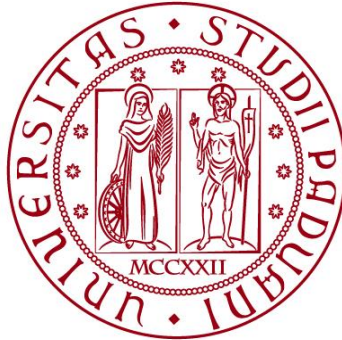


**UNIVERSITÀ DEGLI STUDI DI PADOVA**

SCUOLA DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea Magistrale in Ingegneria Elettronica



**TESI DI LAUREA MAGISTRALE**

**OTTIMIZZAZIONE DELLE PRESTAZIONI DI MISURA DI SISTEMA DI CARATTERIZZAZIONE  
ELETTRICA DI CELLE FOTOVOLTAICHE BASATO SU TECNOLOGIA LED**

Relatore: Prof. MATTEO MENEGHINI

Correlatore: Ing. ALESSANDRO VOLTAN

Laureando: DAMIANO SELVINO MARANGONI

Matricola 1150895

16 Dicembre 2019

ANNO ACCADEMICO 2018-2019



## SOMMARIO

Lo scopo di questo lavoro di Tesi è quello di presentare l'analisi e ottimizzazione delle performance di misura di un simulatore solare basato su tecnologia LED per caratterizzazione di celle fotovoltaiche effettuato presso la sede di Treviso (Baccini Cell Systems) dell'azienda Applied Materials Italia.

La Tesi presenta la seguente struttura:

Capitolo 1: Il primo capitolo contestualizza la sede presso cui ho svolto il mio lavoro di tirocinio e fornisce una rapida descrizione delle funzioni di un generico simulatore solare.

Capitolo 2: Il secondo capitolo è interamente dedicato alla presentazione delle norme previste per la caratterizzazione I-V delle celle solari e ricerca bibliografica: sono state analizzate sia le procedure per definire le principali figure di merito di una cella fotovoltaica, sia le procedure di elaborazione dati quali correzione in temperatura e irradianza. Particolare attenzione viene posta nei confronti delle normative per la misura di uniformità spaziale, tema centrale di questa tesi.

Capitolo 3: L'elaborato prosegue (terzo capitolo) con una descrizione generale della componentistica hardware principale: lampada, sistema di contattazione, carico elettronico e cella di riferimento. Per quel che riguarda la descrizione software, viene riportato il sistema di misura per le principali figure di merito e parametri elettrici di una cella fotovoltaica.

Capitolo 4: Il quarto capitolo è dedicato all'analisi da me svolta relativa al tema dell'uniformità spaziale di irradianza. Il punto di partenza è stato definire matematicamente una procedura con l'intento di abbassare il parametro di non-uniformità (NU) su di un'area posta sul piano della cella sotto misura. Successivamente viene presentato il mio lavoro di simulazione di diversi algoritmi nati a tale scopo in ambiente Matlab.

Capitolo 5: Il quinto capitolo si occupa della descrizione dell'implementazione software di una procedura ben definita di correzione dello spettro della lampada del simulatore solare. Viene inoltre evidenziato il legame tra uniformità spaziale e uniformità spettrale, il lavoro di debug e test di algoritmi per la calibrazione dello spettro della lampada.

Capitolo 6: vengono effettuate le considerazioni finali sulle effettive migliorie apportate al sistema.

In calce alla tesi vengono riportate due appendici contenenti le sezioni di codice da me implementate in ambiente TwinCat 3. Tale ambiente costituisce un linguaggio di programmazione a basso livello spesso utilizzato nell'industria grazie alla sua buona capacità di interazione con PLC.

# INDICE

CAPITOLO 1: INTRODUZIONE .....	1
1.1 BCS (Baccini Cell Systems) .....	1
1.2 Funzioni di un simulatore solare .....	1
CAPITOLO 2: STATO DELL'ARTE .....	7
2.1 Introduzione .....	7
2.2 Norme internazionali per celle monofacciali al silicio .....	7
2.2.1 CEI EN 60904-1: misure delle caratteristiche I-V .....	7
2.2.2 IEC 60904-9 Ed.2 .....	9
2.2.3 IEC 60891: procedure di correzione in temperatura e irradianza delle caratteristiche I-V.....	14
CAPITOLO 3: PANORAMICA GLOBALE DEL SISTEMA .....	19
3.1 Introduzione .....	19
3.2 Panoramica generale.....	20
3.3 Analisi dell'hardware.....	24
3.3.1 La lampada a LED.....	24
3.3.2 Il carico elettronico.....	26
3.3.3 La cella di riferimento .....	28
3.4 Analisi software del sistema di misura .....	29
3.4.1 Calibrazione del sistema .....	29
3.4.2 Acquisizione delle caratteristiche I-V e calcolo parametri elettrici.....	30
CAPITOLO 4: PROCEDURA DI CORREZIONE AUTOMATICA DELL'UNIFORMITA' SPAZIALE DELLA LAMPADA .....	35
4.1 Introduzione .....	35
4.2 Setup sperimentale e dati raccolti con la cella di riferimento .....	37
4.3 Approccio matematico e simulazioni in ambiente MATLAB .....	43
4.3.1 Minimizzazione del parametro di non-uniformità .....	46
4.3.2 Vettore target di uniformità .....	48
4.4 Confronto tra gli algoritmi basato sui dati simulati .....	51
4.5 Implementazione in ambiente TwinCAT .....	55
4.5.1 Rappresentazione della macchina a stati e codice TwinCAT .....	56
4.5.2 L'interfaccia grafica per l'utente .....	58
4.6 Test e dati raccolti .....	60
CAPITOLO 5: PROCEDURA DI CALIBRAZIONE SPETTRALE DELLA LAMPADA .....	63
5.1 Introduzione .....	63

5.2 Setup sperimentale e dati raccolti .....	63
5.2.1 Ripetibilità della lettura dei fotodiodi .....	65
5.2.2 Relazione lineare tra valori digitali e fattori di aggiustamento .....	67
5.3 Approccio matematico al problema.....	68
5.4 Implementazione in ambiente TwinCAT .....	71
5.4.1 L'algoritmo di correzione .....	71
5.4.2 Rappresentazione della macchina a stati .....	73
5.5 Test e dati raccolti .....	76
CAPITOLO 6: CONCLUSIONI.....	80
APPENDICE A: LISTATO DELLA FUNZIONE DI AUTOCALIBRAZIONE DELL'UNIFORMITA' SPAZIALE .....	83
APPENDICE B: LISTATO DELLA FUNZIONE DI CALIBRAZIONE SPETTRALE .....	94
INDICE DELLE FIGURE.....	107
INDICE DELLE TABELLE .....	109
RIFERIMENTI BIBLIOGRAFICI .....	111

# **CAPITOLO 1: INTRODUZIONE**

## **1.1 BCS (Baccini Cell Systems)**

La divisione di Applied Materials con sede a Treviso vede la sua nascita nel 2008 a seguito della cessione dell'azienda trevigiana Baccini ad opera dell'omonimo titolare Gisulfo Baccini.

L'azienda italiana nasce nel 1967 e diventa nel corso degli anni leader nel settore della produzione di macchine per la stampa serigrafica di celle fotovoltaiche in silicio cristallino, arrivando a coprirne il 70% della produzione mondiale.

In seguito alla cessione dell'azienda alla multinazionale californiana leader del mercato delle nanotecnologie Applied Materials di Santa Clara (nata anch'essa nel 1967), nasce un laboratorio di ricerca e sviluppo all'interno dell'azienda con lo scopo di studiare soluzioni efficaci ed innovative nel campo della stampa su celle solari in silicio.

Nel 2019 viene costituito un team di metrologia dentro al quale si colloca la mia esperienza di tirocinio presso AppliedMaterials.

In particolare, il progetto al quale ho preso parte prende il nome di "Botticelli" ed è costituito da un simulatore solare a lampada LED usualmente installato a valle della linea di stampa.

## **1.2 Funzioni di un simulatore solare**

La funzione principale di un simulatore solare è quello di fornire i dati per classificare le celle fotovoltaiche alla fine del ciclo di produzione secondo le principali figure di merito.

La classificazione avviene principalmente discriminando il valore di efficienza di una cella, tuttavia è possibile classificare un particolare set di wafers anche in base ad altre caratteristiche: colore, resistenza di griglia o Fill Factor.

La funzione del simulatore solare è dunque quella di effettuare misure elettriche sulla cella sotto test, estraendone i parametri elettrici (ad esempio resistenza serie/parallelo, tensione di circuito aperto, corrente di corto circuito) e le principali figure di merito (ad esempio efficienza e Fill Factor).

Oltre a ciò un simulatore solare può adempiere ad altre funzioni: per esempio, tramite l'integrazione della tecnica dell'elettroluminescenza, è in grado di iniettare corrente nella cella e di catturare l'immagine nello spettro dell'infrarosso, individuando così eventuali difetti o rotture della cella.

La struttura di un simulatore solare viene presa in considerazione dettagliatamente nel terzo capitolo di questo elaborato, questo paragrafo delinea solamente in generale lo schema di principio di funzionamento di un simulatore.

A valle della linea di serigrafia, un sistema di trasporto trasporta una cella fotovoltaica all'interno del simulatore: essa viene posizionata al di sotto di una lampada ed opportunamente contattata con un carico elettronico (tramite un sistema a quattro punti).

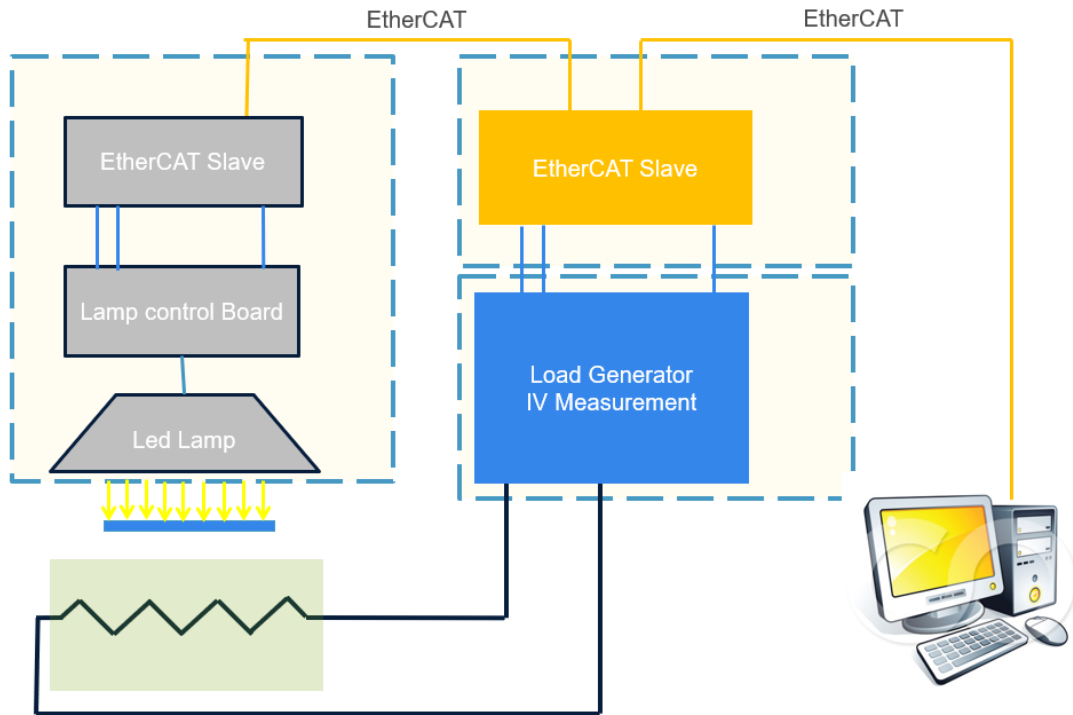
La lampada ha la funzione di illuminare la cella sotto misura emulando quanto più possibile lo spettro solare in condizioni standard. Esistono diversi tipi di standard che definiscono lo spettro solare. Quello a cui viene fatto riferimento in questo elaborato è l'AM1.5 (si rimanda al capitolo successivo per la trattazione delle norme).

Al carico elettronico a tensione costante è affidato il compito di eseguire uno sweep in tensione e di misurare la corrente che fluisce dalla cella a causa dell'effetto fotovoltaico: in tale maniera viene ricostruita la curva tensione-corrente che caratterizza la cella sotto test.

Opportuni sensori rilevano la temperatura della cella durante la fase di misura: tale informazione risulta fondamentale per il calcolo dei parametri elettrici.

Infine, una cella di riferimento è posta in prossimità e alla stessa altezza del wafer sotto test: tramite un'opportuna trasduzione di segnale (da tensione a potenza) è possibile risalire al valore preciso di irradianza con la quale si è svolta la misura. In Figura 1.1 è rappresentata graficamente la struttura del simulatore.





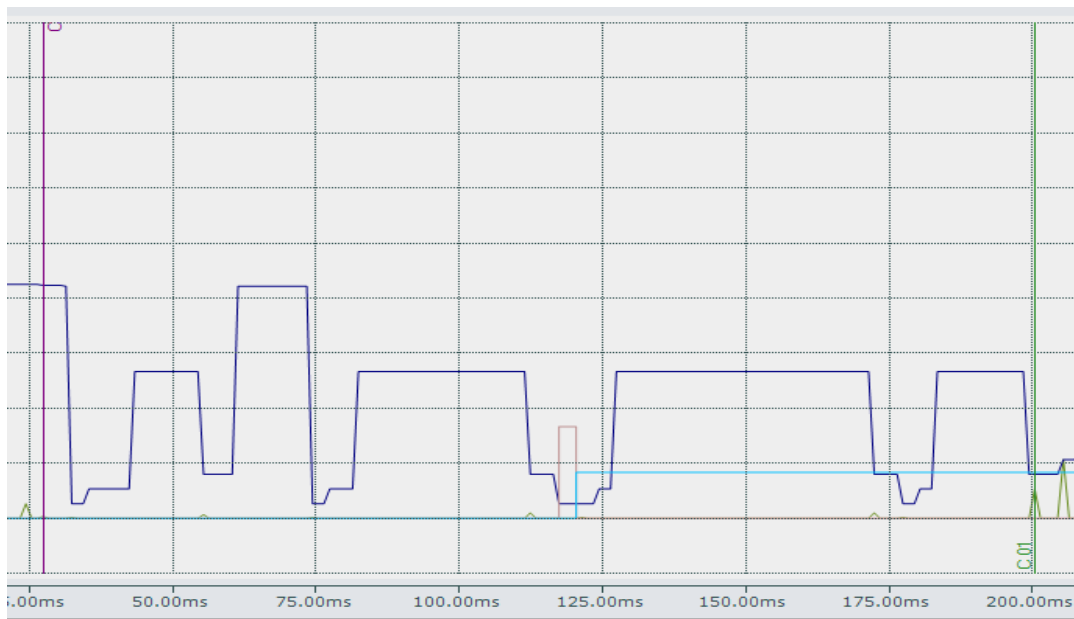
*Figura 1.1: struttura generale del simulatore solare AMAT*

A partire dalle informazioni relative a temperatura, curva I-V e irradianza, il software del simulatore solare elabora i dati e fornisce come output un file contenente le principali figure di merito e i parametri elettrici della cella.

Un parametro di primaria importanza è il throughput del simulatore solare o, equivalentemente, il suo tempo ciclo. Con quest'ultimo termine si indica l'intervallo temporale entro il quale una cella solare arriva al sistema di contattazione ed esce dopo essere stata misurata. Con un'industria che poggia le sue fondamenta sulla velocità di produzione, risulta essenziale essere in grado di effettuare la misura nel minor tempo possibile.

Nel caso specifico del simulatore solare AMAT il tempo di misura effettivo può arrivare a contrarsi fino a 180ms circa, con un tempo ciclo di circa 1s (Figura 1.2).

Da questa breve analisi risulta chiaro il punto cardine della funzione di un simulatore solare: esso deve compiere la misura nel più breve tempo possibile, ottimizzando al meglio la sequenza di operazioni necessarie al compimento della misura.



*Figura 1.2: scope dei segnali digitali in fase di acquisizione*

Un altro aspetto fondamentale che caratterizza le prestazioni di un simulatore solare è quello relativo all'accuratezza e alla ripetibilità delle misure: vengono definite come specifiche di progetto una deviazione standard e una soglia di accuratezza da applicare ad una particolare figura di merito o ad un particolare parametro elettrico.

Per comprendere ciò basti pensare al fatto che ogni giorno in linea di produzione vengono testate decine di migliaia di wafers con un singolo simulatore: non sarebbe accettabile che celle dalle stesse prestazioni e provenienti dallo stesso processo venissero categorizzate diversamente una dall'altra a causa di una scarsa ripetibilità del sistema di misura.

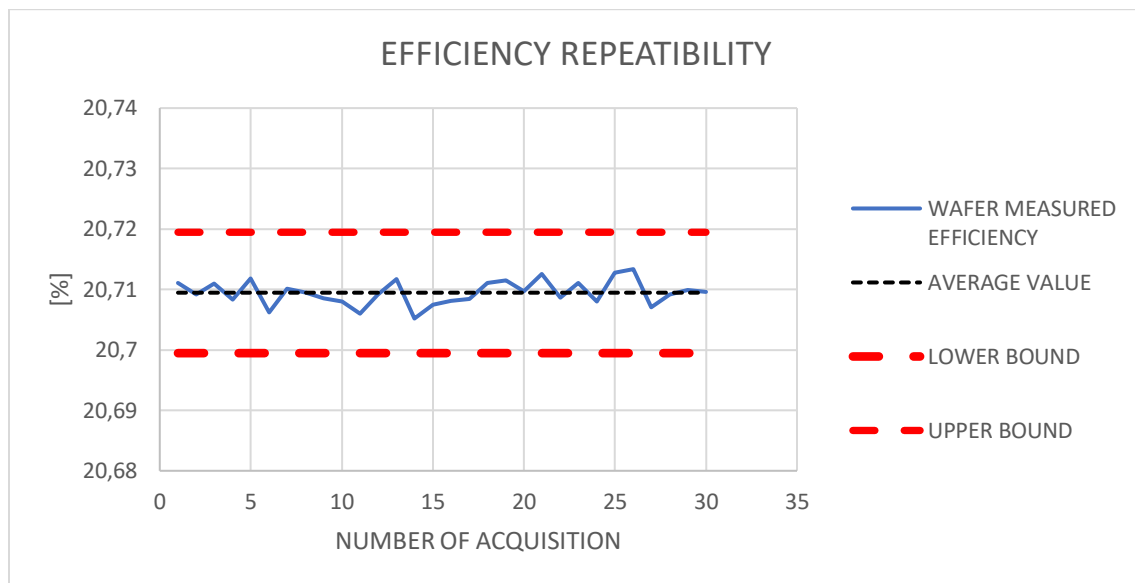


Figura 1.3: misura di efficienza di una cella in silicio cristallino

Inoltre, sarebbe allo stesso modo non accettabile una scarsa accuratezza sulla misura. Si prenda come esempio l'efficienza: celle ad alta efficienza hanno generalmente un costo superiore a quelle caratterizzate da un'efficienza minore. Se il sistema di misura non fosse caratterizzato da una buona accuratezza e, per esempio, classificasse celle ad alta efficienza come celle a bassa efficienza, il produttore sarebbe afflitto da un'ingente danno economico in fase di vendita dei wafers.

Oltre a ciò, se le celle in questione sono destinate alla costituzione di un modulo solare, è necessario che all'interno del modulo siano presenti celle con caratteristiche quanto più possibile simili tra loro. In caso contrario verrebbero alla luce problematiche relative all'efficienza e alla funzionalità del modulo stesso.

In Figura 1.3 si riporta un esempio di trenta misure consecutive sulla stessa cella fotovoltaica in condizioni statiche (ovvero mantenendo la cella contattata al carico elettronico lungo l'arco delle acquisizioni).

Le linee rosse tratteggiate rappresentano un range di 0.02% che rappresenta un'usuale specifica di progetto definita dai produttori di celle: è necessario predisporre il sistema di misura in maniera tale da accertarsi che la ripetibilità non scenda al di sotto di questo limite.

Risulta opportuno specificare, tuttavia, che la mia esperienza di tirocinio si è inserita in un contesto di ricerca e sviluppo: ciò ha sicuramente consentito un grado di libertà maggiore rispetto ad una realtà di produzione di massa.

Quando necessario è stato possibile lasciare in secondo piano aspetti quali la velocità di acquisizione, throughput, ripetibilità e accuratezza, focalizzandosi maggiormente su altre funzionalità della macchina, con lo scopo di migliorarne se possibile le prestazioni. In particolar modo il mio lavoro si è concentrato sulla qualità dell'emissione luminosa e dello spettro della lampada del simulatore solare.

## **CAPITOLO 2: STATO DELL'ARTE**

### **2.1 Introduzione**

La Commissione elettrotecnica internazionale (IEC) è l'organizzazione leader a livello mondiale che prepara e pubblica standard internazionali per tutte le tecnologie elettriche, elettroniche e correlate.

In particolare, in questo lavoro di tesi, vengono considerate quelle normative che definiscono univocamente le procedure relative alle misure elettriche di una cella fotovoltaica.

Tali norme risultano necessarie per più fattori. Prima di tutto, risulta necessario avere a disposizione un'interpretazione univoca dei parametri elettrici e delle figure di merito che caratterizzano una cella fotovoltaica.

In secondo luogo, risulta necessario definire delle condizioni standard entro le quali i test devono essere effettuati. Per rendere ripetibili e confrontabili i dati relativi alle varie misure si devono fissare entro certi limiti i fattori fisici che influiscono sulle prestazioni di un dispositivo a semiconduttore. Esempi pratici sono: temperatura ambientale, potenza con la quale si illumina una cella, forma dello spettro della sorgente luminosa, umidità dell'aria...

Ci si rende conto facilmente che perde di significato il confronto tra due o più misure effettuate in condizioni differenti.

Una delle finalità degli standard internazionali è proprio questa: ricondursi a delle condizioni di misura ben definite per permettere di rendere le misure su celle fotovoltaiche ripetibili e confrontabili tramite criteri oggettivi.

### **2.2 Norme internazionali per celle monofacciali al silicio**

Nel seguito si prendereanno in esame alcune normative riguardanti le misure in luce simulata.

In particolar modo ci si sofferma maggiormente sull'acquisizione delle caratteristiche I-V delle celle, sulla correzione in temperatura e irradianza e sull'uniformità spaziale di irradianza.

#### 2.2.1 CEI EN 60904-1: misure delle caratteristiche I-V

La norma italiana CEI EN 60904-1 dell'anno 2008 “descrive le procedure per la misurazione delle caratteristiche corrente-tensione di dispositivi fotovoltaici effettuata in luce solare naturale o simulata. Tali procedure sono applicabili a singole celle solari, a sottoassiemi di celle solari o a moduli fotovoltaici [1].”

In questa tesi viene presa in considerazione la misura di tensione e corrente in luce simulata per una singola cella solare.

La prima parte della normativa fornisce le prescrizioni generali di misura. Esse possono essere sintetizzate dai seguenti punti:

- Le misure di irraggiamento devono essere effettuate con un dispositivo opportunamente tarato, oppure con un piranometro. Le risposte spettrali del dispositivo di riferimento e del campione di prova devono essere identiche: in caso contrario deve essere applicata una correzione per eliminare l'errore di adattamento tra i due dispositivi.
- La corrente di cortocircuito del dispositivo di riferimento deve variare linearmente con l'irraggiamento
- La temperatura del dispositivo di riferimento e del campione deve essere misurata utilizzando strumenti con una accuratezza di  $\pm 1$  °C con una ripetibilità pari a  $\pm 0,5$  °C.
- La superficie attiva del campione deve giacere sullo stesso piano dell'area attiva del dispositivo di riferimento. La tolleranza d'errore è data da  $2^\circ$ .
- Tensioni e correnti devono essere misurate utilizzando strumenti con una accuratezza pari a  $\pm 0,2$  % della tensione a circuito aperto e della corrente di cortocircuito. La norma specifica anche le caratteristiche che devono assumere i cavi per la misura di tensione e corrente. In particolare modo si specifica che: “[...] per il collegamento si useranno cavi indipendenti, differenti dai terminali del campione, con l'accortezza di tenerli il più corto possibile [1].”
- La corrente di cortocircuito deve essere misurata alla tensione nulla, utilizzando un dispositivo a polarizzazione variabile (preferibilmente elettronico) per compensare la caduta di tensione ai capi della resistenza serie esterna. La norma fornisce una seconda possibilità per la misura della corrente di cortocircuito, specificando che: “[...] in alternativa, la corrente di cortocircuito può essere estrapolata dalla caratteristica corrente-tensione [1].”

Nel caso del simulatore solare presentato in questa tesi, il dispositivo di riferimento utilizzato per le misure di irraggiamento è costituito da una cella monofacciale al silicio, di area pari a 2cm x 2cm. Tale cella è incapsulata in un contenitore dal quale è possibile prelevare due segnali distinti. Il primo, è un segnale di tensione prelevato tramite una resistenza di sensing di valore pari a 1 m $\Omega$ : tale segnale, opportunamente convertito in potenza, serve per la misura di irraggiamento. Il secondo segnale acquisito, invece, è un segnale di temperatura prelevato tramite una PT100 incapsulata nell'involucro della cella stessa.

Per quel che riguarda l'acquisizione di tensione e corrente sul dispositivo campione, viene utilizzato un carico elettronico a polarizzazione variabile. Tale dispositivo è collegato al sistema di contattazione della cella campione tramite un set di cavi della lunghezza di circa un metro. Viene utilizzato un numero di cavi pari alla quantità di busbar presenti nella cella (per entrambe le polarità di tensione e corrente).

Oltre alle indicazioni per l'acquisizione di tensione e corrente, la norma CEI EN 60904-1 fornisce indicazioni sul posizionamento del dispositivo di riferimento e sull'uniformità di irraggiamento. Per sistemi a luce impulsata viene specificato che : “L'uniformità della distribuzione della luce sull'area di prova deve essere nota e periodicamente verificata. L'accuratezza della misura deve essere verificata regolarmente con misure successive nelle stesse condizioni di prova [1].”

