Newport Corporation per il controllore ESP6000 sono presenti alcuni piccoli sistemi per il controllo della movimentazione dinamica, ad esempio il SubVI **"Example, raster"** che fornisce un'implementazione molto efficiente per la risoluzione di esigenze del tipo proposto. Permette infatti di impostare un percorso, lungo due assi, composto da una serie di movimenti singoli. Possono essere impostate sia le distanze da percorrere lungo X e Y, sia il numero di volte che verrà ripetuto il movimento, oltre che velocità, accelerazione, decelerazione per gli attuatori coinvolti.

E' da notare che in questa particolare implementazione il movimento lungo un asse non comincia prima che il sistema abbia rilevato l'avvenuto raggiungimento dell'attuatore alla posizione stabilita, ciò è reso possibile dalla chiamata ad un ulteriore SubVI presente

al suo interno, "Motion complete", che inibisce la partenza dello step successivo se l'attuatore in movimento non ha completato il suo moto.

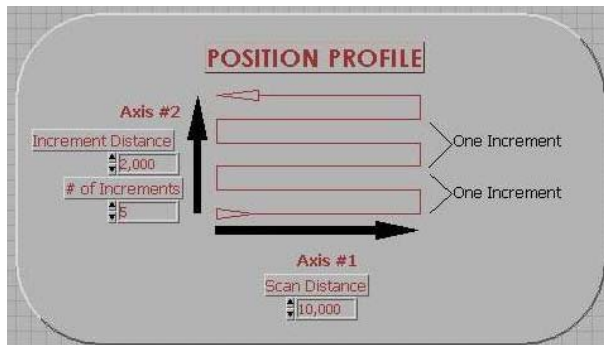


Figura 6.7: Impostazione parametri per il subVI Example, raster

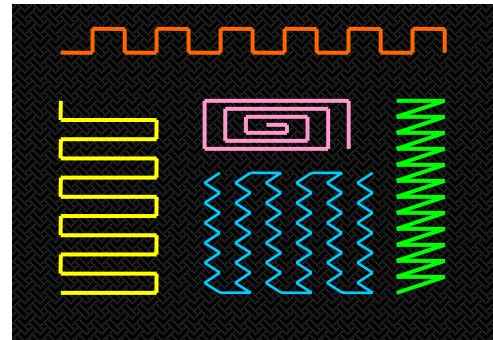


Figura 6.8: Alcuni esempi di moti combinati che sarebbe possibile progettare

Sarebbe possibile, se necessario, implementare una soluzione che permetterebbe di scegliere un percorso predefinito, da modificare per venire incontro alle esigenze del caso.

Il movimento in diagonale sarebbe possibile combinando il moto di due attuatori dello stesso tipo, in questo caso però sarebbe opportuno assicurarsi che le procedure di invio comandi ai drivers del controllore siano nella stessa pagina dei vari loop da implementare, in modo da far inviare il comando nello stesso istante. In realtà una condizione ideale, in cui una rampa di 45° sia perfettamente raggiunta, è obiettivo difficile da raggiungere. Per impostare pendenze diverse si potrebbero impostare valori differenti di velocità, ecc.

6.2 Soluzione Real-Time

In informatica l'espressione *Real-Time Computing (RTC)* si riferisce allo studio di sistemi hardware e software che sono soggetti a "restrizioni di tempo reale", più semplicemente immediate.

Ciò vuol dire che i limiti di tempo a disposizione sono un parametro fondamentale per il tempo di risposta del sistema. Al contrario, in un sistema non real-time non ci sono

limiti di tempo per l'esecuzione, anche se l'elaborazione in un tempo breve è un parametro privilegiato per avere alte prestazioni.

In un sistema real-time l'applicazione (all'interno del contesto) può essere considerata come una "missione cruciale", se non si è completato il lavoro entro il limite di tempo prestabilito allora il sistema ha fallito. Un esempio pratico può essere dato dagli apparati di sicurezza delle automobili, come i dispositivi che comandano airbags e ABS; in questi casi un ritardo nell'esecuzione dei comandi in caso di incidente metterebbe a serio rischio l'incolumità del guidatore e dei passeggeri a bordo.