Per dispositivi di riferimento più piccoli del dispositivo campione (situazione verificata in questo sistema), è necessario eseguire più misure di irraggiamento in posizioni differenti. Successivamente, è possibile posizionare il dispositivo in una zona tale per cui la misura di irraggiamento fornisca come risultato la media delle prove effettuate precedentemente.

#### 2.2.2 IEC 60904-9 Ed.2

La normativa internazionale 60904-9 redatta dalla commissione elettrotecnica internazionale nel 2007 fornisce le definizioni e le procedure di misura per un simulatore solare.

La norma premette che un simulatore solare può essere utilizzato per due funzioni distinte: la caratterizzazione I-V oppure le misure di esposizione all'irraggiamento. I simulatori solari vengono definiti come oggetti che: “[... ] usually consist of three main components: (1) light source(s) and associated power supply; (2) any optics and filters required to modify the output beam to meet the classification requirements and (3) the necessary controls to operate the simulator [2].”

Oltre a ciò risulta vitale classificare i simulatori in base al segnale luminoso da essi utilizzato in un ciclo di misura. La sorgente luminosa può lavorare in steady-state, singolo impulso, impulsi multipli. Quest'ultima condizione di lavoro corrisponde al caso del simulatore utilizzato in Applied Materials, poiché durante un ciclo di misura vengono utilizzati due impulsi luminosi.

Particolare attenzione viene posta nei confronti del tema dell'uniformità spaziale, di cui si riportano le caratteristiche propedeutiche per il lavoro svolto nel quarto capitolo di questa tesi.

Vengono definiti da normativa i seguenti concetti:

- Piano di test: piano destinato a contenere il dispositivo sotto test al livello di irradianza di riferimento
- Area designata: regione del piano di prova valutata per l'uniformità. La normativa permette di definire diverse forme geometriche per l'area designata (rettangolare o circolare)
- Irradianza effettiva: media di tutti i campioni di irradianza acquisiti durante una prova

Sfruttando le definizioni appena enunciate, si definisce un parametro di non-uniformità dato dall'equazione:

$$\text{Non - uniformity (\%)} = \frac{(\text{Max irradianza} - \text{Min irradianza})}{(\text{Max irradianza} + \text{Min irradianza})} \times 100 \quad (2.1)$$

I valori di irradianza minimo e massimo sono quelli rilevati dal dispositivo nell'area di misura designata.

Tale parametro fornisce un'indicazione sullo scostamento puntuale dei valori di irradianza dal valor medio, in relazione all'area di misura scelta. La condizione ottimale (e ideale) prevede un valore percentuale di non-uniformità nullo.

Basandosi su questo parametro è possibile classificare un simulatore solare in tre diverse categorie, come riportato in Tabella 2.1.

Classificazione	Non-uniformità di irradianza
A	2%
B	5%
C	10%

*Tabella 2.1: classificazione di un simulatore solare in base al parametro di non-uniformità di irradianza*

Va precisato che, da normativa, un simulatore può essere classificato anche in base ad altri parametri. Uno di questi è, ad esempio, l'instabilità temporale di irradianza che in questo lavoro di tesi non verrà preso in considerazione a causa della sua scarsa rilevanza.

La normativa IEC 60904-9 fornisce anche delle linee guida per procedere alle misure di irradianza, specificando tuttavia che: “[...] no generalization can be made and non-uniformity is to be evaluated for each system [2].”



I valori di irradianza per un simulatore dipendono, infatti, dalle riflessioni che si verificano all'interno dell'apparato stesso.

Le diverse fasi per la misura del parametro di non-uniformità si articolano nei seguenti punti:

1. Come dispositivo di misura è consigliata una cella al silicio cristallino incapsulata. Deve essere garantita la linearità tra irraggiamento e corrente di corto circuito, inoltre la risposta temporale deve essere conforme alle caratteristiche del simulatore.
2. È necessario designare un'area di misura: un valore opportuno può essere di 160 cm x 240 cm.
3. Successivamente, si divida la superficie in zone da 2cm x 2cm. Alternativamente, si può suddividere l'area designata in almeno 64 posizioni.
4. Le misure di irradianza effettuate sulle zone ottenute al punto precedente devono coprire il 100% dell'area designata.
5. Per un simulatore a luce impulsata risulta utile un secondo dispositivo da posizionare all'esterno dell'area designata. Tramite tale dispositivo si deve monitorare il valore di irradianza e utilizzare tali dati per calibrare il dispositivo da spostare nelle varie posizioni.
6. Si determina il parametro di non-uniformità tramite l'equazione (2.1), con lo scopo di classificare il dispositivo seguendo le indicazioni della Tabella 2.1.

Come descritto nel Capitolo 4, la metodologia di misura utilizzata per i test di irradianza in questa tesi si discosta dalle indicazioni fornite da normativa. In particolar modo si utilizza solamente un dispositivo per le misure (la cella di riferimento) e un numero ridotto di posizioni di misura. Le motivazioni a tali scelte verranno fornite nel quarto capitolo.

La norma internazionale IEC 60904-9 fornisce importanti dettagli anche per quel che riguarda le componenti spettrali emesse dalla fonte luminosa di un simulatore.

È necessario premettere che la normativa si riferisce allo standard AM 1.5. In Figura 2.1 si riporta la definizione di "Air Mass" fornita dal sito PV education [3].

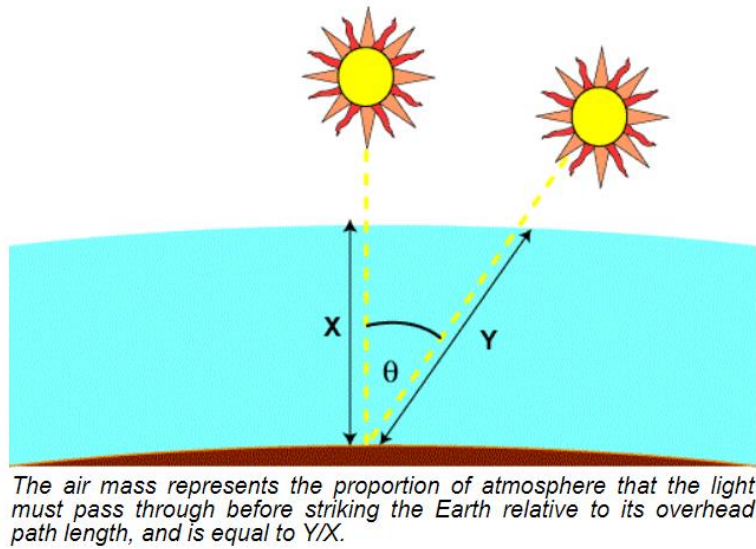


Figura 2.1: definizione di AM

Nel caso di AM 1.5, dunque, il rapporto tra il percorso effettuato dalla luce solare in atmosfera e il suo percorso aereo vale 1.5. A tal valore corrisponde un angolo  $\theta$  di circa  $48.2^\circ$ .

In Figura 2.2 si riporta lo spettro solare in condizioni di AM 1.5 secondo il sito PV education [4]

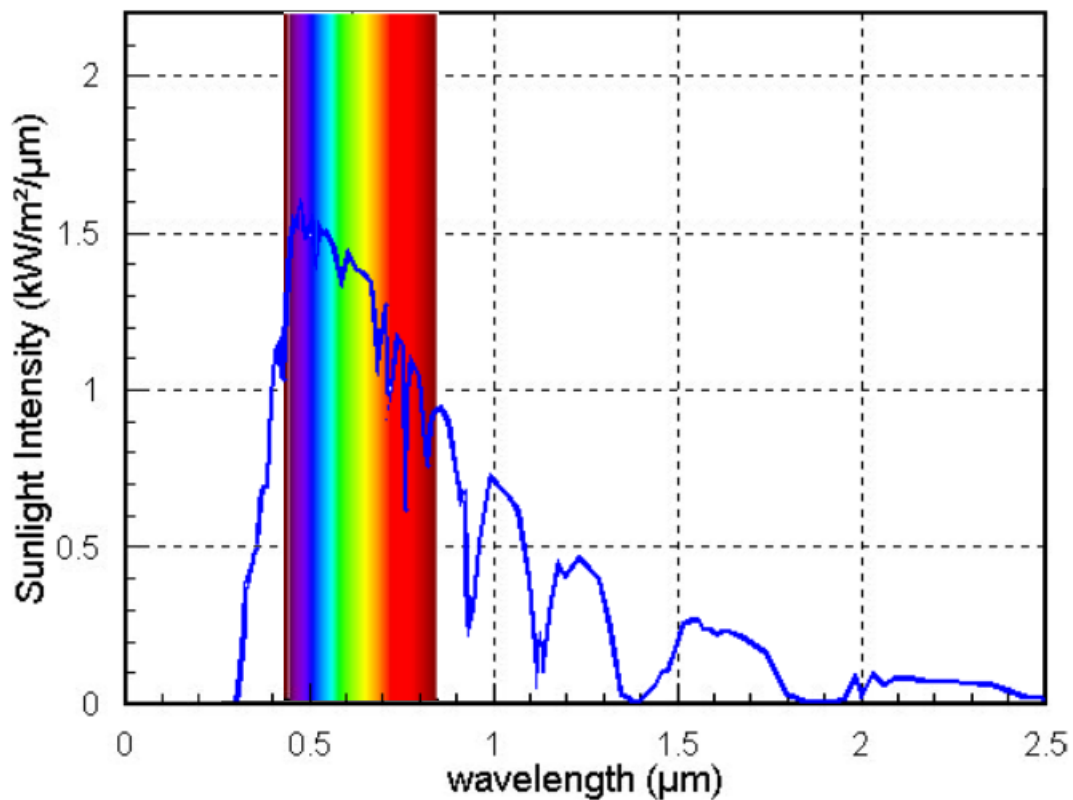


Figura 2.2: intensità della luce solare per lo standard AM 1.5

In Figura 2.2 si è messo in evidenza la porzione di spettro nel visibile.

La normativa IEC 60904-9 definisce il range spettrale di un simulatore solare il range [400nm-1100nm]. Inoltre, la normativa divide il suddetto range in 6 bande separate, ognuna delle quali contribuisce con una certa percentuale all'irradianza totale.

Le sei bande vengono riportate in Tabella 2.2.

	Wavelength range nm	Percentage of total irradiance in the wavelength range 400 nm – 1 100 nm
1	400 – 500	18,4 %
2	500 – 600	19,9 %
3	600 – 700	18,4 %
4	700 – 800	14,9 %
5	800 – 900	12,5 %
6	900 – 1 100	15,9 %

*Tabella 2.2: spettro solare in AM 1.5 diviso in bande*

Si definisce “spectral match” la deviazione percentuale di un simulatore dai valori riportati in Tabella 2.2: “spectral match of a solar simulator is defined by the deviation from AM 1,5 reference spectral irradiance as laid down in Table 1 [2].”

Così come per il tema legato all'uniformità, anche per il matching spettrale è possibile classificare in diverse categorie un simulatore solare. La Tabella 2.3 riassume la classificazione.

Classificazione	Spectral match di tutti gli intervalli riportati in tabella 2.2
A	0.75 – 1.25
B	0.6 – 1.4
C	0.4 – 2.0

*Tabella 2.3: classificazione di un simulatore solare in base al matching spettrale*

La normativa fornisce alcune indicazioni su come effettuare una misura di matching spettrale, ai fini della classificazione del simulatore. Possono essere utilizzati i seguenti strumenti:

- uno spettroradiometro comprendente un monocromatore a griglia e un rivelatore discreto;
- uno spettrometro CCD o array di fotodiodi (CCD = dispositivo accoppiato a carica);
- un gruppo rivelatore multiplo con filtri passa banda;
- un singolo rivelatore con filtri passa banda multipli.

Sfruttando gli strumenti appena elencati, è necessario procedere alla misura delle percentuali che contribuiscono all'irradianza totale del simulatore. Dopo aver fatto ciò, si calcoleranno i rapporti tra i valori misurati per il simulatore e i valori dello spettro solare riportati in Tabella 2.2: a questo punto è possibile procedere alla classificazione del simulatore stesso.

### 2.2.3 IEC 60891: procedure di correzione in temperatura e irradianza delle caratteristiche I-V

La normativa IEC 60891 fornisce le procedure necessarie alla correzione dei campioni di tensione e corrente in temperatura e irradianza.

Tali correzioni risultano necessarie poiché le caratteristiche I-V di una cella fotovoltaica dipendono fortemente dalle condizioni di temperatura e irraggiamento con i quali viene effettuata una misura. In particolar modo, come riportato in IEC 60904-1, la corrente di cortocircuito presenta una relazione lineare con il valore di irradianza utilizzato per illuminare la cella. La temperatura, invece, entro certi limiti, presenta una relazione lineare con la tensione di circuito aperto.

Le correzioni in temperatura e irradianza hanno lo scopo di fornire come output le caratteristiche I-V equivalenti alle condizioni standard: 25° e 1000W/m<sup>2</sup>.

Di fondamentale importanza è il metodo di acquisizione dell'irradianza: essa va quantificata tramite la misura della corrente di cortocircuito del dispositivo di riferimento. Tale corrente si ottiene dai campioni di tensione e corrente del dispositivo di riferimento, opportunamente corretti in temperatura.

In sintesi, la procedura di correzione si sviluppa nei seguenti punti:

1. acquisizione della corrente del dispositivo di riferimento e conversione in irradianza;
2. correzione dei campioni di irradianza in temperatura;
3. calcolo valore effettivo di irradianza come media aritmetica dei campioni;
4. acquisizione delle caratteristiche I-V del campione sotto test;
5. correzione delle curve I-V in temperatura e irradianza.

Il punto numero 2 dell'elenco sopracitato è da implementare tramite la relazione:

$$G = \frac{1000 \text{ Wm}^{-2} \cdot I_{RC}}{I_{RC,STC}} \cdot [1 - \alpha_{RC} \cdot (T_{RC} - 25^{\circ}\text{C})] \quad (2.2)$$

I simboli utilizzati della relazione (2.2) assumono i seguenti significati:

- 1000 Wm<sup>-2</sup> e 25°C fanno riferimento alle condizioni standard di irraggiamento e temperatura;
- I<sub>RC</sub> è la corrente di cortocircuito del dispositivo di riferimento;

- $I_{RC,STC}$  è una costante di calibrazione del dispositivo di riferimento, riferita alla corrente di cortocircuito;
- $T_{RC}$  è la temperatura del dispositivo di riferimento;
- $\alpha_{RC}$  è una costante con dimensioni  $[1/^\circ C]$  e rappresenta il coefficiente di temperatura del dispositivo di riferimento;
- $G$  è l'irradianza corretta in temperatura.

Il punto 5, invece, presenta diverse possibili implementazioni. In particolar modo la normativa IEC 60891 fornisce tre diverse relazioni per correggere le curve I-V in irradianza e temperatura. Due di queste relazioni sfruttano come input alcuni parametri correttivi e una sola curva I-V. La terza relazione, invece, non necessita di parametri correttivi, ma sfrutta più curve I-V relative alla stessa cella campione.

In questa tesi vengono prese in considerazione le metodologie che sfruttano solamente una caratteristica di tensione e corrente, in coerenza con il metodo di calcolo del simulatore solare di Applied Materials.

## METODO 1

Si applica alle coppie di punti di tensione e corrente acquisite il seguente sistema di due equazioni:

$$\left\{ \begin{array}{l} I_2 = I_1 + I_{SC} \cdot \left( \frac{G_2}{G_1} - 1 \right) + \alpha \cdot (T_2 - T_1) \end{array} \right. \quad (2.3)$$

$$\left\{ \begin{array}{l} V_2 = V_1 - R_S \cdot (I_2 - I_1) - \kappa \cdot I_2 \cdot (T_2 - T_1) + \beta \cdot (T_2 - T_1) \end{array} \right. \quad (2.4)$$

dove:

- $I_1$  e  $V_1$  sono i punti della caratteristica I-V acquisiti alla temperatura  $T_1$  e irradianza  $G_1$  ;
- $I_2$  e  $V_2$  sono i punti della caratteristica I-V corretta alla temperatura  $T_2$  e irradianza  $G_2$  . Alle condizioni standard  $T_2$  e  $G_2$  valgono rispettivamente  $25^\circ C$  e  $1000W/m^2$ ;
- $I_{SC}$  è la corrente di cortocircuito misurata nelle condizioni  $(T_1, G_1)$  ;
- $\alpha$  e  $\beta$  rappresentano i coefficienti termici di correzione della corrente e della tensione;
- $R_S$  è la resistenza serie del campione sotto misura, riferita alle condizioni  $(T_1, G_1)$  ;
- $\kappa$  è un fattore di correzione della curva.

La norma specifica che l'equazione (2.3): "is only applicable for I-V curves measured at irradiances which are constant during the acquisition of the entire I-V curve [5]." Ciò significa che nell'arco dell'acquisizione della caratteristica I-V l'irradianza deve essere quanto meno stabile, e tale valore di irradianza è rappresentato dalla media dei campioni di irradianza.

I coefficienti termici nelle equazioni (2.3) e (2.4) devono essere calcolati preventivamente: si riportano a fine capitolo le indicazioni fornite da normativa.

## METODO 2

Il secondo metodo di correzione si basa sul modello a un diodo di un dispositivo fotovoltaico.

Si applica alle coppie di punti di tensione e corrente acquisite il seguente sistema di due equazioni:

$$\begin{cases} I_2 = I_1 \cdot (1 + \alpha_{rel} \cdot (T_2 - T_1)) \cdot \left(\frac{G_2}{G_1}\right) & (2.5) \\ V_2 = V_1 + V_{OC1} \cdot \left(\beta_{rel} \cdot (T_2 - T_1) + a \cdot \ln\left(\frac{G_2}{G_1}\right)\right) - R'_S \cdot (I_2 - I_1) - \kappa' \cdot I_2 \cdot (T_2 - T_1) & (2.6) \end{cases}$$

dove:

- $I_1$  e  $V_1$  sono i punti della caratteristica I-V acquisiti alla temperatura  $T_1$  e irradianza  $G_1$  ;
- $I_2$  e  $V_2$  sono i punti della caratteristica I-V corretta alla temperatura  $T_2$  e irradianza  $G_2$  . Alle condizioni standard  $T_2$  e  $G_2$  valgono rispettivamente  $25^\circ\text{C}$  e  $1000\text{W/m}^2$ ;
- $V_{OC1}$  è la tensione di circuito aperto misurata nelle condizioni  $(T_1, G_1)$  ;
- $\beta_{rel}$  e  $\alpha_{rel}$  rappresentano i coefficienti termici relativi di correzione alle condizioni standard di  $1000\text{W/m}^2$ . Questi due coefficienti sono legati alla tensione di circuito aperto e alla corrente di cortocircuito in STC (standard conditions);
- $R'_S$  è la resistenza serie del campione sotto misura, riferita alle condizioni  $(T_1, G_1)$  ;
- $\kappa'$  è un coefficiente termico di correzione riferito al parametro  $R'_S$  ;
- $a$  è il fattore di correzione dell'irradianza per la tensione a circuito aperto che è collegato alla tensione termica del diodo D della giunzione pn e al numero di celle nS collegate in serie nel modulo.

Il fatto che il sistema comprendente le equazioni (2.5) e (2.6) derivi dal modello a un diodo è palesato dalla dipendenza dal logaritmo del rapporto tra l'irradianza di misura e quella in condizioni standard che presenta la tensione  $V_2$ .

Tra le note, la normativa aggiunge che un tipico valore del parametro  $a$  è 0.06.

In Figura 2.3 si riporta un esempio di applicazione dei due metodi ad una caratteristica I-V di una cella fotovoltaica in silicio monocristallino, acquisita alla temperatura di 20° con un livello di irradianza di 1369 W/m<sup>2</sup>. Sono stati utilizzati valori verosimili per i coefficienti termici,  $I_{sc}=9.93$  A,  $V_{oc}=0.675$  V,  $R_s=0.008$  Ω. Come suggerito da normativa, invece,  $a=0.06$ . I valori standard per temperatura e irradianza sono di 25° e 1000 W/m<sup>2</sup>.

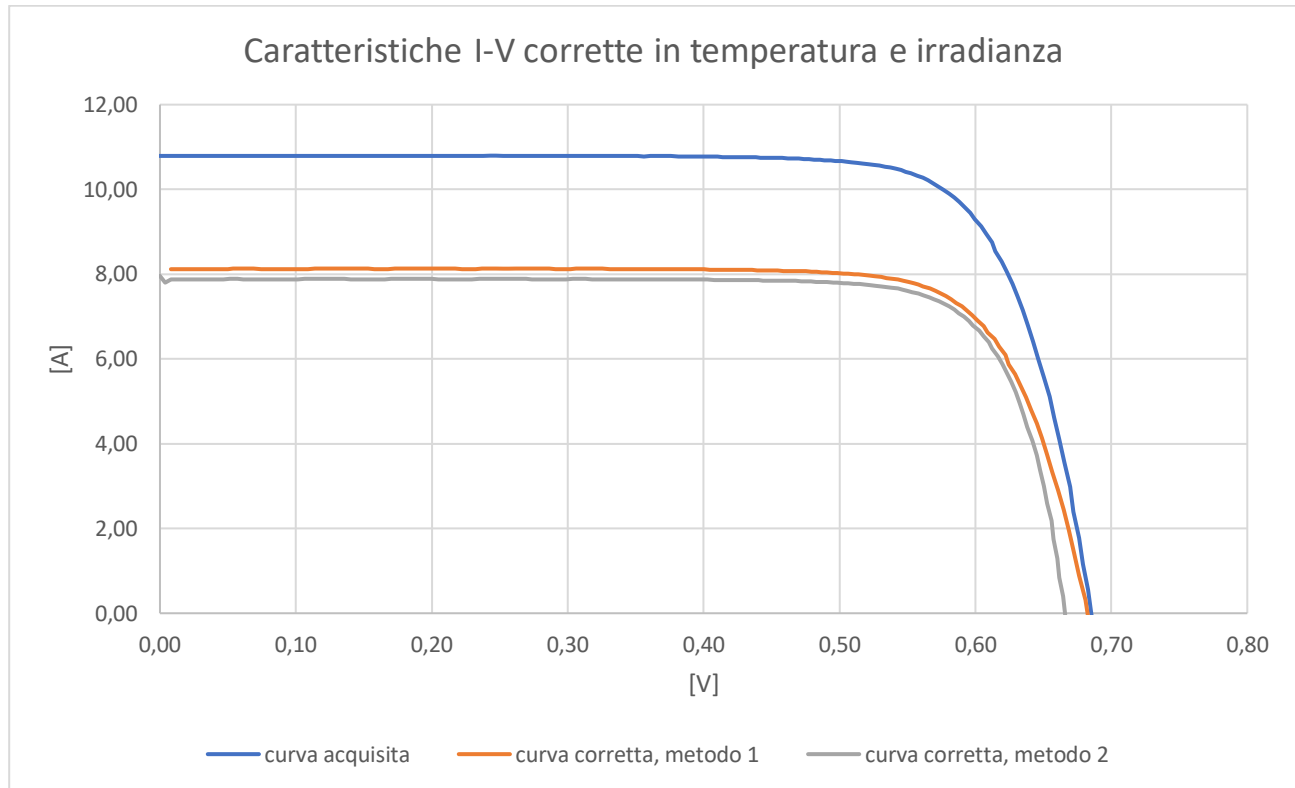


Figura 2.3: esempio di correzione in irradianza e temperatura di una caratteristica I-V

Dalle curve graficate si può notare che entrambe le caratteristiche corrette presentano un valore minore di corrente di cortocircuito: ciò è dovuto al fatto che la misura è stata effettuata con un livello di irradianza di 1,369 volte il valore standard.

Per ciò che riguarda la temperatura, invece, per entrambe le curve si ottiene un valore di tensione di circuito aperto inferiore a quello acquisito. Ciò avviene perchè la temperatura durante la misura è di 5° inferiore al valore standard. Si nota che il secondo metodo di correzione tende a traslare più rigidamente la curva rispetto il metodo 1. Ciò è confermato dalla pendenza della curva nell'intervallo [0,60V-0,70V]: l'inverso di tale pendenza da una stima qualitativa della resistenza serie. Tale resistenza

risulta essere quasi immutata con il secondo metodo di correzione, mentre il primo metodo tende a modificarne il valore (aumentandolo in questo esempio).

### CALCOLO DEI COEFFICIENTI TERMICI

In questa viene preso in considerazione il calcolo solamente dei coefficienti termici  $\alpha$  e  $\beta$ . Il calcolo dei fattori di correzione delle curve non viene riportato. Tale scelta è giustificata dal fatto che questi sono gli unici parametri che tendono a variare considerevolmente al variare della temperatura e del tipo di cella sotto test.

La norma prescrive di:

1. Stabilizzare la temperatura del dispositivo in maniera tale che la variazione massima rientri in  $\pm 2$  °C;
2. Impostare tramite dispositivo di riferimento il livello di irradianza desiderato (o in alternativa nelle condizioni standard);
3. Registrare i valori relativi a  $I_{sc}$  e  $V_{oc}$  e la temperatura alla quale è stata effettuata la misura;
4. Ripetere a step di 5°C, per un range di almeno 30°C i punti 1,2 e 3;
5. Plottare l'insieme di dati relativi a  $V_{oc}$  e  $I_{sc}$  come funzioni della temperatura e costruire una curva con il metodo dei minimi quadrati;
6. Il coefficiente angolare delle curve così create rappresentano  $\alpha$  per la curva corrente/temperatura e  $\beta$  per la curva tensione/temperatura.

Per ottenere i coefficienti termici relativi utilizzati nel secondo metodo di correzione, si applichino le relazioni seguenti:

$$\alpha_{rel} = \frac{\alpha}{I_{sc}} \quad \text{con } I_{sc} \text{ misurata a } 25^{\circ}\text{C} \quad (2.7)$$

$$\beta_{rel} = \frac{\beta}{V_{oc}} \quad \text{con } V_{oc} \text{ misurata a } 25^{\circ}\text{C} \quad (2.8)$$



## CAPITOLO 3: PANORAMICA GLOBALE DEL SISTEMA

### 3.1 Introduzione

La bontà di una determinata misura elettrica su cella fotovoltaica è strettamente correlata al setup hardware e software che caratterizza il simulatore solare.

Durante la mia permanenza in AppliedMaterials sono state numerose le modifiche e gli aggiornamenti apportati all'intero sistema.

Risulta fondamentale ricordare che ogni tipologia di cella richiede un proprio setup particolare per poter essere caratterizzata correttamente.

Come esempio guida si consideri una cella monofacciale e una bifacciale: è altamente improbabile che lo stesso sistema di illuminazione possa fornire risultati validi e precisi in entrambe le situazioni. In un caso la cella assorbe potenza solo dalla superficie illuminata direttamente dalla sorgente preposta; nel secondo caso si verifica un assorbimento anche sulla superficie non direttamente illuminata. È chiaro che risulta relativamente semplice caratterizzare lo spettro della luce che incide sulla superficie superiore della cella, mentre non è possibile far ciò per la porzione di luce riflessa che illumina la superficie sottostante.

Tale esempio ha lo scopo di motivare la presenza di questo capitolo: per procedere ad una corretta analisi dei dati di misura è necessario considerare le condizioni sia hardware (tipo di illuminazione, sistema di contattazione...) che software (tempi di misura, elaborazione dei campioni acquisiti...) che caratterizzano un particolare test.

Di seguito verrà presentato uno schema generale del simulatore solare sia dal punto di vista dei componenti sia dal punto di vista algoritmico.

Successivamente si procederà ad un'analisi più dettagliata del sistema, evidenziando tramite dati sperimentali le migliorie in termini di ripetibilità e accuratezza raggiunte tramite gli aggiornamenti apportati al sistema.

### 3.2 Panoramica generale

Il simulatore solare di AppliedMaterials è inserito in un modulo metallico a monte del sistema di classificazione delle celle solari.

Per fornire una panoramica del sistema, si elencano le operazioni in successione eseguite per la classificazione di una singola cella solare:

1. La cella viene prelevata da un apposito contenitore mediante un braccio meccanico ad aria compressa. Lo stesso braccio la posiziona sopra una coppia di nastri trasportatori (in assenza di ciò, le celle possono essere caricate a mano);
2. La cella viene fatta scorrere lungo il modulo, fino ad arrivare verso un centratore meccanico: tale dispositivo ha il compito di posizionare la cella in maniera coerente al sistema di contattazione. Le piastre meccaniche del centratore muovono la cella in tre direzioni diverse, fino a giungere al posizionamento corretto;
3. La cella solare viene trasportata verso il sistema di contattazione: una coppia di barre con incastonati dei pin metallici ricoperti d'oro;
4. Le barre di contattazione si chiudono, mettendo in contatto le estremità dei pin con i busbar della cella;
5. Un pirometro misura la temperatura della cella e trasferisce il relativo segnale al software;
6. Viene avviata la fase di misura: si effettuano due sweep di tensione (tramite un carico elettronico attivo) e vengono registrate le relative caratteristiche I-V;
7. La lampada LED si attiva dopo aver ricevuto un segnale dal sistema software del simulatore: a questo punto essa si accende prima ad un livello di irradianza di  $1000\text{W/m}^2$  (misurata da cella di riferimento) e successivamente a  $500\text{W/m}^2$ . Vengono registrate le curve I-V ai due livelli di irradianza, sfruttando sempre il carico elettronico;
8. Una volta terminata la fase di misura I-V, le barre di contattazione si aprono e rilasciano il campione sotto test;
9. I nastri trasportatori si riattivano e conducono la cella verso l'uscita del modulo al fine di essere classificata.

In Figura 3.1 si riporta una foto dell'intero modulo.



*Figura 3.1: modulo contenente il simulatore solare di AppliedMaterials*

Da Figura 3.1 si notano due pc a bordo macchina: quello di sinistra è utilizzato per il controllo dell'automazione del sistema di trasporto delle celle. Il pc situato a destra, invece, è utilizzato per gestire il sistema di misura tramite un'interfaccia grafica. Quest'ultimo elaboratore riceve e scambia tutti i segnali digitali atti alla caratterizzazione della cella: temperatura, irradianza, coppie di punti I-V e segnali di comunicazione con il sistema. La presenza di pannelli di plexiglass di color nero opaco è dovuta a questioni di sicurezza: essi bloccano la luce proveniente dalla lampada per proteggere l'occhio umano.

In Figura 3.2 viene riportato il sistema dopo aver rimosso i pannelli protettivi.

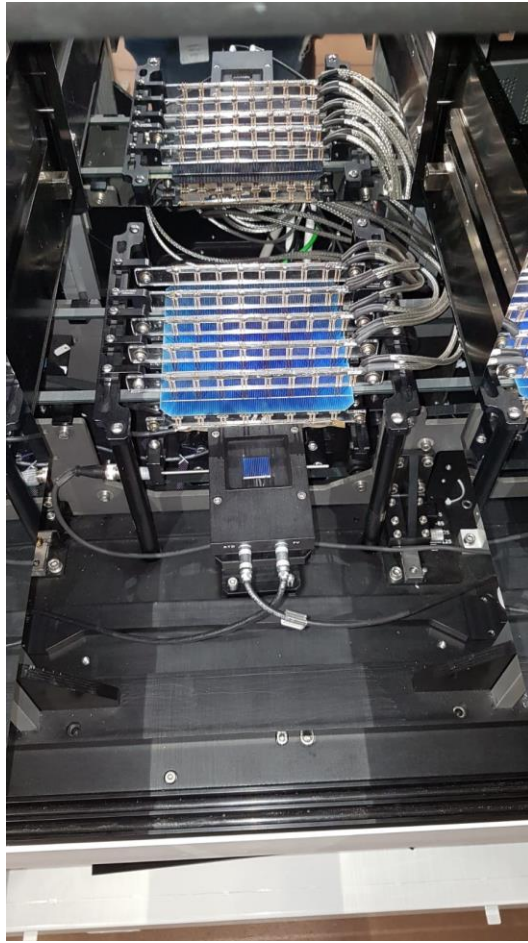


*Figura 3.2: modulo contenente il simulatore solare di AppliedMaterials senza pannelli protettivi*

Nell'immagine si distingue la coppia di nastri trasportatori che attraversa l'intero modulo. Oltre a ciò si nota la presenza del riflettore sorretto da quattro profili metallici. All'apice del sistema riflettente è posta la lampada a LED utilizzata per le misure.

Ciò che si vede all'interno del sistema riflettente è riportato, invece, in Figura 3.3.

Nella foto si osserva la presenza della cella di riferimento ai bordi dell'area di misura. Oltre a ciò, si nota la presenza delle cinque barre di contattazione superiori: ogni barra conta otto coppie di pin metallici utilizzati per lo sweep di tensione e il prelievo di corrente dalla cella. Ogni barra è collegata mediante un cavo schermato al carico elettronico (Figura 3.4).



*Figura 3.3: cella di riferimento e sistema di contattazione*



*Figura 3.4: carico elettronico collegato al sistema di contattazione*

### 3.3 Analisi dell'hardware

#### 3.3.1 La lampada a LED

Il simulatore solare di AppliedMaterials sfrutta una lampada a LED per le misure delle caratteristiche di tensione e corrente delle celle solari.

Tale lampada è costituita da una matrice di LED raggruppati nel seguente sistema:

- L'intera matrice è suddivisa in 10 moduli adiacenti;
- Ogni modulo è suddiviso in 2 zone;
- In ogni modulo è presente un fotodiode, posizionato al centro delle due zone;
- In ogni modulo sono presenti due termistori utilizzati per la rilevazione della temperatura dei substrati;
- L'intera matrice è costituita da 17 catene diverse di LED: ogni catena è adibita alla generazione di una lunghezza d'onda diversa. La sovrapposizione delle righe spettrali delle catene fornisce uno spettro simile a quello della luce solare, in condizioni di AM1.5 (come descritto in Capitolo 2).

In Tabella 3.1 si riporta la numerazione delle catene in corrispondenza alla lunghezza d'onda prodotta.

Catena	lunghezza d'onda (nm)
1	380
2	405
3	425
4	Bianco
5	Bianco
6	500
7	660
8	700
9	700
10	720
11	760
12	800
13	830
14	870
15	925
16	980
17	1050

Tabella 3.1: composizione dello spettro della lampada

In Figura 3.5, invece, si riporta la distribuzione delle catene di LED, dei fotodiode e dei termistori.

CONNECTORS	13	15	15	9	9	1	8	4a	7a	9	28	30	30	24	24	16	23	19a	22a	24
	1	8	8	10	5	3	2	7a	7a	14	16	23	23	25	20	18	17	22a	22a	29
	7a	12	⊗	6	4a	9	11	⊗	13	15	22a	27	⊗	21	19a	24	26	⊗	28	30
	7a	2	11	13	15	14	12	10	6	15	22a	17	26	28	30	29	27	25	21	30
	4a	3	9	8	PD	7a	4a	8	5	7a	19a	18	24	23	PD	22a	19a	23	20	22a
	15	5	14	1	PD	7b	9	Th	3	7b	30	20	29	16	PD	22b	24	Th	18	22b
	15	10	6	13	7b	15	12	2	11	1	30	25	21	28	22b	30	27	17	26	16
	8	4b	⊗	2	7b	15	10	⊗	9	8	23	19b	⊗	17	22b	30	25	⊗	24	23
	15	14	12	11	3	5	6	13	8	4b	30	29	27	26	18	20	21	28	23	19b
	9	7b	7b	4b	14	14	4b	7b	15	15	24	22b	22b	19b	29	29	19b	22b	30	30

⊗	Non-populated site
PD	Photo Diode
Th	Thermistor

Figura 3.5: singolo modulo della lampada

La numerazione presente in figura indica il posizionamento delle 17 catene nella zona più vicina al connettore e delle 17 catene della zona posizionata in centro-lampada.

La lampada è collegata tramite delle tubature a tenuta stagna ad un termo-chiller: tramite un percorso simmetrico attraverso i substrati dei LED, si crea un percorso di raffreddamento dentro al quale scorre acqua controllata termicamente (Figura 3.6).

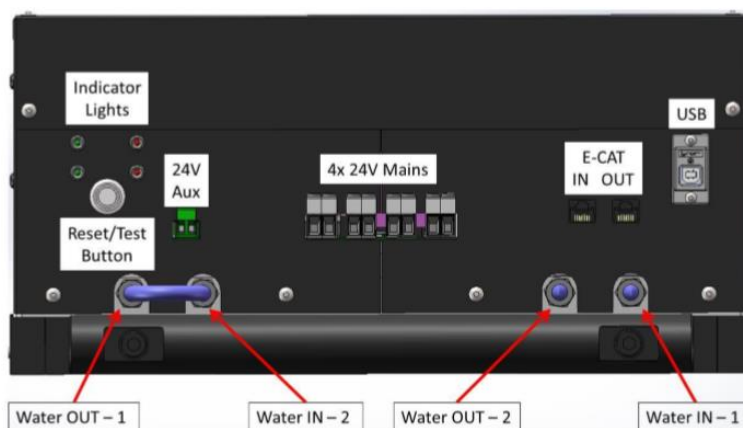


Figura 3.6: pannello posteriore della lampada con collegamenti al chiller

La presenza di un sistema di regolazione della temperatura è dovuta alla forte dipendenza dell'intensità dei LED alla temperatura di lavoro: mantenere una temperatura costante tramite acqua raffreddata permette di garantire la stabilità dello spettro.

### 3.3.2 Il carico elettronico

In Figura 3.7 è rappresentato il pannello frontale del carico elettronico.

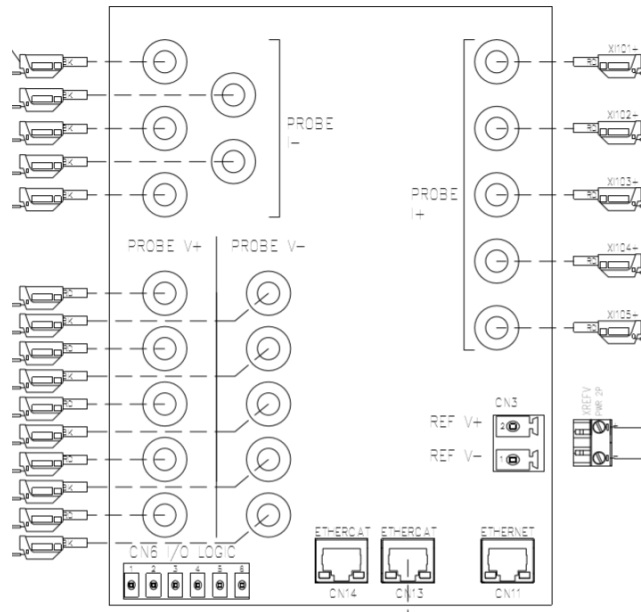


Figura 3.7: pannello frontale del carico elettronico

I connettori di alimentazione non sono rappresentati. Nello schema sono riportati:

- 10 collegamenti di tensione: 5 per le barre di contattazione inferiori e 5 per le barre superiori
- 10 collegamenti di corrente: 5 per le barre di contattazione inferiori e 5 per le barre superiori
- Un ingresso ETHERCAT per comunicare con il modulo
- Un ingresso di tensione per la cella di riferimento: esso viene utilizzato per ottenere l'irradianza emessa dalla lampada

Il carico elettronico è ripartito in diversi canali che permettono l'effettuazione delle diverse prove per la caratterizzazione delle celle.

Dal punto di vista circuitale, utilizzando il modello semplificato ad un diodo di un cella solare, esso può essere rappresentato come un generatore di tensione con in serie una resistenza di ingresso (Figura 3.8).



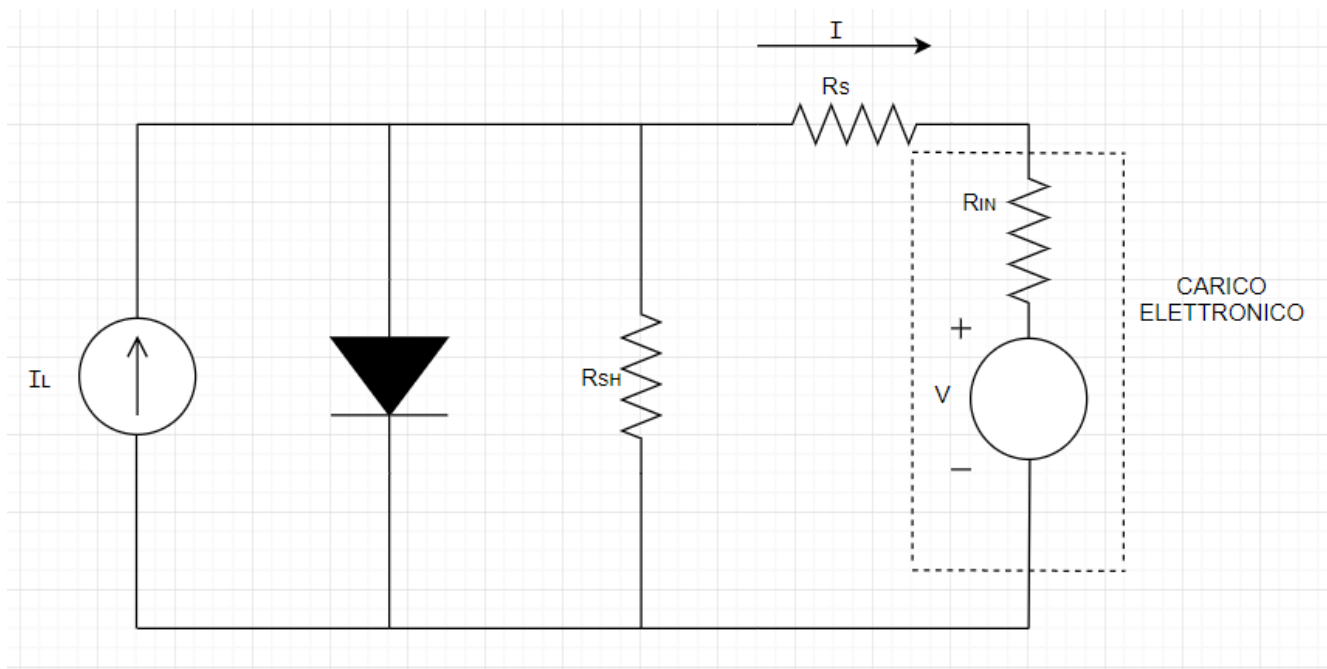


Figura 3.8: modello circuitale semplificato per il carico elettronico connesso alla cella

La resistenza d'ingresso del carico elettronico risulta in serie alla resistenza serie della cella di riferimento. Nello schema non è stata considerata l'impedenza equivalente dei cavi di collegamento tra carico e sistema di contattazione.

Per acquisire le caratteristiche I-V il carico riceve come parametri di "ricetta" alcuni parametri, tra cui:

- Il numero di regioni da utilizzare per acquisire l'intera caratteristica I-V
- Il range di tensione da applicare
- Il tempo di campionamento
- Il passo di quantizzazione della tensione
- Il parametro di oversampling dei campioni
- La direzione di acquisizione (tensioni crescenti o decrescenti)

In Figura 3.9 si riporta la sezione dell'interfaccia utilizzata per la modifica della ricetta di acquisizione.

Region	Start Volt [mV]	Sample Time [us]	Amplitude [uV]	Over Sampling	Direction
1	80	100	4000	1	1
2	-76	200	200	1	1
3	-80	300	2500	6	-1
4	0	0	0	0	0

Figura 3.9: definizione della ricetta di acquisizione da interfaccia grafica

Al carico elettronico è demandato anche il compito di interrompere l'acquisizione nel caso in cui si verificassero malfunzionamenti come overcurrent o contattazione non corretta della cella campione.

### 3.3.3 La cella di riferimento

Il dispositivo di riferimento prescritto da normativa, come descritto nel secondo capitolo, è una cella fotovoltaica di silicio monocristallino di dimensioni pari a 2cm x 2cm, incapsulata in un involucro dal quale escono due distinti segnali: quello di temperatura e quello di tensione.

La temperatura è acquisita tramite una PT100, con una precisione di  $\pm 0.2^{\circ}\text{C}$ .

La tensione misurata dalla cella, invece, viene tradotta in un segnale di corrente tramite una resistenza di sensing del valore di  $1\text{m}\Omega$ .

Come prescritto da normativa, la corrente di cortocircuito del dispositivo di riferimento presenta un legame lineare con l'irradianza alla quale la cella viene illuminata. Dal segnale di corrente prelevato, dunque, si ottiene una densità di potenza tramite un opportuno riscaldamento delle grandezze.

Posizionare la cella di riferimento all'interno dell'area di test significa ottenere un valore di irradianza da associare alla misura effettuata sul dispositivo campione.

In Figura 3.10 si riportano i tipici andamenti dei due segnali in un numero di acquisizioni pari a 30.

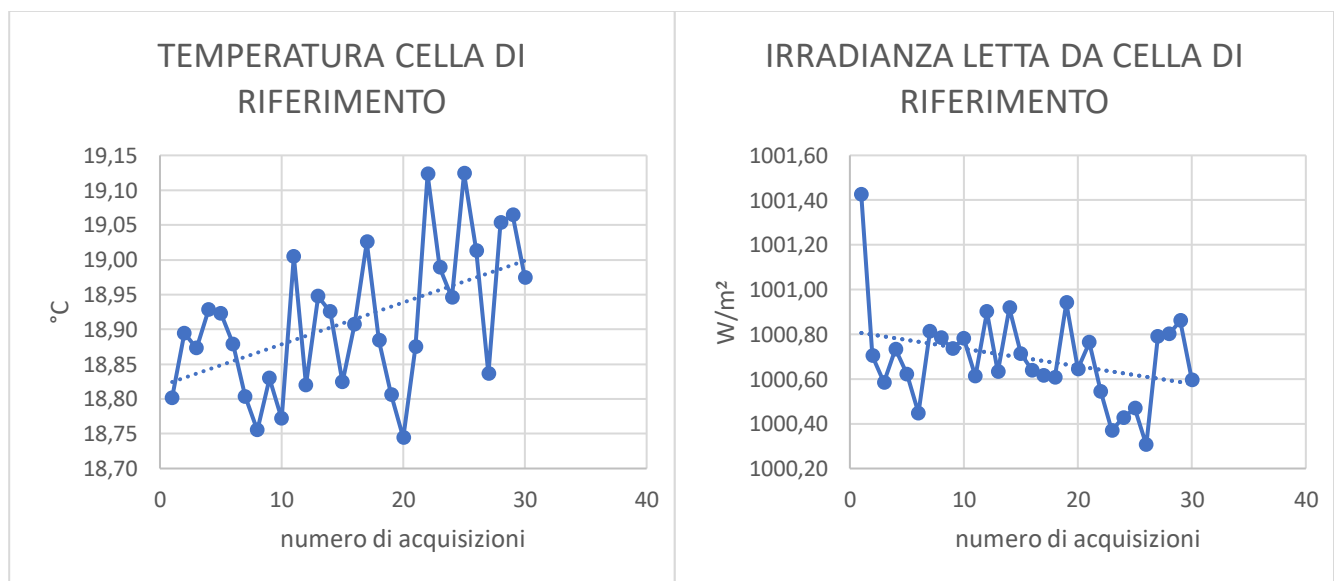


Figura 3.10: segnali di temperatura e irradianza letti da cella di riferimento

La temperatura della cella di riferimento serve a correggere i valori di tensione (e dunque di irradianza) acquisiti dalla cella stessa. Come analizzato nel paragrafo 2.2.4, viene applicata una correzione lineare per ottenere un valore di densità di potenza in condizioni standard.

## 3.4 Analisi software del sistema di misura

### 3.4.1 Calibrazione del sistema

Prima di procedere alla caratterizzazione di una cella solare con il simulatore solare di AppliedMaterials risulta necessario procedere alla calibrazione del sistema.

La calibrazione risulta necessaria per due motivi principali:

- La lettura del valore di irradianza con la cella di riferimento non tiene conto della potenza persa a causa dell'ombreggiamento dovuto al sistema di contattazione. Infatti, le barre di contattazione superiori riflettono una certa percentuale di luce emessa dalla lampada. È necessaria una calibrazione, dunque, per portarsi nelle condizioni standard di 1000W/m<sup>2</sup> effettivi sulla cella sotto test.
- Poiché non si ha nessuna misura diretta della densità di potenza assorbita dalla cella campione, è necessario calibrare il sistema sfruttando una cella dalle caratteristiche note, per relazionare il suo valore di corrente di cortocircuito con l'irradianza effettivamente assorbita.

In Figura 3.11 si riporta la sezione dell'interfaccia grafica utilizzata per la calibrazione del sistema

The screenshot displays the '1.IVT - Calibration System' interface. It features a top navigation bar with 'MONITOR', 'STATISTIC', 'DIAGNOSTICS', 'DATA & RECIPES', 'UTILITY', 'CONFIG', and 'CALIBRATION'. The 'CALIBRATION' section is active, showing three main panels:

- Calibration Commands:** Includes buttons for 'Start Calibration' (CALIB), 'Load calibration data file' (LOAD), and a 'Calibration Method' dropdown set to 'Pmax Target'. It also shows fields for 'Number of calibration cycles' (6), 'Waiting time between 2 calibration measures [ms]' (2000), 'Calibration counter' (0), 'Irr [W/m<sup>2</sup>]' (1185.967), and 'Vofs 1000 [V]' (-0.001).
- Calibration Settings:** Includes 'Cable Compensation' (0.000), 'Rsh Offset' (0.000), 'Voc Offset' (0.000), 'Isc Offset' (0.110), and 'FF Offset' (0.000).
- Calibration Results:** A table showing parameters, targets, and target deltas.

Parameter	Target	Target Delta
Isc	0.000	0.000
Voc	0.000	0.000
Pmax	0.000	0.000
Impp	0.000	0.000
Vmpp	0.000	0.000
FF	0.000	0.000
Eff	0.000	0.000
Irr	0.000	0.000

At the bottom, there are 'Calibration Eff Offset Settings' with 'Threshold T1, T2' (19.000, 22.000) and 'Eff Offset 1, 2, 3' (0.000, 0.000, 0.000). The interface also includes a status bar at the bottom with 'AUTO', 'MAN', 'SERIAL MEASUREMENT', 'ACQ IV', 'ACQ EL', and 'RESET' buttons.

Figura 3.11: calibrazione del simulatore solare

Come visibile nell'immagine, è presente una tabella contenente le figure di merito e i principali parametri elettrici di una cella fotovoltaica.

La procedura di calibrazione consiste nei seguenti passi:

1. Viene contattata una cella solare dai parametri noti, solitamente si tratta di una cella caratterizzata da appositi laboratori.
2. Si inseriscono manualmente i parametri da certificato, specificando il range di incertezza dichiarato nel certificato stesso.
3. Viene attivata la lampada: essa esegue un numero di illuminazioni predefinito (solitamente 6) con lo scopo di ottenere come output di misura parametri quanto più possibile simili a quelli da certificato
4. Ad ogni ciclo di misura viene calcolato un parametro di irradianza: esso rappresenta il valore letto dalla cella di riferimento che equivale alle condizioni standard per la cella campione.

A titolo esemplificativo, in Figura 3.11 il parametro di irradianza di calibrazione calcolato a fine procedura vale circa  $1186\text{W/m}^2$ : ciò sta a significare che per illuminare la cella campione con  $1000\text{W/m}^2$  effettivi, è necessario che la densità di potenza emessa dalla lampada sia tale da leggerne  $1186$  tramite cella di riferimento.

Di conseguenza è possibile concludere che  $186\text{W/m}^2$  sono persi a causa dell'effetto di ombreggiamento dovuto alle barre di contattazione.