Nell'uso corrente, avendo a disposizione una potenza di calcolo molto elevata perfino nei personal computers portatili, un sistema abbastanza veloce viene considerato erroneamente come real-time; in realtà ciò non è vero in tutti i casi in quanto il sistema operativo, che definisce la priorità ai processi in esecuzione e gestisce gli accessi alla memoria, generalmente non permette di avere un'interazione diretta a basso livello con l'hardware fisico.

A volte è possibile intervenire direttamente disabilitando processi in esecuzione in background, bloccando interrupts provenienti da periferiche del sistema o innalzando la priorità di esecuzione a quelli per cui si vuole una risposta real-time, ma non sempre è prevedibile se questi accorgimenti adottati produrranno il risultato atteso. [22]

Quindi, per riassumere le informazioni fin qui esposte, si può dire:

- o real-time non sempre vuol dire molto veloce
- o real-time vuol dire assoluta affidabilità
- o i sistemi real-time hanno vincoli temporali che devono essere rispettati per evitare un insuccesso
- o un sistema si dice deterministico se ha la facoltà di completare un compito entro un determinato periodo di tempo

Facendo un confronto tra un sistema operativo *general purpose* ed uno real-time, risulta evidente perché quest'ultimo si dimostra essere più affidabile in un certo tipo di applicazioni.

Un general purpose OS ha dei *tasks* ad alta priorità che possono in qualsiasi momento essere disturbati da tasks di priorità più bassa, così ad esempio l'avvio dello *screen saver* o una qualsiasi *utility* dell'hard disk che viene eseguita in *background* può presentarsi come richiesta al processore proprio nel momento in cui si sta effettuando un'acquisizione dati "*time critical*", e non è detto che in questa condizione la richiesta che sarà eseguita per prima sia necessariamente quella che l'utente ha definito ad alta priorità.

Così anche altre periferiche di sistema come la tastiera, il mouse o la stampante possono bloccare l'esecuzione del programma perché inviano interrupts che il processore deve comunque gestire.

In un real-time OS, invece, esiste uno *scheduler* che assegna all'applicazione una priorità alta e saremo sicuri che tale applicazione sarà eseguita senza interruzioni, inoltre generalmente sono composti da postazioni stand-alone, senza periferiche di input come mouse o tastiera. [21]

Per venire incontro a queste esigenze di esecuzione entro un tempo definito, necessarie ad esempio in impianti industriali e nucleari, sono presenti varie soluzioni che comprendono sia sistemi operativi real-time sia hardware a se stante real-time che dialoga con il chi lo controlla ma esegue quanto dovuto in modo indipendente.

Un esempio di hardware dedicato può essere dato da schede per il controllo di processi real-time o dai *Programmable Logic Controllers (PLC)*.

7 Appendice

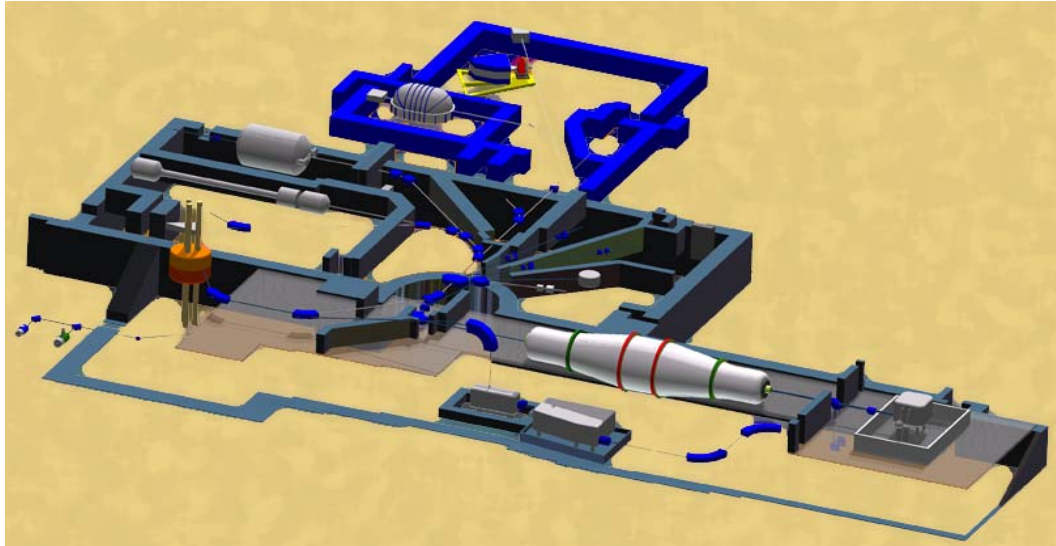
7.1 INFN-LNS: I Laboratori Nazionali del Sud

L'*Istituto Nazionale di Fisica Nucleare*, in sigla *INFN*, è l'istituto italiano che promuove, coordina ed effettua la ricerca scientifica nel campo della fisica subnucleare, nucleare e astroparticellare, nonché lo sviluppo tecnologico necessario alle attività in tali settori.

Opera in stretta connessione con l'Università e nell'ambito della collaborazione e del confronto internazionale (dalla seconda metà degli anni '50 iniziò la partecipazione dell'INFN alle attività di ricerca del *CERN*, il *Centro europeo di ricerche nucleari di Ginevra*).

L'attività dell'INFN si basa su due tipi di strutture di ricerca complementari: le Sezioni e i Laboratori Nazionali. Le 19 Sezioni hanno sede in dipartimenti universitari e realizzano il collegamento diretto tra l'Istituto e le Università.

I quattro Laboratori, con sedi a Catania (*LNS*), Frascati (*LNF*), Legnaro (*LNL*) e Gran Sasso (*LNGS*), ospitano grandi apparecchiature e infrastrutture messe a disposizione della comunità scientifica nazionale e internazionale. [25]



**Figura 7.1: Layout 3D delle apparecchiature
presenti all'interno dei LNS**

7.2 L'acceleratore di particelle "Tandem" dei LNS

Un acceleratore di particelle è una macchina il cui scopo è quello di produrre fasci di ioni o particelle subatomiche con elevata energia cinetica. I metodi per accelerare particelle sono basati sull'uso di campi elettrici e magnetici, di cui i primi forniscono energia alle particelle accelerandole ed i secondi servono a curvarne la traiettoria (ad esempio negli acceleratori circolari: ciclotrone e sincrotrone).

Presso i Laboratori Nazionali del Sud sono presenti ed operativi un acceleratore lineare detto "*Tandem*" (per il principio di funzionamento che verrà esposto) ed un ciclotrone superconduttore.



**Figura 7.2: Vista frontale
dell'acceleratore lineare "Tandem"**



**Figura 7.3: Il ciclotrone
superconduttore**

Il Tandem è così chiamato perché l'accelerazione a cui sono soggette le particelle avviene in due stadi. Al suo interno è presente un lungo tubo composto da anelli di materiale vetroso, per isolare elettrostaticamente le particelle dall'esterno, e un gas isolante sotto pressione.

L'accelerazione avviene bombardando atomi neutri (si possono accelerare tutti i tipi di atomi, ad eccezione dei gas nobili) con elettroni, in modo da ottenere ioni negativi. Questi ioni si trovano quindi in un punto a basso potenziale e vengono attratti per via elettrostatica verso il centro dell'acceleratore ad alto potenziale positivo (15 MV).