Tale procedura è giustificata dalla relazione lineare tra corrente di cortocircuito e irradianza. Il valore di  $I_{sc}$  dichiarato da certificato della cella dai parametri noti si riferisce a condizioni standard di illuminazione: variando il parametro di irradianza durante il ciclo di calibrazione fino a ottenere come output di misura il valore corretto di  $I_{sc}$  per la cella di calibrazione, è possibile regolare la potenza della lampada per raggiungere le condizioni standard di illuminazione a fronte delle perdite per ombreggiamento.

### 3.4.2 Acquisizione delle caratteristiche I-V e calcolo parametri elettrici

Dopo aver concluso la fase di calibrazione del sistema, è possibile procedere alla caratterizzazione delle celle campione. Per ogni cella solare vengono eseguite quattro prove differenti. In Tabella 3.2 si riportano i quattro tipi di prova con i relativi parametri di input e output.

tipo di prova	irradianza [ $\text{W/m}^2$ ]	range di tensione tipico [V]	parametri calcolati
dark reverse	0	da -10 a 0	$I_{01}, I_{02}$
dark shunt	0	da -0.05 a 0.05	Rsh
one sun	1000	da 0 a 0.75	$\eta, FF, V_{oc}, I_{sc}, P_{max}$
one half sun	500	da 0 a 0.75	$R_s$

Tabella 3.2: elenco delle prove effettuate su singola cella campione

Le prove vengono effettuate nello stesso ordine con cui sono riportate in Tabella 3.2.

Per ognuna delle quattro prove vengono acquisiti i campioni di tensione e corrente. Dopo aver applicato la correzione in temperatura e irradianza, dalle curve I-V vengono estrapolati i parametri elettrici e le figure di merito riportati in tabella. I parametri assumono i seguenti significati:

- $I_{01}, I_{02}$  sono le correnti di saturazione del modello a due diodi della cella solare

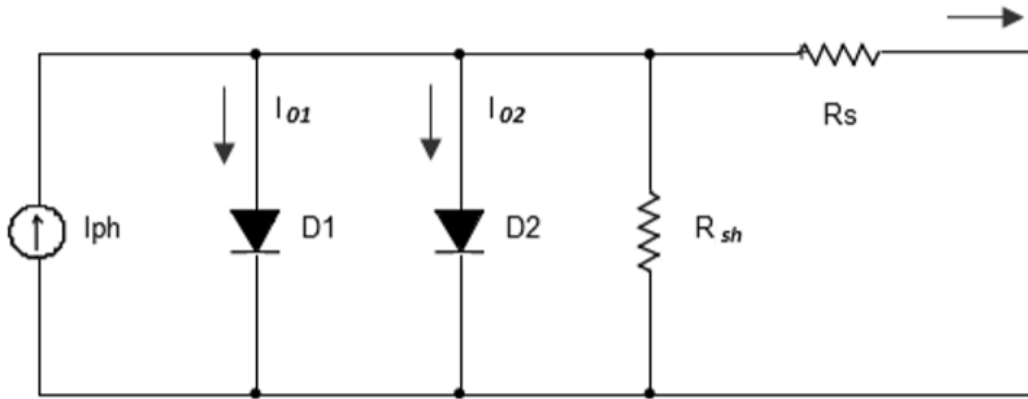


Figura 3.12: modello a due diodi della cella solare

- $R_{sh}$  è la resistenza parallelo [ $\Omega$ ]

La resistenza parallelo viene calcolata come inverso della pendenza della retta tangente la curva di corrente nel punto  $(0, I_{sc})$  a lampada spenta.

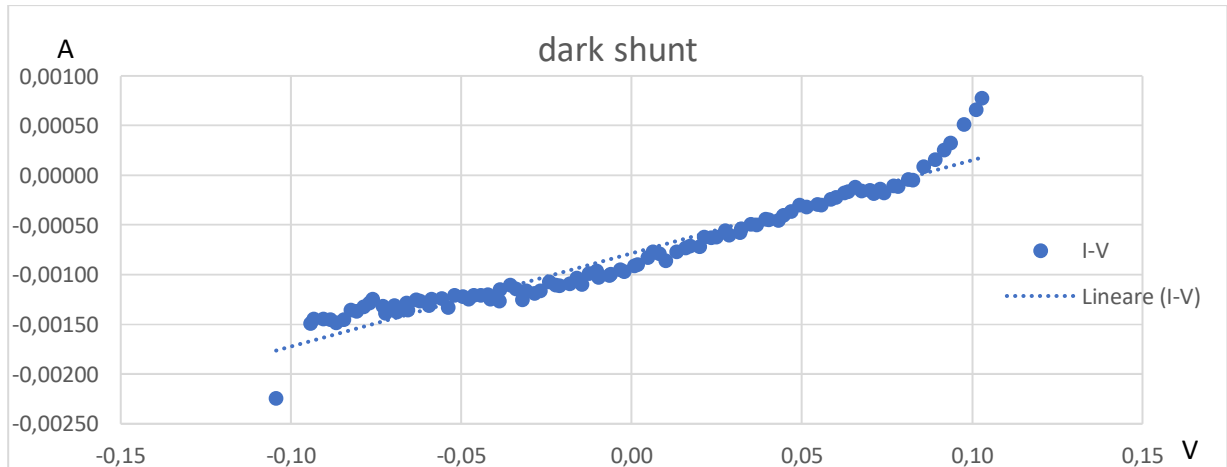


Figura 3.13: acquisizione I-V durante la prova di dark shunt

- $\eta$  è l'efficienza della cella solare [%]

L'efficienza viene calcolata tramite la relazione:

$$\eta = \frac{P_{\max} [\text{W}]}{\text{Area} [\text{m}^2] \cdot 1000 [\text{W}/\text{m}^2]} \cdot 100 \quad (3.1)$$

- FF è il Fill Factor [%]

Il fill factor viene calcolato tramite la relazione:

$$\eta = \frac{P_{\max} [\text{W}]}{V_{\text{oc}} [\text{V}] \cdot I_{\text{sc}} [\text{A}]} \cdot 100 \quad (3.2)$$

- $V_{\text{oc}}$  è la tensione di circuito aperto [V]

La tensione di circuito aperto viene calcolata tramite interpolazione lineare della curva I-V nell'intorno del punto (0,  $V_{\text{oc}}$ ). Una volta ottenuta l'equazione della retta interpolatrice, il valore di tensione di circuito aperto è dato dall'intersezione di quest'ultima con l'asse delle ascisse.

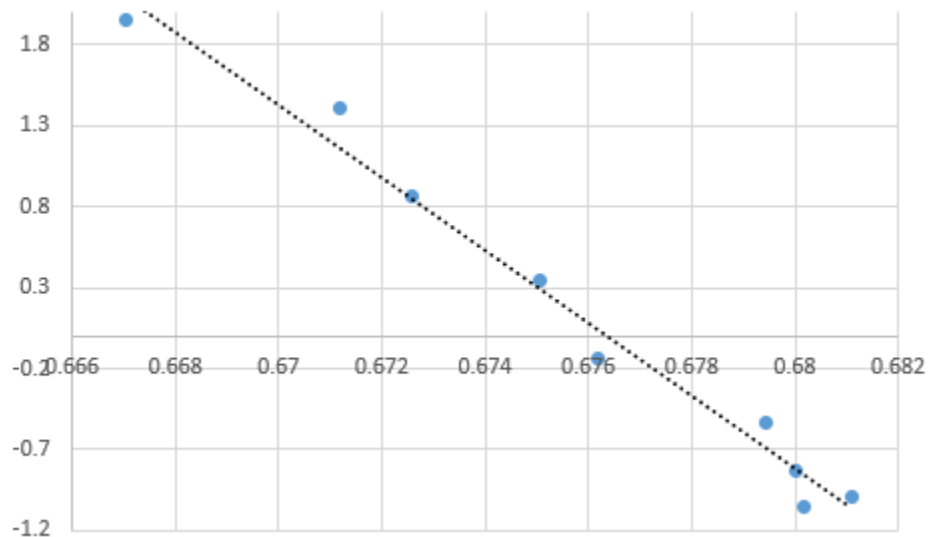


Figura 3.14: calcolo della tensione di circuito aperto

- $I_{\text{sc}}$  è la corrente di cortocircuito [A]

La corrente di cortocircuito viene calcolata tramite interpolazione lineare della curva I-V nell'intorno del punto ( $I_{\text{sc}}$ , 0). Una volta ottenuta l'equazione della retta interpolatrice, il valore di corrente di cortocircuito è dato dall'intersezione di quest'ultima con l'asse delle ordinate.

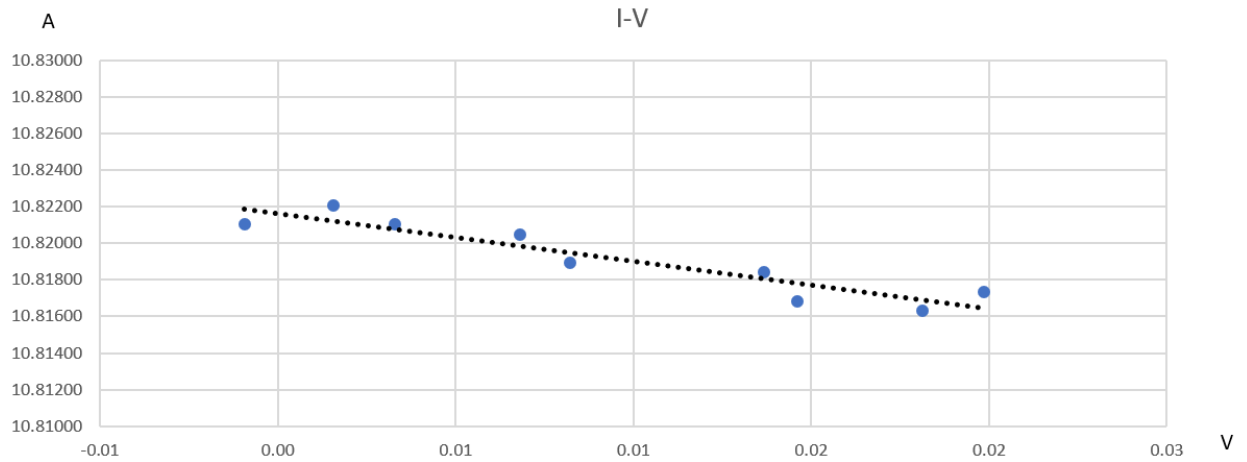


Figura 3.15: calcolo della corrente di cortocircuito

- Pmax è la massima potenza erogabile dalla cella solare [W]  
Tale parametro viene calcolato tramite un algoritmo di ricerca del massimo del prodotto tra tensione e corrente per tutti i punti della curva a 1000W/m<sup>2</sup>.

- Rs è la resistenza serie [Ω]  
La resistenza serie viene calcolata sfruttando sia la curva a 1000 W/m<sup>2</sup> che quella a 500 W/m<sup>2</sup>.  
A differenza della resistenza parallelo, non è sufficiente una semplice interpolazione nell'intorno del punto (0,Voc) della caratteristica I-V. Ciò è dovuto al fatto che la curva risulta meno stabile in questa zona della curva rispetto all'intorno del punto (Isc,0).

La procedura corretta prevede di calcolare i punti (Isc1-ΔI, Visc1) e (Isc2- ΔI, Visc2), con ΔI pari a 1A.

A questo punto si ricava il coefficiente angolare della retta passante per i due punti tramite la relazione:

$$m = \frac{(Isc1-\Delta I)-(Isc2-\Delta I)}{Visc1-Visc2} [1/\Omega] \quad (3.3)$$

Il valore della resistenza serie Rs è dato da

$$Rs = \frac{1}{m} [\Omega] \quad (3.3)$$

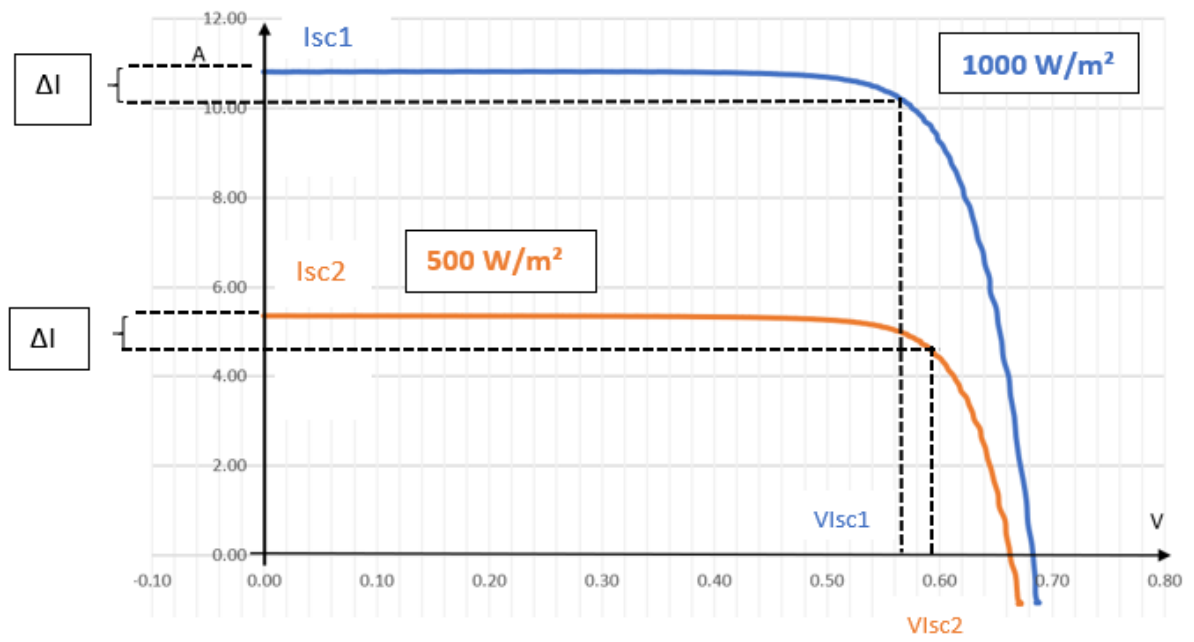


Figura 3.15: calcolo della resistenza serie  $R_s$



# CAPITOLO 4: PROCEDURA DI CORREZIONE AUTOMATICA DELL'UNIFORMITA' SPAZIALE DELLA LAMPADA

## 4.1 Introduzione

Per un simulatore solare assume un ruolo chiave il tema dell'uniformità spaziale e dell'uniformità spettrale.

Risulta di primaria importanza assicurarsi che lo spettro della sorgente luminosa e che l'intensità della sorgente stessa rispettino le norme definite nel secondo capitolo di questo elaborato, identificando la relativa classe di appartenenza.

Varie possono essere le cause che portano la lampada LED a degli scostamenti non trascurabili dalle condizioni standard definite: degrado nel tempo delle catene di LED, variazioni della geometria del simulatore o variazioni della distanza tra piano dei LED e superficie di misura.

L'obiettivo di questo e del prossimo capitolo è quello di fornire una procedura di auto calibrazione della lampada affinché si possa garantire, a fronte delle problematiche appena presentate, una condizione di non-uniformità spaziale prestabilita mantenendo quanto più possibile una forma accettabile per lo spettro.

L'intera procedura è rappresentabile intuitivamente tramite il diagramma di flusso di Figura 4.1.

La procedura inizia con la determinazione dell'attuale parametro di non-uniformità (NU) sul piano di misura della cella.

Successivamente, viene eseguita una calibrazione della lampada con lo scopo di abbassare tale parametro.

A fine calibrazione viene avviata una seconda procedura che è in grado di misurare lo spettro di emissione della lampada: tali informazioni vengono salvate in un file di calibrazione che funge da target per le future correzioni delle righe spettrali affette da variazioni.

In questa maniera l'intenzione è quella di garantire che il simulatore solare utilizzi una sorgente luminosa con caratteristiche quanto più possibile vicine alle condizioni di fabbrica, a fronte di un degrado dovuto al ripetuto utilizzo della lampada nel tempo.

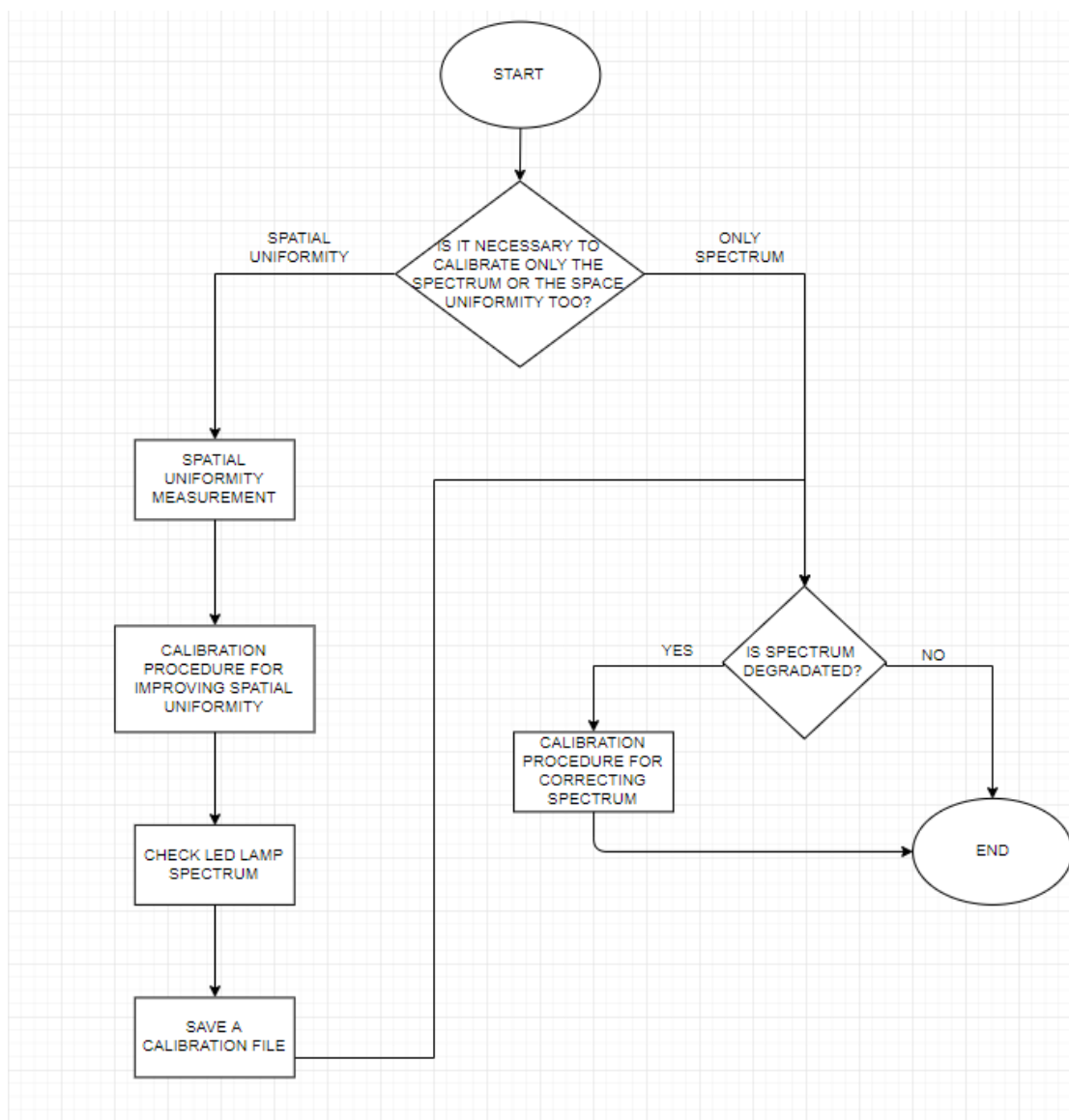


Figura 4.1: diagramma di flusso rappresentativo della procedura di calibrazione della lampada LED

In questo capitolo viene presentato il problema dal punto di vista matematico, spiegando la strategia utilizzata per la definizione della funzione di calibrazione per l'uniformità spaziale. Vengono inoltre riportati i risultati delle simulazioni effettuate in ambiente Matlab, nonché il setup sperimentale adottato per le misure, l'implementazione della procedura in ambiente TwinCAT e i risultati dei test finali effettuati.

## 4.2 Setup sperimentale e dati raccolti con la cella di riferimento

Tramite la cella di riferimento viene misurata la non-uniformità sul piano dei nastri che sorreggono i wafers sotto test. Per poter far ciò è stato necessario definire le dimensioni dell'area sotto esame e una sua opportuna suddivisione in sotto-aree (pixels).

La norma per le misure di uniformità spaziale prescrive di suddividere l'area scelta in 64 pixels o, alternativamente, in sotto-aree dalle dimensioni di 2cm x 2cm.

Tuttavia, la funzione di auto calibrazione presa in esame in questa sede non si prefigge lo scopo di valere come procedura di classificazione. Si vuole dimostrare, anzi, che considerando un minor numero di pixels rispetto alla prescrizione della norma, i risultati dei test sono comunque confrontabili (questa affermazione verrà verificata nel paragrafo 5.3.)

Alla luce di ciò è stata creata una griglia di 5 righe x 4 colonne, con disposizione simile a quella delle zone della lampada.

In tale maniera risulta possibile fissare la griglia su di un supporto in plexiglass e fissare saldamente il supporto sotto la lampada: spostando la cella di riferimento manualmente attraverso il sistema di riferimento creato si acquisiscono 20 valori di irradianza e viene così calcolato il parametro di non-uniformità in percentuale secondo la formula:

$$NU = \frac{(IrrMax - IrrMin)}{(IrrMax + IrrMin)} \times 100 \quad (4.1)$$

Dove i valori IrrMax e IrrMin rappresentano rispettivamente il massimo e il minimo valore misurato di irradianza all'interno della griglia.

La Figura 4.2 rappresenta la griglia utilizzata per il test.

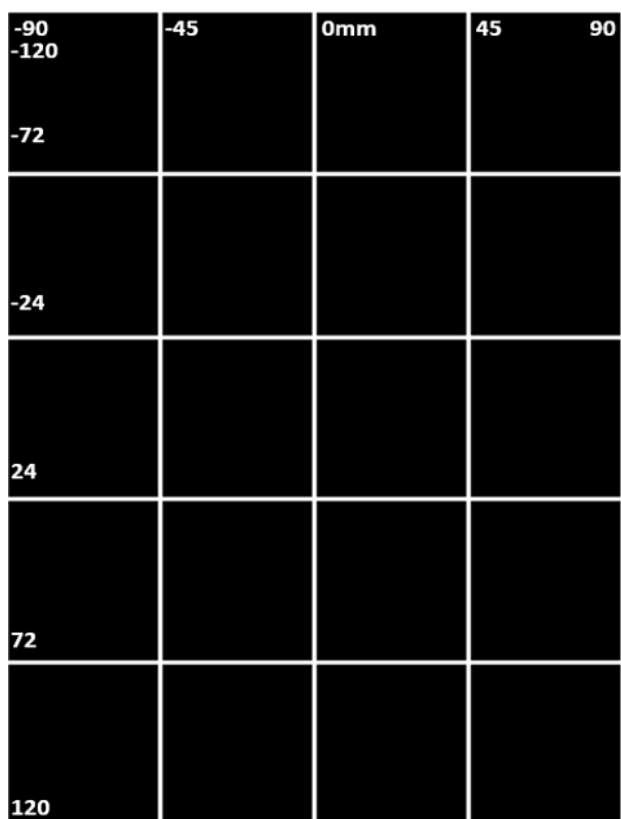


Figura 4.2: griglia di misura per uniformità spaziale

Come si evince dalle indicazioni poste lungo gli assi, l'area scelta per i test ha dimensioni pari a 18cm x 24 cm.

A questo punto è necessario precisare alcuni aspetti relativi alla procedura di misura affinché i risultati possano essere considerati validi.

Prima di tutto, la normativa IEC 60904-9 prescrive di procedere contemporaneamente alle acquisizioni di irradianza sui diversi pixels: ciò è richiesto per assicurarsi che le misure siano riferite alla sorgente luminosa nelle medesime condizioni di potenza e temperatura.

Nel caso delle misure prese in considerazione in questo elaborato, le acquisizioni sui diversi pixel vengono eseguite sequenzialmente: tuttavia, le condizioni ambientali nell'arco del test sono da considerarsi costanti.

A riprova di questa affermazione si riporta la Figura 4.3: essa rappresenta la temperatura rilevata dalla cella di riferimento durante l'arco delle 20 acquisizioni.

Se si applica un'interpolazione lineare alla serie si ottiene un coefficiente angolare di 0.0042°C/acquisizione. Moltiplicando per il numero di acquisizioni:

$$\Delta T = 0.0042 \times 20 = 0.084^{\circ}\text{C} \quad (4.2)$$

Tale valore risulta inferiore di circa quattro volte la variazione causata dal rumore in acquisizione.

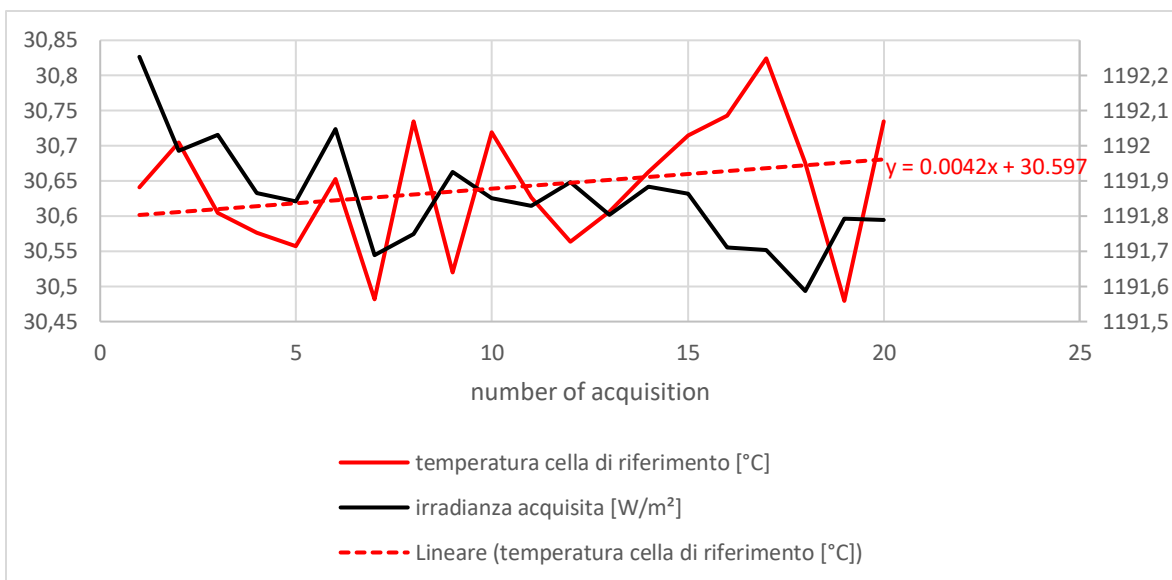


Figura 4.3: andamento della temperatura della cella di riferimento nell’arco di 20 acquisizioni

È possibile confrontare l’impatto del rumore in acquisizione della temperatura della cella di riferimento sulla correzione di irradianza con il rumore in acquisizione dell’irradianza stessa. A tale scopo si consideri la tabella 4.1:

	irradianza acquisita [W/m <sup>2</sup> ]	temperatura cella di riferimento [°C]
<b>max</b>	1192.253	30.824
<b>min</b>	1191.587	30.479
<b>Δ</b>	0.666	0.345
<b>media</b>	1191.855	30.641

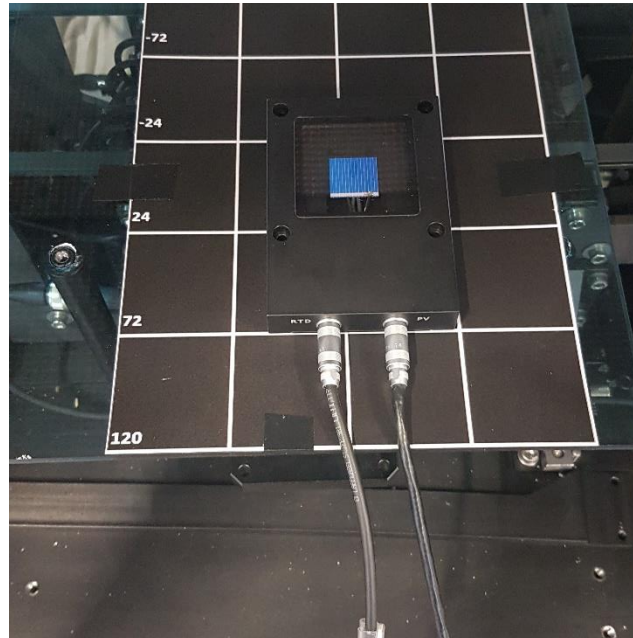
Tabella 4.1: irradianza e temperatura cella di riferimento per 20 acquisizioni

Dai valori in tabella si può leggere che nell’arco del test la differenza tra il valore massimo ed il valore minimo di irradianza risulta essere Δ=0.67 W/m<sup>2</sup>. Tale valore non risulta compensato in temperatura. Scegliendo la media aritmetica dei 20 valori come parametro rappresentativo ed applicando la correzione alle temperature Tmax e Tmin si ottiene una differenza pari a:

$$\Delta_{\text{corrected}} = \text{Irr}_{\text{corrected}} [\text{Irr}_{\text{media}}, \text{Tmax}] - \text{Irr}_{\text{corrected}} [\text{Irr}_{\text{media}}, \text{Tmin}] \quad (4.3)$$

$$\Delta_{\text{corrected}} = \text{Irr}_{\text{media}} \left( \frac{B (\text{Tmax} - \text{Tmin})}{100} \right) = 0.189 \text{ [W/m}^2\text{]} \quad (4.4)$$

Dunque, è possibile concludere che il rumore in acquisizione della temperatura ha un effetto trascurabile rispetto alla rumorosità stessa della misura di irradianza. Ciò significa che se si avessero 20 diverse celle posizionate sui 20 pixels e si procedesse ad acquisire contemporaneamente in tutte le zone, si otterrebbe un risultato molto simile rispetto al caso delle misure sequenziali.



*Figura 4.4: setup sperimentale dei test per la misura di uniformità spaziale*

Di seguito viene riportata una tabella riassuntiva con tre test effettuati per la misura del parametro di non uniformità (Tabella 4.2).

		Irradianza [W/m <sup>2</sup> ]		
		test 1	test 2	test 3
pixel	1	1205.05	1203.24	1203.71
	2	1202.42	1199.96	1201.08
	3	1207.52	1205.81	1205.89
	4	1212.61	1211.93	1213.02
	5	1211.52	1210.37	1210.50
	6	1203.21	1203.03	1202.57
	7	1201.23	1200.55	1200.90
	8	1202.53	1201.42	1201.95
	9	1221.43	1220.23	1220.05
	10	1209.43	1208.46	1208.13
	11	1206.16	1205.47	1205.55
	12	1206.03	1204.74	1204.44
	13	1226.89	1226.47	1226.63
	14	1218.89	1217.86	1219.00
	15	1215.26	1214.52	1214.24
	16	1212.21	1211.72	1212.34
	17	1222.87	1223.50	1222.96
	18	1222.47	1220.78	1220.92
	19	1219.77	1217.39	1219.00
	20	1216.46	1215.06	1216.40
	max	1226.89	1226.47	1226.63
	min	1201.23	1199.96	1200.90
	<b>NU[%]</b>	1.06	1.09	1.06

*Tabella 4.2: valori di irradianza e parametro NU per tre test*

Dai risultati dei test è possibile estrapolare due informazioni. La prima, riguardante il parametro NU, indica che la procedura manuale di misura risulta essere ripetibile: si ha uno scostamento dello 0.03% tra il secondo test e gli altri due.

In secondo luogo, osservando le tre misure su ognuno dei 20 pixel, ci si rende conto che l'entità del range di misura è comparabile con  $\Delta=0,67 \text{ W/m}^2$ , valore ricavato in Tabella 4.1.

Ciò sta a significare che l'errore commesso nel posizionare la cella di riferimento manualmente, anziché sfruttare una struttura meccanica, risulta trascurabile se comparato dall'errore introdotto dalla lettura di irradianza della cella di riferimento.

Per dimostrare ciò si è inoltre calcolato un valore NU fittizio per 20 acquisizioni, ma con cella di riferimento fissa nella stessa posizione. In tal modo è possibile identificare un valore di risoluzione al

di sotto del quale non è possibile scendere con l'attuale configurazione hardware. I risultati sono riportati in Tabella 4.3:

Irradianza [W/m <sup>2</sup> ]		
1212.94		
1212.64		
1212.74		
1212.58		
1212.57		
1212.73		
1212.46		
1212.38		
1212.68	max	1212.94
1212.49	min	1212.25
1212.52	<b>NU[%]</b>	<b>0.03</b>
1212.62		
1212.50		
1212.55		
1212.50		
1212.33		
1212.28		
1212.25		
1212.56		
1212.42		

*Tabella 4.3: risoluzione della misura del parametro NU*

Come si evince dalla tabella, scostamenti dell'ordine di 0.03% relativi al parametro NU sono giustificati dalla rumorosità della lettura di irradianza da parte della cella di riferimento.

Le considerazioni svolte fin qui si prefiggono lo scopo di convalidare il setup hardware per le misure, dimostrando come i test siano ripetibili e come le scelte operative (misure sequenziali, spostamento della cella a mano) producano un rumore trascurabile per i dati raccolti.



### 4.3 Approccio matematico e simulazioni in ambiente MATLAB

Una volta definito dettagliatamente il setup sperimentale per le misure, è stato necessario identificare una procedura che permettesse di pilotare i valori di irradianza letti singolarmente sui pixel definiti nel precedente paragrafo e dunque il valore di non-uniformità.

Il fulcro di questa operazione poggia sulla struttura della lampada stessa: come descritto nel paragrafo 3.3.1, la matrice dei LED è suddivisa in dieci moduli, composti a loro volta da due zone separate.

Per ognuna di queste zone è presente un parametro in ingresso denominato “local adjustment factor”: tale parametro viene definito come l’incremento percentuale di irradianza rispetto al valore nominale di emissione della zona stessa. In termini matematici:

$$IRR_{adj} = IRR \left( 1 + \frac{ADJ}{100} \right) \quad (4.5)$$

Dunque, la presenza di queste variabili permette di diminuire o aumentare selettivamente la potenza emessa dalle varie zone della lampada. Una volta impostato un determinato valore di aggiustamento, al firmware della lampada è delegato il compito di tradurre tale parametro in un incremento o decremento della corrente di alimentazione dei LED.

Così facendo risulta possibile, in linea teorica, diminuire il valore di irradianza massimo letto nei venti pixel e aumentare quello minimo.

Di conseguenza risulta intuitivo comprendere che il relativo parametro NU diminuisce di valore.

Ricavare una funzione analitica che espliciti la relazione tra potenza emessa da una singola zona e irradianza letta su ogni pixel non risulta fattibile: va considerato infatti che questa funzione dipende quasi completamente dalla geometria del sistema e le riflessioni multiple che avvengono nel percorso di illuminazione non sono identificabili, se non per via simulata.

La soluzione è stata quella di quantificare la percentuale di luce letta su ogni singolo pixel mantenendo accesa una singola zona volta per volta. In questa maniera, sfruttando la sovrapposizione degli effetti, è possibile scrivere:

$$U_j = \sum_{k=1}^{20} W_{ji} \left( 1 + \frac{ADJ_i}{100} \right) \quad (4.6)$$

Di seguito si riporta il significato dei termini utilizzati nell'equazione (4.6):

- $U_j$  è l'irradianza letta sul pixel  $j$ -esimo, con  $1 \leq j \leq 20$ . La lettera "U" è stata scelta in sostituzione del termine "Uniformity".
- $W_{ji}$  è l'elemento della matrice quadrata  $W$  appartenente alla riga  $j$ -esima e alla colonna  $i$ -esima: tale matrice svolge il ruolo di funzione di trasferimento. Infatti, l'elemento di indice  $(j,i)$  rappresenta l'irradianza letta sul pixel  $j$ -esimo con solo la zona  $i$ -esima accesa. Inoltre, essa è rappresentabile come rapporto tra uscita e ingresso del sistema. La lettera "W" è stata scelta come abbreviazione del termine "weight" (per indicare il peso che ha ogni zona della lampada su ogni singolo pixel). Risulta  $1 \leq i \leq 20$ .
- $ADJ_i$  è il fattore di aggiustamento di potenza della zona  $i$ -esima della lampada.

Riscrivendo vettorialmente la relazione (4.6) si ottiene la forma più compatta:

$$\mathbf{U} = \mathbf{W} \left( \mathbf{1} + \frac{\mathbf{ADJ}}{100} \right) \quad (4.7)$$

Le dimensioni di  $\mathbf{U}$ ,  $\mathbf{W}$  e  $\mathbf{ADJ}$  risultano essere rispettivamente:  $20 \times 1$ ,  $20 \times 20$  e  $20 \times 1$ .

È necessario, tuttavia, verificare che la sovrapposizione degli effetti sia applicabile in questo contesto. Non è scontato, infatti, che non possano subentrare degli effetti non lineari nella misura di irradianza sommando i contributi delle venti zone.

Un esempio può essere fornito dalla fisica della cella solare con la quale si procede per le acquisizioni. Come descritto nel paragrafo 3.3.3, la cella di riferimento applica una conversione al segnale di corrente generato dai fotoni incidenti sulla sua superficie tramite una resistenza di carico di valore basso ( $1\text{m}\Omega$  circa). In tale maniera si ottiene un segnale di tensione che successivamente viene convertito in irradianza. Il fatto di utilizzare una resistenza di valore prossimo alle condizioni di corto circuito fa sì che l'irradianza letta dalla cella di riferimento abbia un andamento lineare con il punto di tensione individuato dalla retta di carico.

Tuttavia, per particolari celle fotovoltaiche la linearità tra queste grandezze risulta applicabile solo per un limitato range di potenza assorbita: si parla dunque di condizioni di bassa/alta illuminazione per diversificare le caratteristiche I-V della cella.

Questa intrinseca non linearità può portare all'impossibilità di applicazione del principio di sovrapposizione degli effetti: infatti, illuminando la cella con l'intera lampada accesa si ottiene un valore di circa  $1000 \text{ W/m}^2$ . Il contributo di una singola zona risulta di conseguenza circa un ventesimo, ovvero  $50 \text{ W/m}^2$ .

Ci si rende conto facilmente che le due situazioni rientrano nei casi sopra descritti di alta e bassa illuminazione.

Un'ulteriore problematica che si è dovuta prendere in considerazione è quella relativa al rumore ambientale: la somma delle venti acquisizioni introduce nel valore di irradianza un termine di rumore (dovuto alla non perfetta oscurità della camera di misura) venti volte maggiore rispetto alla singola misura con lampada completamente accesa. È stato dunque necessario sottrarre ai valori di irradianza questo termine aggiuntivo.

Si riporta una tabella di confronto in cui la misura di uniformità è stata effettuata sommando i contributi delle venti zone e con lampada totalmente accesa (con fattori di aggiustamento nulli):

		Irradianza [W/m <sup>2</sup> ]	
		$\sum W_{ji}$	U <sub>i</sub>
pixel	1	1205.05	1203.71
	2	1202.42	1201.08
	3	1207.52	1205.89
	4	1212.61	1213.02
	5	1211.52	1210.50
	6	1203.21	1202.57
	7	1201.23	1200.90
	8	1202.53	1201.95
	9	1221.43	1220.05
	10	1209.43	1208.13
	11	1206.16	1205.55
	12	1206.03	1204.44
	13	1226.89	1226.63
	14	1218.89	1219.00
	15	1215.26	1214.24
	16	1212.21	1212.34
	17	1222.87	1222.96
	18	1222.47	1220.92
	19	1219.77	1219.00
	20	1216.46	1216.40
	max	1226.89	1226.63
	min	1201.23	1200.90
	<b>NU</b>	<b>1.06</b>	<b>1.06</b>

Tabella 4.4: verifica applicazione sovrapposizione degli effetti

Come leggibile da tabella, i parametri NU calcolati nei due casi differiscono per una quantità inferiore a  $\Delta=0.03\%$ .

Si può concludere dunque che l'equazione (4.8) risulta applicabile.

Dopo aver verificato che la matrice quadrata  $\mathbf{W}$  ha determinante non nullo, è possibile invertire la relazione (4.8):

$$100 [(\mathbf{W}^{-1}\mathbf{U}) - \mathbf{1}] = \mathbf{ADJ} \quad (4.9)$$

Tale equazione viene ripresa nel paragrafo 4.3.2 per lo studio di un particolare algoritmo di calibrazione.

#### 4.3.1 Minimizzazione del parametro di non-uniformità

Il primo algoritmo preso in considerazione si prefigge lo scopo di minimizzare il parametro di non-uniformità (NU) incrementando e decrementando ogni singolo elemento del vettore  $\mathbf{ADJ}$  di un fattore pari a  $\pm 1\%$ .

Nello specifico, l'algoritmo simula l'incremento della corrente di alimentazione dei led su ogni singola zona e misura il relativo parametro NU. Successivamente si applica la stessa procedura decrementando gli elementi di  $\mathbf{ADJ}$ .

In tale maniera, ad ogni ciclo di iterazione si ottengono  $2 \times \dim(\mathbf{ADJ})$  valori di non-uniformità. Tra questi viene individuato il valore minimo e si applica il relativo incremento/decremento per il vettore  $\mathbf{ADJ}$ .

L'algoritmo termina quando si verifica una delle seguenti condizioni:

1. Si raggiunge un prefissato numero massimo di iterazioni;
2. Si raggiunge un valore desiderato di non-uniformità;
3. Si raggiunge un prefissato valore minimo o massimo per gli elementi di  $\mathbf{ADJ}$ .

L'output fornito dall'algoritmo è rappresentato dai vettori  $\mathbf{ADJ}$  e  $\mathbf{U}$ .

Di seguito, in Figura 4.5, si riporta il diagramma di flusso che descrive l'algoritmo.

Va precisato, inoltre, che all'interno dell'algoritmo è presente un controllo sui valori assunti dagli elementi del vettore  $\mathbf{ADJ}$ : è possibile impostare un valore massimo e un valore minimo per tutti i fattori di aggiustamento.

Una volta raggiunti i valori limite per un particolare fattore di aggiustamento l'algoritmo procede con la minimizzazione di NU incrementando o decrementando i rimanenti elementi di  $\mathbf{ADJ}$ .

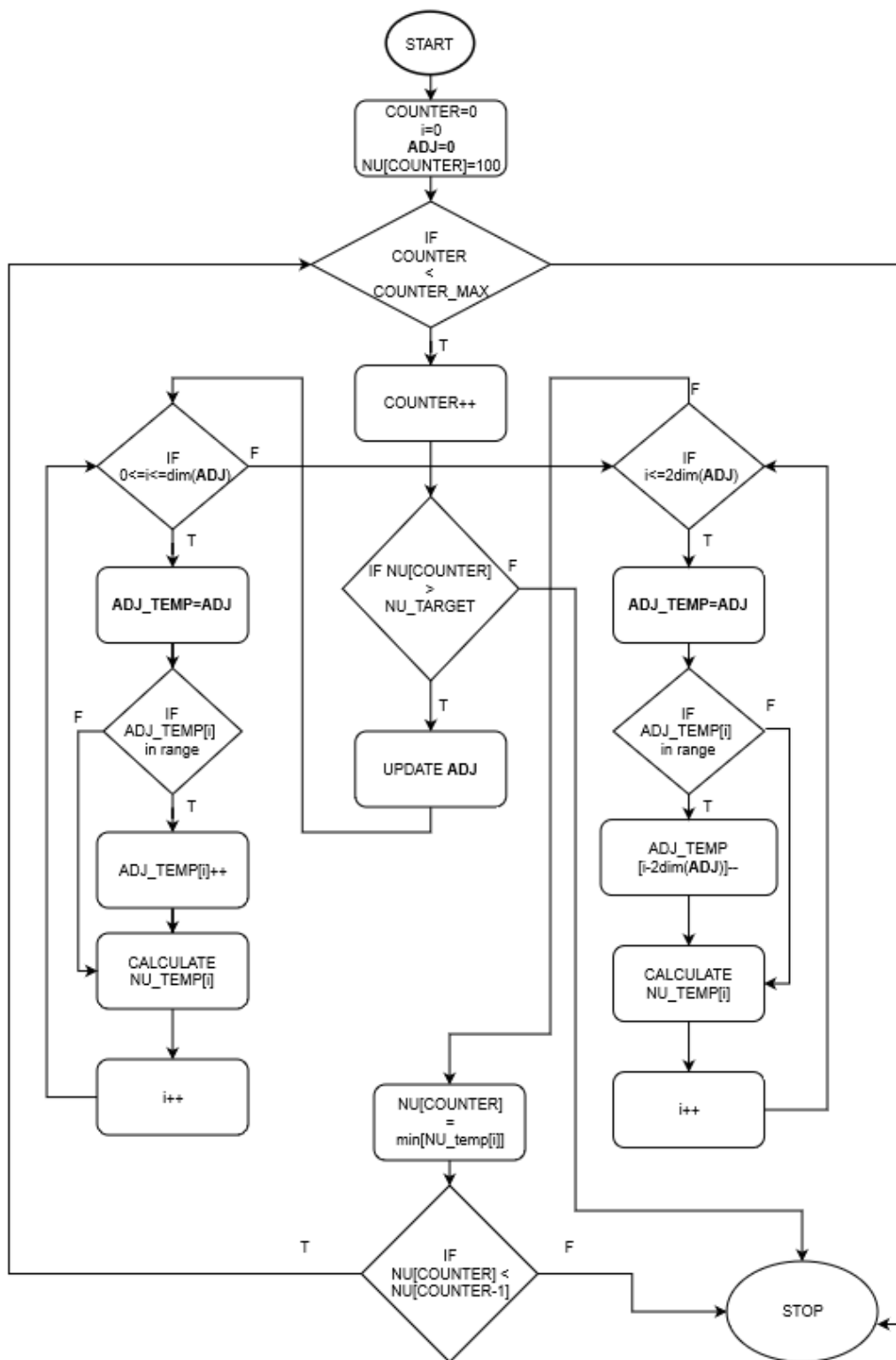


Figura 4.5: diagramma di flusso dell'algoritmo di minimizzazione della non-uniformità

### 4.3.2 Vettore target di uniformità

Il secondo algoritmo preso in considerazione opera in maniera sostanzialmente differente da quello relativo alla minimizzazione di NU. Il punto focale di questo algoritmo è l'identificazione di una mappa target di uniformità (vettore **U**): una volta individuata, si cerca di variare il vettore **ADJ** con lo scopo di ottenere una mappa di uniformità quanto più simile al target prestabilito.

Per ottenere un vettore target si considera la particolare simmetria e geometria della lampada.

La geometria del riflettore individua un punto di simmetria sul piano dell'area di misura (rappresentato in Figura da un cerchio blu). Tale punto coincide con la proiezione del centro della lampada sul piano stesso. A tale scopo si consideri la Figura 4.6:

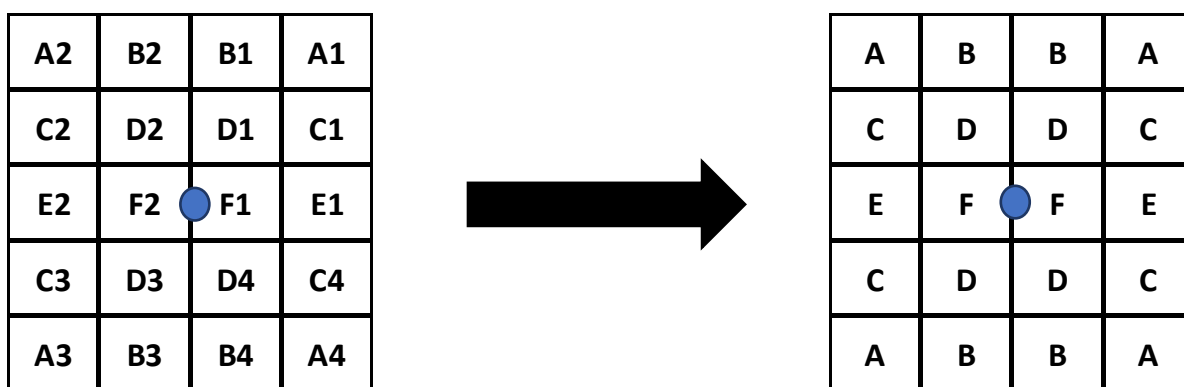


Figura 4.6: simmetria dei valori di irradianza sul piano dell'area sottostante la lampada

La griglia di sinistra rappresenta valori misurati di irradianza: lettere uguali tra loro e con pedice numerico diverso indicano valori di irradianza che ci si aspetta essere simili.

La griglia di destra rappresenta, invece, il target che l'algoritmo si prefigge di raggiungere.

Negli angoli e nei bordi (A,B,C,E) è possibile identificare le zone più "fredde", mentre le zone centrali (D,F) rappresentano un valore maggiore di irradianza.

Per calcolare il target di irradianza si esegue una media aritmetica per le classi di valori E ed F, mentre per le rimanenti si effettua una media aritmetica tra i due valori mediani misurati fisicamente:

- $A = \{(A1+A2+A3+A4) - \max\{A1,A2,A3,A4\} - \min\{A1,A2,A3,A4\}\} / 2;$
- $B = \{(B1+B2+B3+B4) - \max\{B1,B2,B3,B4\} - \min\{B1,B2,B3,B4\}\} / 2;$
- $C = \{(C1+C2+C3+C4) - \max\{C1,C2,C3,C4\} - \min\{C1,C2,C3,C4\}\} / 2;$
- $D = \{(D1+D2+D3+D4) - \max\{D1,D2,D3,D4\} - \min\{D1,D2,D3,D4\}\} / 2;$
- $E = (E1+E2) / 2;$
- $F = (F1+F2) / 2.$

Il fatto di eliminare dal calcolo della media valori estremi di irradianza per ogni classe permette effettivamente di ottenere un valore NU più basso di quello misurato, e permette di non prendere in considerazione nell'individuazione del target moduli malfunzionanti/degradati.

Successivamente, l'algoritmo varia a passi di  $\pm 1\%$  il vettore **ADJ** con lo scopo di produrre un vettore **U** quanto più simile al target calcolato.

In Figura 4.7 si riporta il diagramma di flusso dell'algoritmo di simulazione.