Le particelle, giunte al terminale positivo, attraversano un sottile strato di Carbonio (detto *stripper*) che ne "strappa" elettroni rendendole questa volta ioni positivi. Si innesca quindi un'altra accelerazione dovuta alla repulsione elettrostatica tra ioni e alto potenziale positivi e le particelle sono spinte verso il terminale a basso potenziale



Figura 7.4: Interno del Tandem

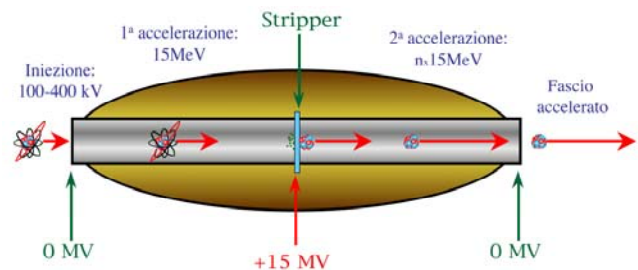


Figura 7.5: Principio di funzionamento

L'energia del fascio così prodotto risulta pari a $(Q+1)eV_0$, dove V_0 è il potenziale del terminale centrale e Q è la carica ionica dopo il passaggio dallo stripper.

Per $V_0 = 15$ MV, Q può raggiungere Z (numero atomico dell'elemento) per atomi leggeri, mentre per quelli pesanti è limitato a un valore pari a $10 - 12$.

La massima energia è quindi producibile con fasci di Carbonio (105 MeV) e Ossigeno (135 MeV) fino a 200 MeV con atomi pesanti. La massima corrente dipende dal tipo di ioni prodotti e dagli stati di carica richiesti, valori tipici vanno da 10 a 200 nA. [26]

8 Riferimenti bibliografici

1. Imaging monolithic silicon detector telescope; INFN-LNS ACTIVITY REPORT 2006
F. Amorini, T. Basso, C. Boiano, G. Cardella, L. Cosentino, A. Di Pietro, G. Fallica, P. Figuera, P. Finocchiario, A. Pappalardo, F. Rizzo, N. Randazzo, V. Russo, S. Scirè, C. Scirè, V. Sipala, A. Trifirò, M. Trimarchi, G. Valvo
2. Total Reaction cross section measurements of ${}^7,8\text{Li}+{}^{28}\text{Si}$ around the Coulomb barrier: preliminary results; INFN-LNS ACTIVITY REPORT 2006
A. Musumarra, F. De Luca, A. Di Pietro, P. Figuera, P. Finocchiario, M. Fisichella, M. Lattuada, T.J. Mertzimekis, A. Pakou, G. Randisi, G. Scalia, S. Scirè, C. Scirè, V. Scuderi
3. Development of an Irradiation Set-up for Deep Lithography with Protons – Pedro Vynck (2004) Vrije Universiteit Brussel
4. G.Cardella et al. Nucl. Instr. And Meth A 378 (1996) 262
5. S.Tudisco et al. Nucl. Instr. And Meth A 426 (1999) 436
6. A.Musumarra et al. Nucl. Instr. And Meth A 409 (1998) 414
7. F.Amorini et al. Nucl. Instr. And Meth. A 550, 248, (2005) and references therein
8. Nuove tecniche di microdiagnostica con scintillatori per fasci ionici di bassa intensità - Alfio Pappalardo (2002)
9. http://it.wikipedia.org/wiki/Motore_in_corrente_continua
10. http://it.wikipedia.org/wiki/Motore_passo-passo
11. <http://www.vincenzov.net/tutorial/passopasso/stepper.htm>
12. <http://documents.exfo.com/Products/LifeSciences/LSS-8000-Inchworm.pdf>
13. http://en.wikipedia.org/wiki/Inchworm_motor
14. Introduction to Motion Control – The Newport Resource 2006/2007; Newport Corporation
15. 850G Series Linear Actuator User Manual; Newport Corporation
16. ESP6000 – UNIDRIVE6000 User's Manual; Newport Corporation

17. DC Interface Box Operator's Manual; Newport Corporation
18. http://it.wikipedia.org/wiki/Controllori_PID
19. ESP6000 – UNIDRIVE6000 User's Manual; Newport Corporation
20. Guida all'uso di LabVIEW – National Instruments
21. Introduction to LabVIEW Real-Time
22. http://en.wikipedia.org/wiki/Real-time_computing
23. Misure e misurazioni, Termini e definizioni fondamentali – UNI 4546
24. Guida all'espressione dell'incertezza nella misurazione – UNI CEI 9
25. <http://it.wikipedia.org/wiki/INFN>
26. <http://www.lns.infn.it/accelerator/tandem.html>