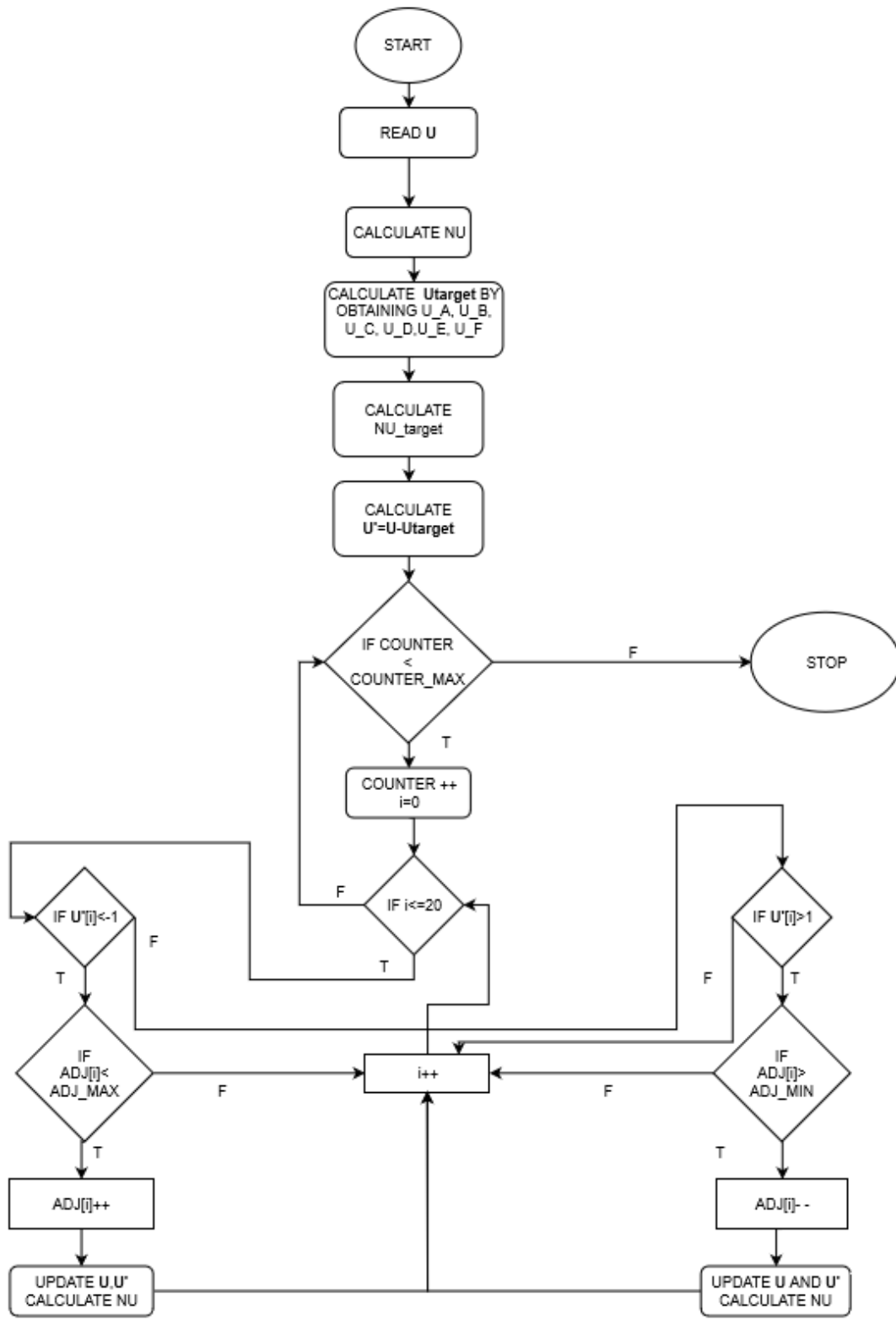


Figura 4.7: diagramma di flusso dell' algoritmo con vettore target



#### 4.4 Confronto tra gli algoritmi basati sui dati simulati

Per comparare il comportamento dei due algoritmi è necessario stabilire alcuni parametri in comune.

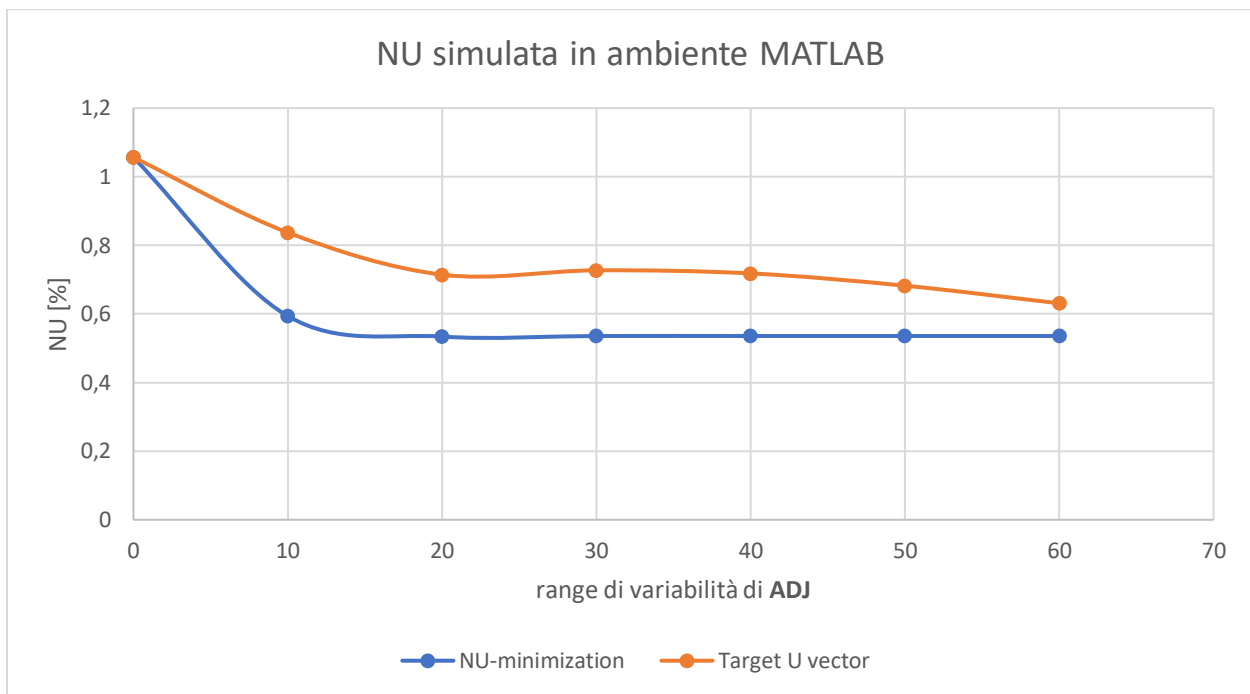
Prima di tutto risulta fondamentale fissare un range di variabilità per gli elementi di **ADJ** e ciò corrisponde a stabilire di quanto è possibile variare la potenza emessa dalla lampada.

In secondo luogo, è necessario fornire come input agli algoritmi un vettore **U** misurato sul quale applicare la correzione. Tale vettore (con relativo parametro NU) è riportato in Tabella 4.5.

pixel	<b>U</b> misurato [W/m <sup>2</sup> ]
1	1205.05
2	1202.42
3	1207.52
4	1212.61
5	1211.52
6	1203.21
7	1201.23
8	1202.53
9	1221.43
10	1209.43
11	1206.16
12	1206.03
13	1226.89
14	1218.89
15	1215.26
16	1212.21
17	1222.87
18	1222.47
19	1219.77
20	1216.46
NU= 1.06 % ; irradianza media= 1212.2 [W/m <sup>2</sup> ]	

Tabella 4.5: elementi di **U** misurati per il confronto degli algoritmi

Di seguito si riporta in Figura 4.8 la relazione che intercorre tra il valore NU simulato dai due algoritmi in ambiente MATLAB e l'ampiezza del range scelto per i fattori di correzione della corrente di alimentazione dei LED. I valori presenti in ascissa fanno riferimento ad un range simmetrico (es [-20%,+20]).



*Figura 4.8: relazione tra diminuzione di NU e ampiezza range del vettore ADJ*

Dal grafico si può concludere che l’algoritmo di minimizzazione del parametro NU risulta essere il più efficiente. Infatti, già limitando il range di variabilità di **ADJ** a [-10%, 10%] si ottiene un valore NU pari a 0.59% contro NU= 0.84% fornito dall’algoritmo che sfrutta il vettore **U** target.

Un altro aspetto da considerare è in che maniera il vettore **ADJ** viene incrementato dai due algoritmi: risulta preferibile non scostarsi troppo dalle condizioni di fabbrica della lampada, ovvero da **ADJ=0**.

Inoltre, è preferibile non aumentare eccessivamente la potenza emessa dai singoli moduli per diverse ragioni: la prima riguarda il tempo di vita della lampada stessa. Aumentare la potenza emessa significa alimentare le catene di LED tramite una corrente di maggiore intensità: lo stress termico causato da ciò può comportare un sensibile aumento della degradazione dei substrati e dunque una riduzione del tempo di vita dei LED.

Un altro tema è quello relativo al consumo di energia elettrica del simulatore solare: utilizzare una potenza maggiore per illuminare le celle può comportare costi significativamente maggiori se si considera che il simulatore è destinato a lavorare in maniera continuativa con un tasso di migliaia di emissioni luminose giornaliere.

Infine, come specificato nel paragrafo 4.1, alla calibrazione di uniformità spaziale può seguire una fase di calibrazione dello spettro della lampada. Se si considera che la correzione dello spettro è necessaria a fronte del degrado della matrice di LED e che il massimo range di variabilità per i fattori di

aggiustamento coincide con [-60%,+60%], risulta immediato comprendere che la porzione [0%, +60%] deve essere destinata preferibilmente alla correzione dello spettro.

Dalle considerazioni appena presentate si è deciso di fissare un intervallo di variazione degli elementi di **ADJ** in maniera tale che appartenga all'intervallo [-10%, 0%].

Di seguito, in Tabella 4.6, si riporta un confronto tra i vettori **ADJ** e **U** forniti in output dai due algoritmi, con range fissato a [-10%, 0%].

pixel	NU minimization		U target	
	[W/m <sup>2</sup> ]	ADJ [%]	[W/m <sup>2</sup> ]	ADJ [%]
1	1195.34	0	1195.40	-1
2	1193.47	0	1193.66	0
3	1199.35	0	1199.66	0
4	1205.04	0	1205.50	0
5	1198.46	0	1200.33	0
6	1191.45	0	1193.14	0
7	1190.82	0	1192.27	0
8	1193.20	0	1194.50	0
9	1204.37	-1	1208.67	-2
10	1194.10	0	1197.92	0
11	1192.63	-1	1195.85	0
12	1194.20	0	1196.84	0
13	1205.05	-4	1212.40	-8
14	1199.38	0	1205.79	-3
15	1198.24	0	1203.60	-1
16	1197.47	0	1201.85	0
17	1197.22	-10	1207.09	-1
18	1199.54	-10	1208.10	-1
19	1199.88	0	1206.96	-1
20	1199.09	0	1205.03	0
	NU= 0.59%		NU= 0.84%	
	irr. media=1197.42		irr. media=1201.23	

Tabella 4.6: confronto delle prestazioni degli algoritmi con range [-10%, 0%].

Si nota che l'algoritmo di minimizzazione tende a modificare un minor numero di fattori di aggiustamento, concentrandosi effettivamente sulle zone con irradianza che si discosta maggiormente dalla media. Al contrario, invece, l'algoritmo che sfrutta il vettore target tende a modificare un maggior numero di fattori di aggiustamento.

La spiegazione di ciò sta nel fatto che mentre il primo algoritmo tende solamente a diminuire la potenza delle zone più illuminate, il secondo deve emulare il più possibile un vettore i cui pixel sono generalmente tutti differenti dal vettore  $U$  misurato.

Di conseguenza è anche vero affermare che il comportamento dell'algoritmo con  $U$  target dipende in maniera maggiore dalle condizioni iniziali rispetto a quello di minimizzazione di  $NU$  e diventa di efficienza paragonabile solamente quando il range di variabilità si dilata.

Infine, si può notare come l'algoritmo di minimizzazione produca un vettore  $U$  caratterizzato da un'irradianza media inferiore all'algoritmo concorrente. A tal proposito si riporta in Figura 4.9 un grafico di confronto.

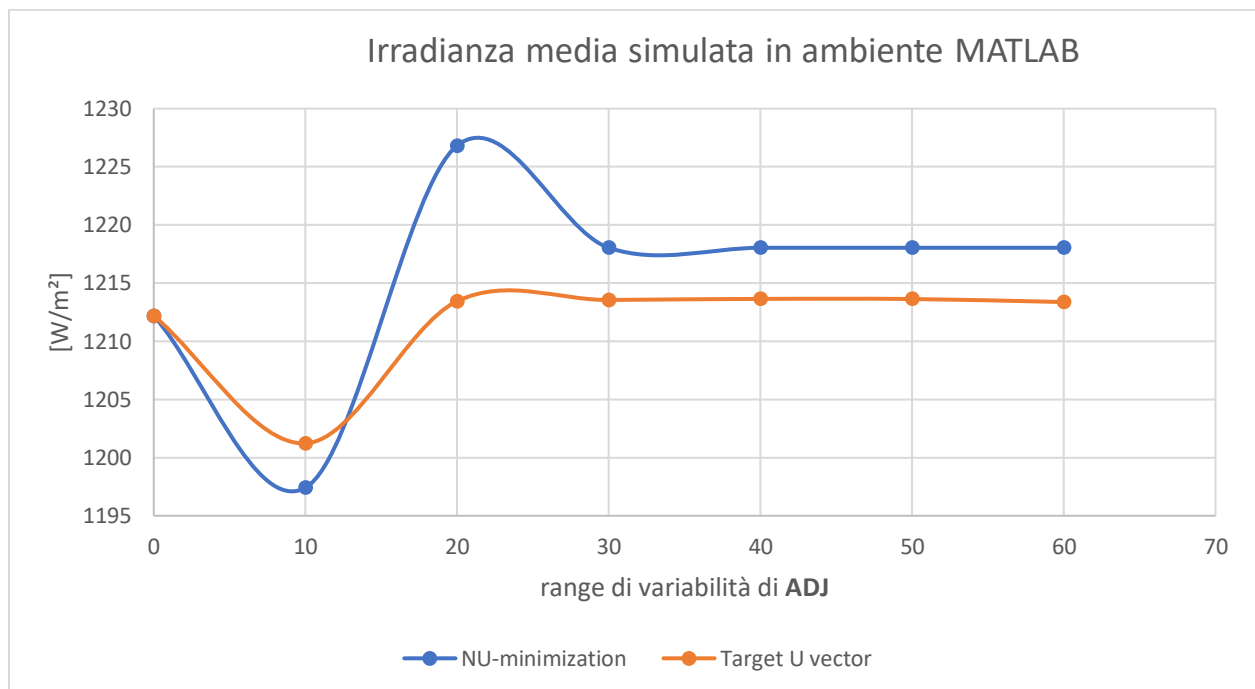


Figura 4.9: relazione tra irradianza media e ampiezza range del vettore  $ADJ$

Come si nota dal grafico in figura, per range limitati in modulo da 10% l'algoritmo di minimizzazione fornisce un'effettiva diminuzione dell'irradianza media, ulteriore fattore di forza dell'algoritmo.

Dalle considerazioni effettuate in questo paragrafo si deduce che la scelta ottimale per la funzione di calibrazione di uniformità spaziale della lampada è rappresentata dall'algoritmo di minimizzazione, con possibilità di variare i fattori di aggiustamento della corrente dei LED tra -10% e 0%.

Nei prossimi paragrafi verrà illustrata l'implementazione in ambiente TwinCat dell'algoritmo, fornendo i risultati raccolti dai test effettuati come effettivo riscontro delle simulazioni fin qui effettuate.

## 4.5 Implementazione in ambiente TwinCAT

L'implementazione dell'algoritmo di correzione dell'uniformità spaziale in ambiente TwinCAT è stata effettuata creando un'apposita funzione blocco comprendente ventotto stati.

Essi possono essere suddivisi nelle seguenti categorie:

- Idle state (1): in questo stato vengono resettate tutte le variabili e i messaggi indirizzati all'utente. Il sistema è pronto a ricevere un comando;
- Gestione file di testo (15): il vettore  $\mathbf{U}$  e la matrice  $\mathbf{W}$  vengono memorizzati in due file di testo. In tale maniera è possibile caricare i dati necessari per la calibrazione e/o l'ultima misura effettuata. Il salvataggio e il caricamento dati sono demandati all'interfaccia grafica;
- Simulazione real time (1): questo stato permette di inserire manualmente da interfaccia i fattori di aggiustamento e di ottenere istantaneamente una stima del parametro di non-uniformità e del vettore  $\mathbf{U}$ ;
- Calcolo (3): tramite questi stati viene implementato l'algoritmo a cui è stato dedicato il quarto capitolo di questa tesi;
- Comunicazione (2): questa coppia di stati gestisce le informazioni da inviare all'interfaccia grafica (es. messaggi d'errore) e comunica con la lampada, leggendo e scrivendo quando richiesto i valori dei coefficienti di aggiustamento;
- Acquisizione (6): servono per gestire l'accensione dei LED della lampada sia in fase di misura che in fase di calibrazione. I dati acquisiti relativi all'irradianza vengono salvati nelle strutture interne della funzione blocco;

In Figura 4.10 si riporta la lista completa.

```

1 {attribute 'qualified_only'}
2 {attribute 'strict'}
3 TYPE eSpatial_Uniformity :
4 (
5     idle_state := 0,
6     open_U_file,
7     read_U_file,
8     convert_U_file,
9     close_U_file,
10    open_W_file,
11    read_W_file,
12    convert_W_file,
13    close_W_file,
14    acquisition_U_vector,
15    acquisition_W_matrix,
16    flash_W,
17    W_measure,
18    write_W_file,
19    open_W_file_write,
20    write_body_W,
21    flash_U,
22    U_measure,
23    write_U_file,
24    open_U_file_write,
25    write_body_U,
26    adjs_algorithm_check_NU,
27    reset_variables_for_algorithm,
28    calculate_actual_parameters,
29    reset_alarms,
30    set_adj_to_lamp,
31    real_time_simulation,
32    close_U_file_write
33 );
34 END_TYPE

```

*Figura 4.10: codice sorgente TwinCat, lista degli stati*

#### 4.5.1 Rappresentazione della macchina a stati e codice TwinCAT

Le sei classi descritte nel paragrafo precedenti sono rappresentate con un diagramma in Figura 4.11.

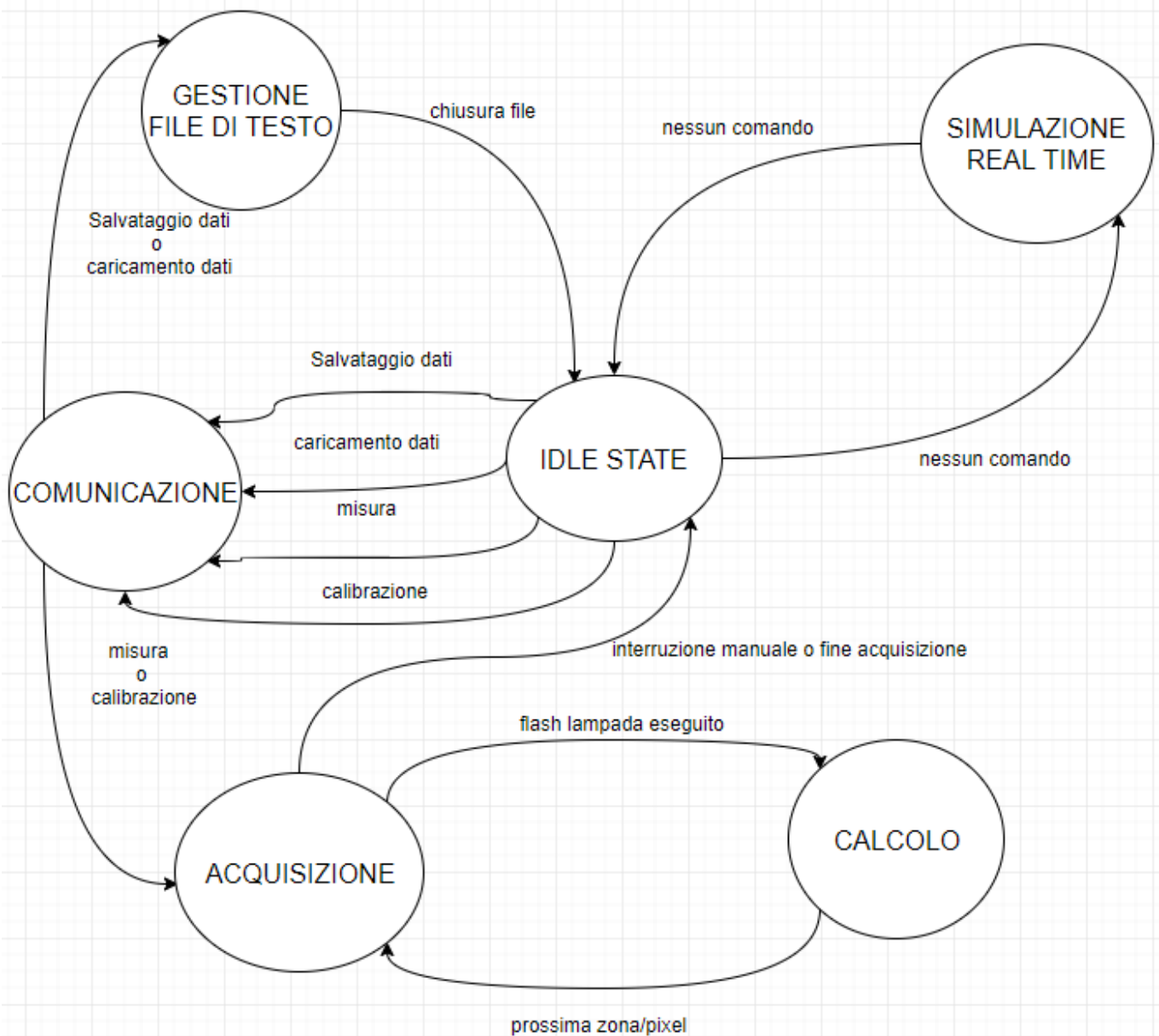


Figura 4.11: diagramma a stati dell'implementazione TwinCAT della procedura di calibrazione di uniformità

Per semplicità visiva il diagramma mostra solo le sei classi di stati, tuttavia quelli relativi a gestione file di testo, acquisizione e calcolo sono diversificati a seconda che l'utente desideri procedere con una misura di uniformità oppure con una calibrazione.

In particolar modo, durante la fase di misura viene aggiornato solamente il file contenente il vettore  $\mathbf{U}$ , mentre in fase di calibrazione viene aggiornato anche il file contenente la matrice  $\mathbf{W}$ .

Un'altra differenza è quella relativa alla gestione della lampada: per effettuare una misura di uniformità la lampada si accende attivando contemporaneamente tutte le venti zone, mentre durante un ciclo di calibrazione le zone vengono accese sequenzialmente e in maniera ordinata.

In Appendice A si riporta l'intero listato scritto in ambiente TwinCat.

#### 4.5.2 L'interfaccia grafica per l'utente

Per permettere all'utente un utilizzo chiaro e intuitivo della funzione di calibrazione dell'uniformità spaziale è stata creata un'apposita sezione all'interno dell'interfaccia grafica.

In Figura 4.12 viene riportata la struttura adottata.



Figura 4.12: interfaccia grafica per l'utente per la funzione di calibrazione di uniformità spaziale

La pagina risulta essere divisa in due parti: a sinistra sono presenti le informazioni relative ad una semplice misura di uniformità, mentre nella zona di destra sono riportate le informazioni per la calibrazione e l'abbassamento del parametro NU.

I riquadri colorati rappresentano i venti pixel sui quali viene misurato il valore di irradianza. Per facilità visiva viene usata una scala di colori per rappresentare le zone più o meno illuminate (blu per le zone fredde, rosso per le zone calde).



La matrice presente sotto la dicitura “adjustment”, invece, rappresenta i fattori di aggiustamento in percentuale. In particolar modo la matrice presente a sinistra indica i valori attualmente impostati nella lampada, la matrice di destra, invece, è editabile e riporta il risultato della calibrazione. Per ogni variazione dei fattori presenti nella sezione di destra è possibile osservare l’effetto simulato della misura di irradianza sui pixel in real-time.

Gli altri paramentri visibili sono l’irradianza media letta sull’area di misura e il parametro di non uniformità, misurati e simulati.

Infine, sono riportati i comandi dedicati al caricamento e al salvataggio dei dati, allo start della procedura di misura e calibrazione. Una finestra contiene i messaggi per comunicare con l’utente, riportando eventuali errori della procedura o conferme di salvataggio/caricamento dati.

Una volta premuti i bottoni “uniformity calibration” o “Uniformity measurement” compare una finestra di dialogo riportata in Figura 4.13.

Tale finestra indica all’utente dove posizionare la cella di riferimento via via che si procede alla lettura di irradianza sui vari pixel, fornendo inoltre la possibilità di interrompere la procedura.

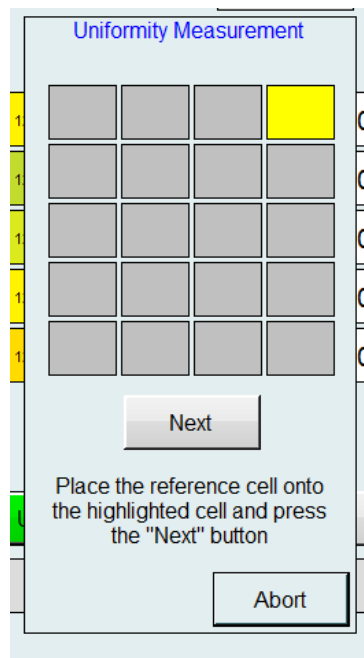


Figura 4.13: Finestra di dialogo per lo spostamento della cella di riferimento

## 4.6 Test e dati raccolti

Per validare la procedura di calibrazione di uniformità spaziale sono stati eseguiti vari test, con la finalità di verificare che la procedura automatica è effettivamente in grado di abbassare il valore di non uniformità (NU).

In questo paragrafo vengono riportate quattro prove, effettuate in condizioni diverse tra loro.

Le condizioni si diversificano in:

- Estensione dell'area di misura
- Distanza tra cella di riferimento e piano dei LED della lampada

In Tabella 4.7 si riportano le condizioni per le quattro prove.

	Test 1	Test 2	Test 3	Test 4
<b>Area di misura [mm x mm]</b>	240x160	240x180	260x180	260x180
<b>Distanza tra cella di riferimento e piano dei LED [mm]</b>	260	260	305	320

*Tabella 4.7: caratteristiche delle prove per uniformità spaziale*

I primi due test sono accomunati dalla stessa distanza tra cella di riferimento e piano dei LED della lampada. Tuttavia, l'estensione dell'area di misura risulta leggermente inferiore per il primo test.

Il terzo e quinto test presentano un'area più estesa rispetto ai primi due. Inoltre, la cella di riferimento è stata allontanata dalla lampada.

La scelta di variare area e distanza è dovuta al fatto che l'uniformità spaziale dipende fortemente da questi due parametri: all'aumentare delle due grandezze il valore NU aumenta progressivamente, come visibile in Tabella 4.8.

		Misura pre-calibrazione	Simulazione	Misura post calibrazione
<b>Test 1</b>	NU [%]	1,06	0,68	0,69
	Irradianza media [W/m <sup>2</sup> ]	1211,46	1187,10	1187,18
<b>Test 2</b>	NU [%]	1,14	0,60	0,60
	Irradianza media [W/m <sup>2</sup> ]	1240,84	1215,85	1215,92
<b>Test 3</b>	NU [%]	3,21	2,58	2,60
	Irradianza media [W/m <sup>2</sup> ]	1167,44	1169,45	1169,43
<b>Test 4</b>	NU [%]	3,81	1,37	1,35
	Irradianza media [W/m <sup>2</sup> ]	1213,93	1201,13	1200,23

*Tabella 4.8: risultati delle prove per uniformità spaziale*

Analizzando la tabella è possibile effettuare alcune considerazioni:

- Il valore di non uniformità NU cresce effettivamente all'aumentare dell'area di misura e della distanza tra cella di riferimento e piano dei LED
- Per tutti i test la procedura di calibrazione è stata in grado di diminuire sensibilmente il valore NU
- Per tutti i test il valore di irradianza media post calibrazione è minore rispetto a quello di partenza: ciò è dovuto al fatto che i fattori di aggiustamento imponibili appartengono all'intervallo [-10,0]. Ciò significa che la lampada globalmente riduce la sua potenza di emissione.
- La differenza tra valori simulati durante la fase di calibrazione e valori misurati dopo la calibrazione è minima. Dunque, l'algoritmo di simulazione è in grado di prevedere ottimamente i valori di irradianza.

## CONCLUSIONI

La procedura di calibrazione implementata per la lampada del simulatore solare risulta affidabile.

Sebbene non siano state seguite le indicazioni fornite da normativa, le informazioni fornite dalla procedura risultano essere un buon indice delle prestazioni della lampada in tema di uniformità spaziale.

In fase di produzione, presso un cliente, la procedura di calibrazione e misura descritta in questo e nel precedente capitolo può essere sfruttata come mezzo di indagine per eventuali malfunzionamenti o degradi dei LED.

Tramite l'interfaccia grafica il suo utilizzo risulta essere sufficientemente semplice: una possibile miglioria del sistema può essere quella di sfruttare un sistema meccanico per lo spostamento della cella di riferimento.

Tuttavia, nonostante l'incertezza introdotta dalla fase di spostamento manuale della cella di riferimento, i dati presentano una buona ripetibilità.

Un altro aspetto da considerare è il margine di miglioramento delle prestazioni: attualmente la calibrazione sfrutta uno scostamento basso della corrente di alimentazione dei LED ( da -10% a 0%) rispetto al valore nominale. Aumentando tale intervallo è possibile ottenere miglioramenti significativi.



# **CAPITOLO 5: PROCEDURA DI CALIBRAZIONE SPETTRALE DELLA LAMPADA**

## **5.1 Introduzione**

In questo capitolo è stato affrontato il problema di identificare una procedura in grado di correggere l'intensità dello spettro di emissione della lampada, a fronte di un eventuale degrado dei LED.

Come già descritto nel quarto capitolo, il degrado dello spettro può essere causato da diversi fattori: usura della lampada, rotture e malfunzionamenti, controllo della temperatura dei substrati non adeguato.

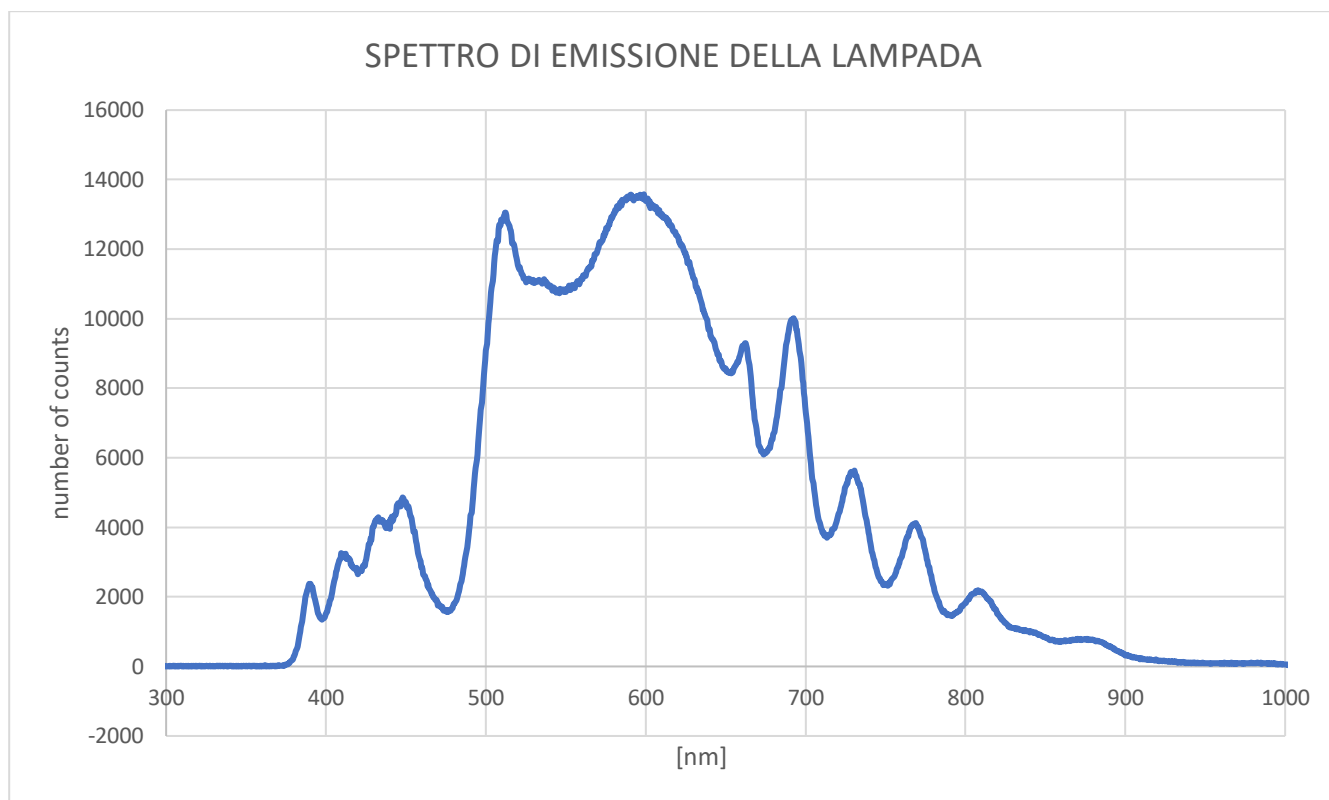
La procedura di calibrazione dello spettro descritta e implementata in questo capitolo si prefigge lo scopo di mantenere una forma dello spettro quanto più simile possibile alle condizioni di fabbrica della lampada.

Per essere in grado di fornire una procedura valida è stato necessario, come per il tema dell'uniformità spaziale, identificare un adeguato setup sperimentale per la raccolta dei dati. Oltre a ciò si è dovuto formulare il problema con un adeguato formalismo matematico.

Infine, è stato necessario scegliere un oggetto in grado di modulare l'intensità di emissione dei LED con il quale procedere alla correzione dello spettro.

## **5.2 Setup sperimentale e dati raccolti**

In Figura 5.1 si riporta la forma dello spettro di emissione della lampada.



*Figura 5.1: spettro della lampada del simulatore solare*

Lo spettro riportato è stato acquisito tramite uno spettrometro Ocean HR 2000 collegato ad un cavo di fibra ottica posto in direzione perpendicolare al piano dei LED della lampada.

Ciò che la procedura di correzione si prefigge, è di mantenere quanto più simile lo spettro della lampada a quello di Figura 5.1.

Tuttavia, per modulare l'intensità dello spettro non è immaginabile scegliere di sfruttare uno spettrometro: ciò comporterebbe l'obbligo da parte del cliente di dotarsi del necessario hardware e software necessario all'acquisizione.

Lo strumento scelto per effettuare le misure quantitative dello spettro è rappresentato dal set di dieci fotodiodi situati tra moduli adiacenti di LED della lampada: ogni photodiode restituisce un valore digitale a 12 bit proporzionale all'intensità della luce rilevata.

Il setup sperimentale risulta dunque relativamente semplice: viene posto uno specchio riflettente sotto la lampada in maniera tale che esso giaccia su un piano parallelo a quello dei LED della lampada. Tale specchio viene posizionato, inoltre, ad un'altezza tale per cui ogni apertura presente del riflettore venga eliminata.

In tale maniera, quando la lampada non è accesa, si viene a creare una camera oscura dentro la quale non può penetrare la luce ambientale esterna.

Scegliendo dunque una determinata lunghezza d'onda, tramite il valore digitale restituito in dai photodiodi, è possibile quantificare l'intensità luminosa.

Per quel che riguarda, invece, la modulazione dell'intensità luminosa, la scelta effettuata è stata nuovamente quella di sfruttare i coefficienti di aggiustamento locali della corrente di alimentazione dei LED.

Tuttavia, a differenza del caso dell'uniformità spaziale, per la calibrazione dello spettro vengono modificati tutti i 340 fattori singolarmente: in questa maniera è possibile incrementare o decrementare ogni singola catena di LED per ogni singola zona della lampada.

Affinchè si possa procedere con lo studio di un algoritmo per correggere lo spettro, è necessario inoltre verificare il buon funzionamento del setup. In particolar modo risulta indispensabile:

- Ricavare un parametro che quantifichi la ripetibilità della lettura dei fotodiodi;
- Verificare che ad una variazione dei fattori di aggiustamento corrisponda una proporzionale variazione del valore digitale fornito dai fotodiodi.
- 

#### 5.2.1 Ripetibilità della lettura dei fotodiodi

Per quantificare il valore di ripetibilità sono state effettuate 300 accensioni sequenziali della lampada, dopo un periodo di stabilizzazione termica di qualche minuto.

In Figura 5.2 viene riportato l'andamento di 5 fotodiodi durante l'arco del test.

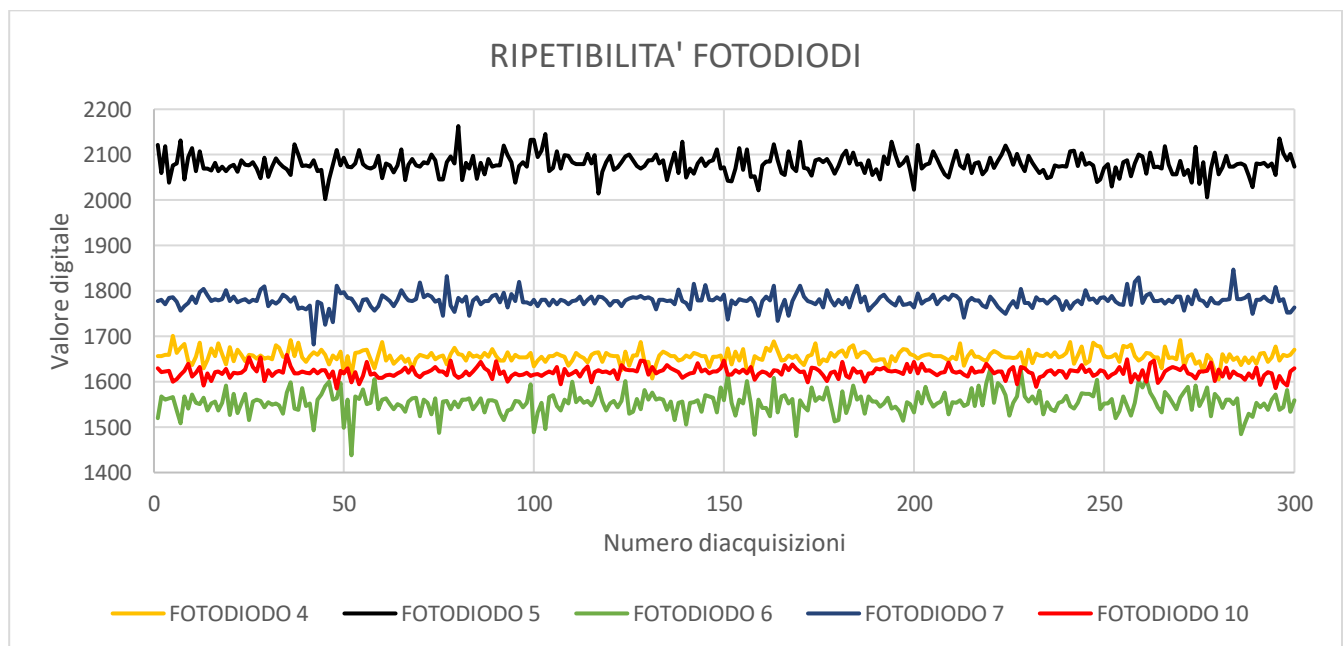


Figura 5.2: andamento di 5 fotodiodi durante le acquisizioni

Una considerazione che si può fare leggendo il grafico è che la lettura dei fotodiodi sul piano dei LED non è uniforme. Il fotodiodo numero 5, ad esempio, rileva un'intensità corrispondente a circa 2100, contro il valore 1550 restituito dal fotodiodo numero 6.

Ciò può essere dovuto a vari motivi, tra cui la conformazione geometrica del riflettore: fotododi posti in zone più centrali ricevono una porzione riflessa di luce maggiore rispetto ai sensori posti negli angoli. Un'altra motivazione può essere data dalla responsività dei fotodiodi stessi: essa non è conosciuta e nulla garantisce che sia la stessa per tutti i 10 sensori.

Per quantificare la ripetibilità in lettura dei fotodiodi si è scelto di considerare la quantità:

$$\Delta = \frac{F_{\max} - F_{\min}}{F_{\text{avg}}} \times 100 \quad (5.1)$$

$F_{\max}$ ,  $F_{\min}$  e  $F_{\text{avg}}$  rappresentano rispettivamente il valore massimo, minimo e medio restituito dai fotodiodi nell'arco delle 300 acquisizioni.

In Tabella 5.1 si riportano i valori di ripetibilità per tutti i fotodiodi.

	<b>MAX</b>	<b>MIN</b>	<b>AVG</b>	<b><math>\Delta</math> [%]</b>
<b>F1</b>	1681	1578	1632,0	6,3
<b>F2</b>	1758	1562	1681,1	11,7
<b>F3</b>	1646	1572	1606,1	4,6
<b>F4</b>	1701	1605	1655,7	5,8
<b>F5</b>	2163	2002	2078,7	7,7
<b>F6</b>	1627	1438	1554,3	12,2
<b>F7</b>	1847	1682	1778,7	9,3
<b>F8</b>	1743	1613	1690,5	7,7
<b>F9</b>	1724	1514	1632,9	12,9
<b>F10</b>	1659	1585	1620,7	4,6

*Tabella 5.1: valori di ripetibilità in lettura dei fotodiodi*

Dai valori riportati in tabella si nota che il comportamento dei fotodiodi non è riassumibile tramite un unico valore: alcuni fotodiodi presentano un buon livello di ripetibilità (come il numero 1,3 o 10). Altri, invece, tendono a discostarsi maggiormente dal valore medio durante l'arco delle acquisizioni (come il numero 2,6 o 9).

Dai dati appena riportati si può concludere che la procedura di correzione dello spettro deve tener conto di questo comportamento: risulta necessario in particolar modo filtrare o quantomeno mediare i valori digitali restituiti dai 10 sensori.



### 5.2.2 Relazione lineare tra valori digitali e fattori di aggiustamento

La finalità di questo paragrafo è quella di verificare l'esistenza di una relazione lineare tra i fattori di aggiustamento locali e il valore digitale restituito dai fotodiodi.

A tal proposito si è eseguito il seguente test: sono stati raccolti i valori letti dai fotodiodi per 500 acquisizioni; successivamente è stato impostato un valore di +18% e +36% per i fattori di aggiustamento locali. A questo punto si è proceduto ad acquisire nuovamente i valori digitali dei fotodiodi per i due diversi livelli di corrente di alimentazione dei LED.

In Figura 5.3 si riporta il risultato del test per gli stessi fotodiodi riportati in Figura 5.2.

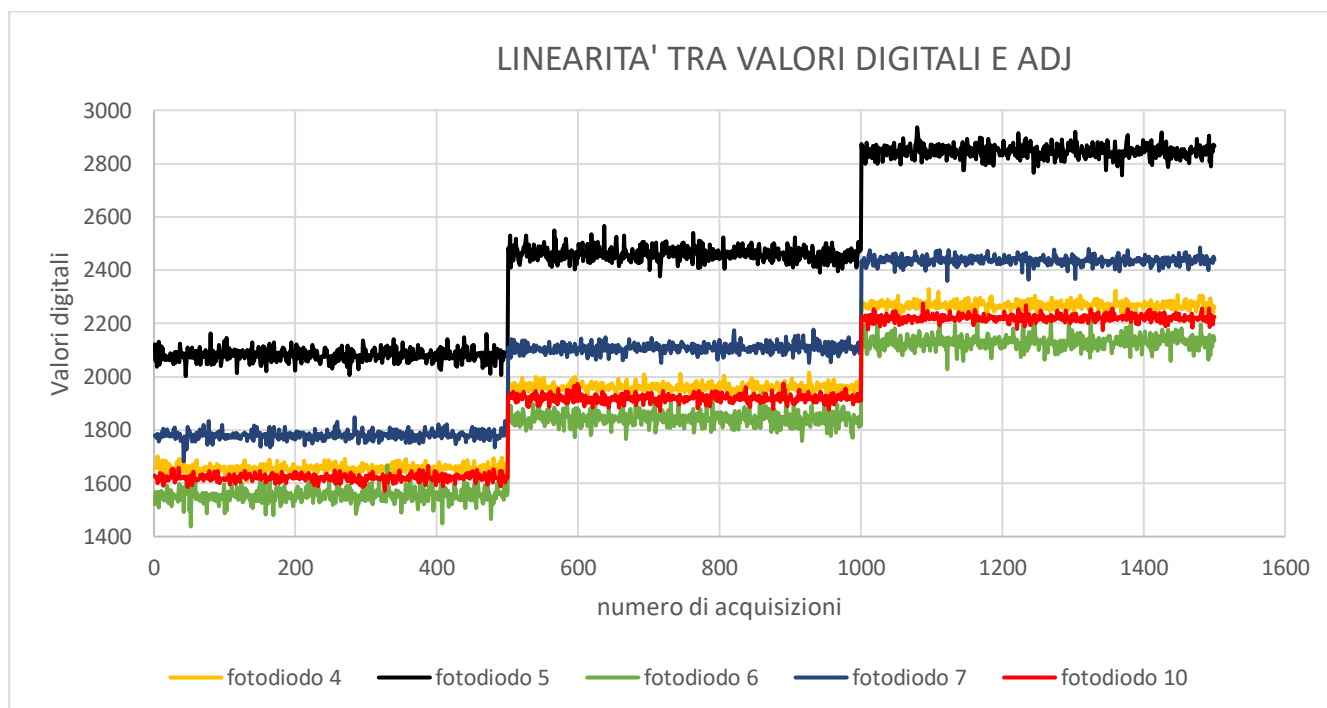


Figura 5.3: andamento di 5 fotodiodi durante il test per la linearità

A meno del rumore già quantificato nello scorso paragrafo, dal grafico si osserva chiaramente l'andamento a gradino dei fotodiodi dovuto all'impostazione dei tre diversi valori per i fattori di aggiustamento: 0%, +18%, +36%.

Per un'analisi più quantitativa si riportano in Tabella 5.2 i tre valori medi corrispondenti ai tre diversi fattori di aggiustamento, per ogni fotodiode. A partire dai valori medi è stata calcolato lo scostamento in percentuale rispetto al valore medio nominale corrispondente a 0%.

	Letture a 0%	Letture a +18%	Scostamento calcolato	Letture a +36%	Scostamento calcolato
<b>F1</b>	1631,34	1931,93	18,43	2234,57	36,98
<b>F2</b>	1680,88	1990,98	18,45	2302,09	36,96
<b>F3</b>	1606,27	1901,96	18,41	2198,32	36,86
<b>F4</b>	1656,14	1960,41	18,37	2268,21	36,96
<b>F5</b>	2078,93	2461,32	18,39	2845,78	36,89
<b>F6</b>	1555,72	1843,13	18,47	2131,51	37,01
<b>F7</b>	1779,72	2107,34	18,41	2436,24	36,89
<b>F8</b>	1690,59	2000,59	18,34	2313,79	36,86
<b>F9</b>	1632,86	1934,55	18,48	2237,56	37,03
<b>F10</b>	1621,41	1919,41	18,38	2220,54	36,95

Tabella 5.2: linearità tra valori letti dai fotodiodi e fattori di aggiustamento

Dai valori riportati in tabella si può concludere che i fotodiodi rispondono linearmente ad una variazione della corrente di alimentazione dei LED della lampada: grazie a ciò, è possibile procedere alla costruzione di una funzione adatta alla correzione dello spettro della lampada.

### 5.3 Approccio matematico al problema

Dopo aver verificato la linearità tra corrente di alimentazione dei LED della lampada (fattori di aggiustamento) e valori digitali restituiti dai fotodiodi, è possibile ricavare una semplice relazione che permetta di correggere lo spettro a fronte di degradi o malfunzionamenti.

Per far ciò risulta sufficiente confrontare un valore di riferimento digitale con il valore corrispondente in fase di correzione, calcolare lo scarto percentuale tra i due valori e sommare tale percentuale all'attuale fattore di aggiustamento locale. A tal proposito si osservi l'equazione (5.2).

$$ADJ_{corrected} = ADJ + \left( \frac{F_{ref} - F_{read}}{F_{ref}} \right) \times 100 \quad (5.2)$$

I simboli utilizzati assumono i seguenti significati:

- $ADJ_{corrected}$  è il singolo fattore di aggiustamento corretto dopo la fase di calibrazione;
- $ADJ$  è il singolo fattore di aggiustamento presente nel dizionario della lampada durante la fase di acquisizione;
- $F_{ref}$  è il valore digitale del singolo fotodiodo scelto come target;
- $F_{read}$  è il valore digitale del singolo fotodiodo letto durante la fase di calibrazione.

Va precisato che durante la fase di calibrazione l'equazione (5.2) deve essere applicata ad ogni singola zona della lampada e ad ogni singola catena di LED. Dunque,  $ADJ_{corrected}$  e  $ADJ$  rappresentano gli elementi di due matrici di dimensioni 20x17.

Tutti gli elementi Fref assumono la funzione di target per la calibrazione: per ogni lunghezza d'onda e per ogni zona della lampada deve essere presente un valore digitale pre-acquisito che rappresenti la porzione di spettro che si vuole garantire tramite la procedura di correzione. Perciò, l'insieme di questi valori digitali rappresenta in modo equivalente la forma e l'intensità dello spettro della lampada in condizioni di fabbrica.

Come analizzato nel paragrafo 5.2.1, tuttavia, la lettura dell'intensità luminosa da parte dei fotodiodi è caratterizzata da diversi fattori di ripetibilità.

Applicare l'equazione (5.2) ai valori digitali dei fotodiodi senza alcun tipo di filtraggio risulta impraticabile. Si consideri, a titolo di esempio, la Tabella 5.3.

	<b>Fref</b>	<b>Fread</b>	<b>ADJcorrected [%]</b>
<b>F1</b>	1639	1642	0
<b>F2</b>	1722	1679	2
<b>F3</b>	1572	1611	-2
<b>F4</b>	1656	1657	0
<b>F5</b>	2122	2060	3
<b>F6</b>	1520	1567	-3
<b>F7</b>	1778	1781	0
<b>F8</b>	1717	1672	3
<b>F9</b>	1594	1630	-2
<b>F10</b>	1629	1621	0

*Tabella 5.3: applicazione dell'equazione (5.2) senza filtraggi*

I dati riportati in Tabella 5.3 rappresentano i risultati del seguente test:

1. Per una determinata zona e lunghezza d'onda sono stati acquisiti i dieci valori digitali dei fotodiodi. Tali valori sono stati considerati come gli elementi target Fref;
2. Senza apportare alcuna modifica al sistema e mantenendo gli stessi fattori di aggiustamento (tutti nulli) si è ripetuta l'acquisizione descritta del punto 1: tali valori sono stati considerati come gli elementi letti durante la fase di calibrazione (Fread);
3. È stata applicata l'equazione (5.2) ottenendo i dieci corrispondenti valori di aggiustamento (ADJcorrected).

Ciò che ci si aspettava era di ottenere un set di dieci fattori di aggiustamento corretti tutti nulli: idealmente, non avendo apportato alcuna modifica al sistema, i valori Tref e Tread dovrebbero coincidere per ogni fotodiodo.

Dunque:

$$ADJ_{corrected} = ADJ + \left( \frac{F_{ref} - F_{read}}{F_{ref}} \right) \times 100 = ADJ + 0 = 0 \quad (5.3)$$

Invece, come si evince dalla tabella, a causa della rumorosità in lettura dei fotodiodi solamente quattro valori su dieci risultano essere nulli (dopo aver effettuato l'arrotondamento delle cifre decimali all'unità).

Una situazione di questo tipo non risulta accettabile come risultato della calibrazione.

Si è deciso dunque di sostituire i valori  $F_{ref}$  e  $F_{read}$  con una semplice media aritmetica di acquisizioni consecutive e ripetere il test descritto precedentemente. I risultati vengono riportati in Tabella 5.4, dove  $T_{ref}$  e  $T_{read}$  rappresentano la media di 20 acquisizioni consecutive.

	<b>Fref</b>	<b>Fread</b>	<b>ADJcorrected [%]</b>
<b>F1</b>	1627,75	1635,35	0
<b>F2</b>	1689,85	1680,55	1
<b>F3</b>	1600,35	1606,05	0
<b>F4</b>	1662,15	1659,60	0
<b>F5</b>	2080,55	2077,35	0
<b>F6</b>	1551,15	1554,20	0
<b>F7</b>	1781,25	1779,20	0
<b>F8</b>	1694,75	1685,60	1
<b>F9</b>	1636,05	1630,10	0
<b>F10</b>	1617,80	1626,25	-1

Tabella 5.4: applicazione dell'equazione (5.2) con media aritmetica su 20 acquisizioni

Si nota che i risultati migliorano sensibilmente anche solo tramite una semplice media aritmetica: sette fattori di aggiustamento su dieci risultano essere correttamente nulli. I tre rimanenti, invece, non superano in valore assoluto l'1%.

Effettuando la scelta di mantenere un numero pari a 20 acquisizioni consecutive per effettuare la media dei valori digitali dei fotodiodi, è possibile riportare la relazione che permette di effettuare la correzione dello spettro tramite i fattori di aggiustamento:

$$ADJ_{corrected} = ADJ + \left( \frac{\sum_{i=1}^{20} F_{ref_i} - \sum_{i=1}^{20} F_{read_i}}{\sum_{i=1}^{20} F_{ref_i}} \right) \times 100 \quad (5.4)$$

## 5.4 Implementazione in ambiente TwinCAT

### 5.4.1 L'algoritmo di correzione

L'algoritmo di correzione dello spettro della lampada è suddiviso in due fasi distinte e separate temporalmente:

1. La prima fase è costituita dalla raccolta dei dati necessari alla formazione di uno spettro da considerare come target della calibrazione. Questi dati devono essere raccolti nella fase iniziale di vita della lampada, quando tutte le catene di LED risultano non degradate. Tramite l'acquisizione dei valori digitali dei fotodiodi, si dispongono di tutte le informazioni per procedere ad un'eventuale futura fase di correzione;
2. La seconda fase avviene nel momento in cui il buon funzionamento dei LED della lampada non è più garantito. Non risulta fattibile stimare con precisione un arco temporale entro il quale lo spettro emesso della lampada degrada di una certa quantità. Durante questa fase viene ripetuta l'acquisizione dei valori digitali dei fotodiodi: tali valori vanno confrontati con quelli raccolti al punto 1. Successivamente, tramite l'applicazione della relazione (5.4), si procede alla variazione dei fattori di aggiustamento della lampada.

Da quanto detto, si intuisce che l'algoritmo atto all'acquisizione dei valori digitali dei fotodiodi risulta essere lo stesso per i due punti appena descritti. Oltre a ciò, è importante verificare che durante le due fasi le condizioni di setup del simulatore siano le medesime: fattore di aggiustamento globale, posizione dei riflettori, posizione e altezza dello specchio.

In Figura 5.4 viene riportato il diagramma di flusso relativo all'acquisizione dei valori digitali dei fotodiodi.

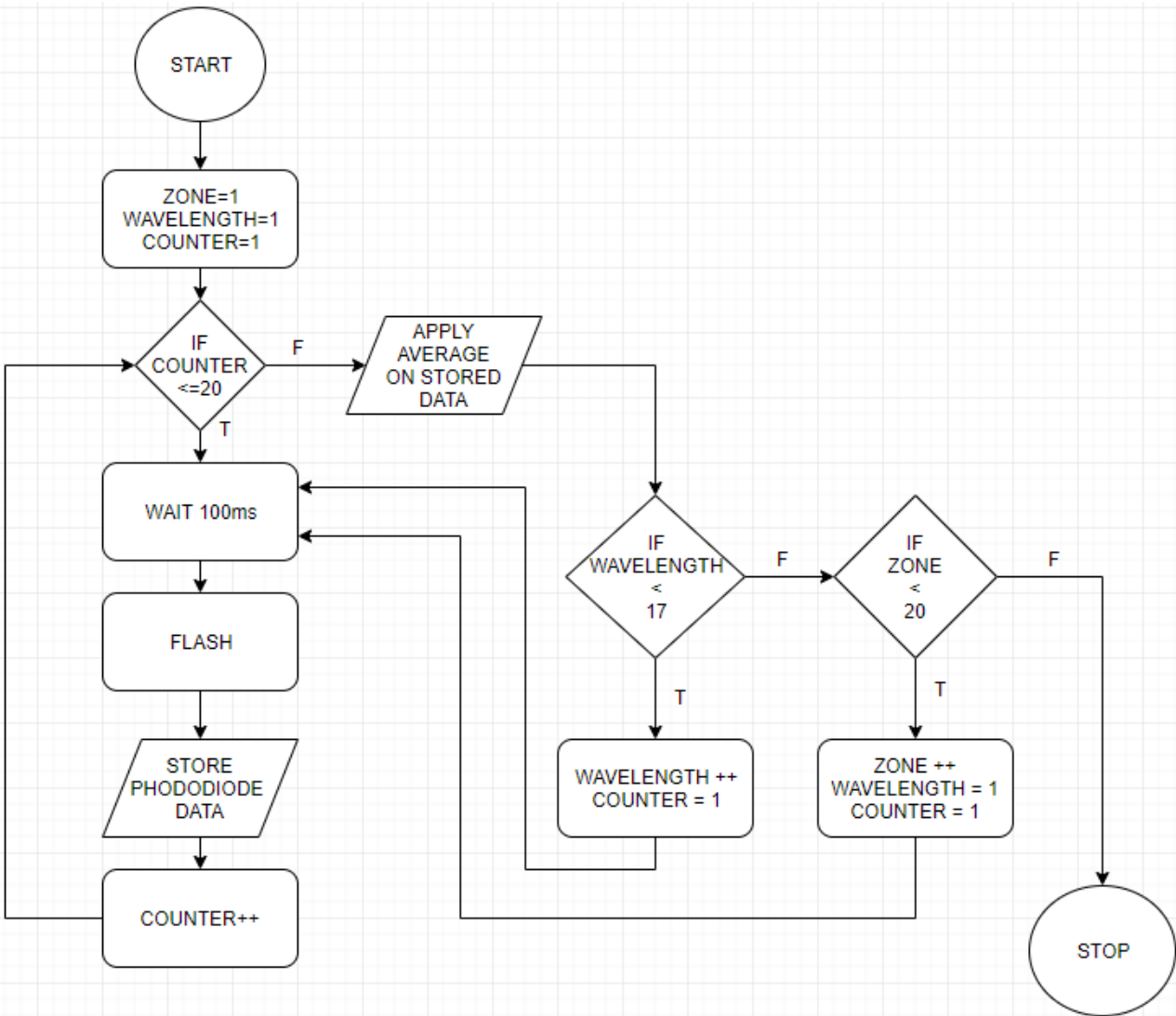


Figura 5.4: diagramma di flusso per acquisizione valori digitali dei fotodiodi

All'interno della procedura è stato inserito un blocco di timeout pari a 100ms. Questo timeout permette alla lampada di stabilizzarsi tra un flash e il successivo.

La fase di acquisizione dei dati risulta relativamente lunga. Considerando che vengono effettuati 20 flash per ognuna delle 17 lunghezze d'onda e per ognuna delle 20 zone della lampada separati da un ritardo di 100ms, con durata di flash pari ad altrettanti millisecondi, si può scrivere:

$$t [s] = (0.1 [ms] + 0.1 [ms]) \times 17 \times 20 \times 20 = 1360 [s] \quad (5.5)$$

Dunque la fase di flash e raccolta dati impiega circa 23 minuti per essere effettuata.

Considerando, tuttavia, che la correzione dello spettro della lampada è un'operazione da eseguire sporadicamente (ovvero con frequenza annuale), perde di significato il tempo necessario alla procedura per terminare.

#### 5.4.2 Rappresentazione della macchina a stati

Per una completa definizione della procedura di correzione dello spettro, risulta necessario definire degli stati con i quali gestire i dati raccolti, effettuare i calcoli necessari alla calibrazione, comunicare con il dizionario degli oggetti della lampada.

L'intera procedura di correzione dello spettro conta 46 stati, suddivisi in 7 categorie:

- Idle state (1): in questo stato vengono resettate le variabili utilizzate in acquisizione, il sistema rimane fermo in attesa di istruzioni;
- Gestione della lampada (12): in questo set di di stati avviene la comunicazione con la lampada del simulatore. Vengono letti i fattori di aggiustamento locali presenti, viene impostata la lunghezza del flash, viene letto il fattore di aggiustamento globale. Vengono inoltre impostati i contatori relativi a lunghezza d'onda e zona della lampada, fissati inizialmente a valore unitario;
- Acquisizione dello spettro target (7): si esegue la procedura riportata in Figura 5.4. la lampada viene accesa per ogni lunghezza d'onda e per ogni zona 20 volte. Si registrano i valori digitali letti dai fotodiodi e viene eseguita la media aritmetica sulle 20 acquisizioni;
- Scrittura file di calibrazione o del report di calibrazione (12): viene creato un file di testo che riporta le informazioni relative all'acquisizione dello spettro target. In particolar modo vengono scritti data e ora dell'acquisizione, i fattori di aggiustamento locali e globale con il quali sono state effettuate le acquisizioni. Infine, vengono riportati tutti i valori digitali dei fotodiodi. Previo comando, gli stessi stati possono essere usati per la creazione di un file report al termine della procedura di correzione. Tale file contiene le stesse informazioni appena elencate;
- Acquisizione e correzione dello spettro (7): in questo insieme di stati si procede come per l'acquisizione dello spettro target;
- Lettura file di calibrazione (5): viene aperto e letto il file di testo di calibrazione. In particolar modo vengono confrontati i valori digitali e viene calcolato il fattore di aggiustamento locale tramite l'equazione (5.4);
- Scrittura su lampada (2): vengono scritti sul dizionario della lampada i 340 fattori di aggiustamento dopo la fase di correzione. Nel caso in cui lo spettro non risulti degradato, vengono ripristinati i fattori di aggiustamento pre-esistenti.

In Figura 5.5 si riporta un diagramma di flusso contenente le sette classi di stati appena descritte.

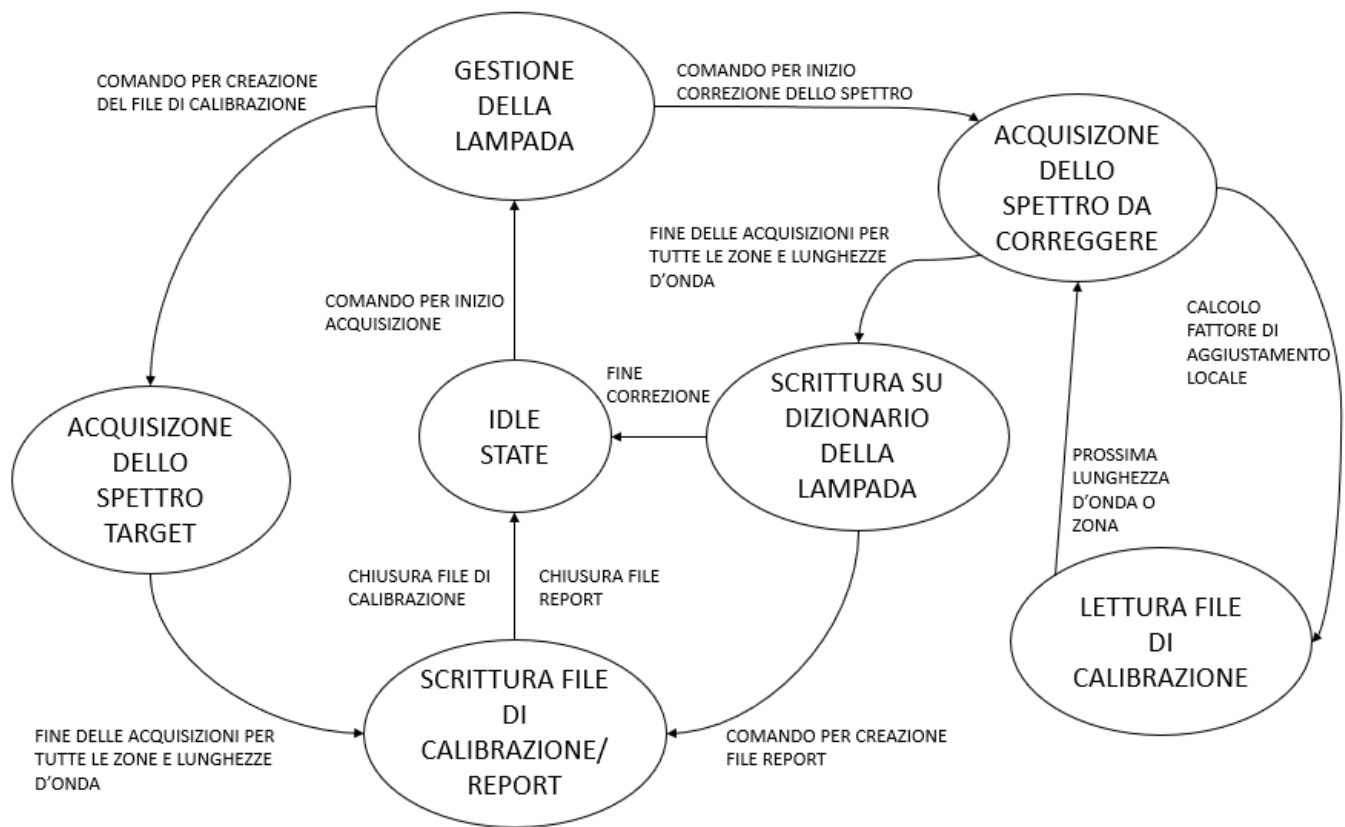


Figura 5.5: diagramma degli stati per la procedura di correzione tramite fotodiodi

Risultano necessarie alcune precisazioni.

Per quel che riguarda il confronto dei valori dei fotodiodi, non tutti i valori registrati risultano necessari.

I valori utilizzati per il confronto in fase di correzione fanno riferimento al fotodiodo più vicino alla zona accesa: questa scelta è stata effettuata poiché per tale fotodiodo si hanno meno riflessioni della luce e il percorso luminoso risulta diretto. A conferma di tale affermazione si registrano valori generalmente più elevati per fotodiodi posti in vicinanza delle zone accese. Si riporta in figura 5.6 una sezione del file di testo di calibrazione.

Si può notare che il fotodiodo numero 1, posto esattamente a metà tra la zona 1 e la zona 2, presenta valori generalmente più elevati degli altri nove fotodiodi.



Date: 2019.09.19 - 13:03:35

-- Test Settings --

Global Intensity[0.1%]: 0.0

Zone_1 Wavelength[%]:	1.0	8.0	1.0	1.0	1.0	1.0	-3.0	-2.0	1.0	1.0	1.0	1.0
Zone_2 Wavelength[%]:	1.0	2.0	-1.0	0.0	0.0	0.0	-3.0	-1.0	-1.0	0.0	-1.0	0.0
Zone_3 Wavelength[%]:	-12.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	0.0	0.0
Zone_4 Wavelength[%]:	-6.0	-2.0	1.0	0.0	0.0	0.0	5.0	2.0	-2.0	0.0	0.0	0.0
Zone_5 Wavelength[%]:	8.0	2.0	0.0	0.0	0.0	0.0	-4.0	0.0	0.0	0.0	0.0	0.0
Zone_6 Wavelength[%]:	6.0	1.0	0.0	0.0	0.0	0.0	1.0	2.0	3.0	0.0	0.0	0.0
Zone_7 Wavelength[%]:	0.0	-5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Zone_8 Wavelength[%]:	1.0	5.0	3.0	3.0	3.0	3.0	2.0	5.0	3.0	3.0	2.0	3.0
Zone_9 Wavelength[%]:	-2.0	1.0	1.0	0.0	0.0	0.0	3.0	2.0	0.0	0.0	-1.0	0.0
Zone_10 Wavelength[%]:	0.0	2.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	0.0
Zone_11 Wavelength[%]:	1.0	1.0	1.0	0.0	0.0	0.0	-1.0	0.0	-2.0	0.0	0.0	0.0
Zone_12 Wavelength[%]:	2.0	-1.0	0.0	0.0	0.0	0.0	-2.0	1.0	-1.0	0.0	0.0	0.0
Zone_13 Wavelength[%]:	-15.0	-14.0	-10.0	-10.0	-10.0	-10.0	-9.0	-10.0	-10.0	-10.0	-11.0	-11.0
Zone_14 Wavelength[%]:	-4.0	2.0	0.0	0.0	0.0	-1.0	5.0	0.0	0.0	0.0	0.0	0.0
Zone_15 Wavelength[%]:	5.0	6.0	1.0	0.0	0.0	0.0	1.0	1.0	-1.0	0.0	1.0	0.0
Zone_16 Wavelength[%]:	-4.0	-2.0	0.0	0.0	0.0	0.0	3.0	-2.0	2.0	0.0	-1.0	0.0
Zone_17 Wavelength[%]:	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
Zone_18 Wavelength[%]:	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0
Zone_19 Wavelength[%]:	1.0	5.0	0.0	0.0	0.0	0.0	3.0	1.0	0.0	0.0	0.0	0.0
Zone_20 Wavelength[%]:	-14.0	-15.0	-15.0	-15.0	-15.0	-15.0	-15.0	-14.0	-15.0	-15.0	-14.0	-14.0

-- Test Result --

ZoneN	Wavelength	Photo1	Photo2	Photo3	Photo4	Photo5	Photo6	Photo7	Photo8	Photo9	Photo10
1	1	436	308	354	295	354	267	329	317	346	278
1	2	843	672	1273	1208	1303	1191	715	691	692	658
1	3	779	272	272	271	274	252	964	271	262	257
1	4	839	530	574	552	524	486	508	504	505	478
1	5	627	618	626	623	618	591	621	615	602	589
1	6	1274	572	597	607	601	561	588	592	561	554
1	7	877	1480	1514	1484	1450	1368	851	1445	821	1431
1	8	352	1153	1272	1205	1215	1091	1179	1141	330	307
1	9	499	438	1542	1599	1494	1491	408	1539	411	413
1	10	1297	905	949	970	914	867	902	899	909	860
1	11	1110	932	958	982	904	857	878	859	871	822
1	12	1233	1132	1149	1163	1102	1061	1122	1116	1111	1069
1	13	614	556	549	558	535	516	548	540	543	524
1	14	343	318	1312	1331	1251	1218	1253	1282	300	298
1	15	711	671	684	685	656	616	653	657	659	626
1	16	759	684	709	698	680	647	699	698	688	684
1	17	1274	1183	1248	1200	1201	1106	1219	1155	292	1134
2	1	424	355	318	325	336	292	333	349	333	341
2	2	927	808	1387	1285	1405	1282	789	807	772	728
2	3	738	296	280	283	277	269	973	294	277	271
2	4	786	600	598	610	543	538	532	565	528	537
2	5	653	631	637	634	631	611	629	628	627	602
2	6	1328	588	593	613	589	576	593	614	586	580

Figura 5.6: sezione del file di testo di calibrazione

Una seconda precisazione riguarda la risposta spettrale dei fotodiodi: non è stata effettuata alcuna considerazione su come ogni fotodiodo risponda alle differenti lunghezze d'onda.

Ciò è dovuto al fatto che i valori digitali vengono confrontati a coppie per gli stessi fotodiodi: l'applicazione dell'equazione (5.4) rende il risultato della correzione indipendente dalla risposta spettrale dei fotodiodi stessi.

Un'ultima precisazione riguarda l'output della fase di correzione dello spettro. A seguito delle considerazioni sul rumore in acquisizione effettuate nei precedenti paragrafi, si è scelto di impostare un valore di soglia per i fattori di aggiustamento calcolati.

Se un fattore di aggiustamento calcolato risulta essere inferiore al valore di soglia, allora si considera tale risultato come rumore, e il relativo fattore di aggiustamento non viene modificato.

Dunque, a titolo esemplificativo, se il fattore di aggiustamento di una relativa lunghezza d'onda e zona risulta essere nullo e dopo la fase di correzione la relazione (5.4) restituisce un valore pari a +1%, tale fattore non viene modificato. Il fattore di soglia attualmente impostato è pari a 3%.

In appendice B si riporta il listato completo scritto per implementare la funzione di correzione spettrale.

## 5.5 Test e dati raccolti

Per effettuare un test di correzione spettrale è necessario disporre di due elementi:

- Uno spettro target da utilizzare in fase di correzione
- Uno spettro corrotto a causa del degrado dei LED

Non avendo a disposizione una lampada che emettesse uno spettro degradato, si è deciso di simulare tale comportamento tramite i fattori di aggiustamento. In particolar modo, prima è stato acquisito lo spettro target con tutti i fattori di aggiustamento nulli. Successivamente, si è proceduto ad una modifica selettiva dei coefficienti di aggiustamento: riducendo, ad esempio, la corrente di alimentazione per determinate lunghezze d'onda, è stato simulato il degrado delle relative catene.

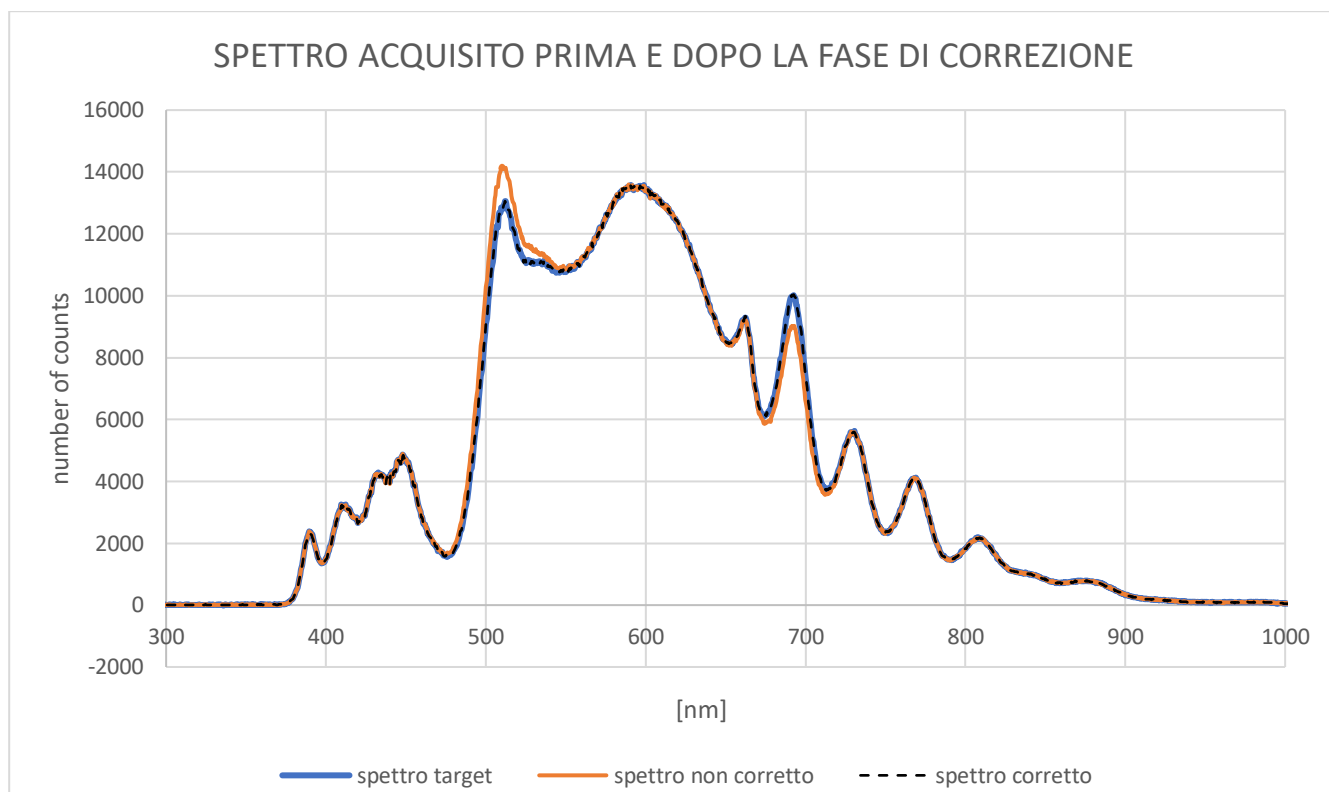
Per il primo test effettuato è stato sfruttato anche lo spettrometro Ocean Optics hr2000.

Il test si suddivide nelle seguenti sequenze:

1. Per tutte le zone è stato impostato un fattore di aggiustamento di +30% per la catena 9 (verde) e di -15% per la catena 6 (rosso);
2. È stato misurato lo spettro tramite spettrometro ed è stato creato il file di testo di calibrazione;
3. Tutti i fattori di aggiustamento sono stati annullati;
4. È stata eseguita la procedura di correzione spettrale;
5. È stato misurato nuovamente lo spettro tramite spettrometro.

Lo scopo del test è quello di verificare che la procedura è in grado di correggere in maniera autonoma lo spettro.

In Figura 5.7 si riportano gli andamenti spettrali acquisiti.



*Figura 5.7: spettro acquisito prima e dopo la correzione*

Si possono effettuare alcune considerazioni:

- Per la catena 9 (verde) la differenza tra lo spettro target e quello corretto è dello 0.37% ;
- Per la catena 6 (rosso) la differenza tra lo spettro target e quello corretto è del 0.63%.

La correzione, dunque, risulta essere precisa e in grado di modificare opportunamente i fattori di aggiustamento per ricostruire lo spettro corretto.

Il secondo test effettuato, invece, si suddivide nelle seguenti sequenze:

1. Sono stati modificati i fattori di aggiustamento relativi alle catene 1,8,13 e 20 impostando rispettivamente i valori: +1%, +3%, -10% e -15% ;
2. È stato acquisito lo spettro target e creato il relativo file di testo di calibrazione;
3. Sono stati annullati tutti i fattori di aggiustamento;
4. È stata avviata la procedura di correzione spettrale e creato il report di calibrazione.

I risultati relativi ai fattori di aggiustamento prima e dopo la calibrazione vengono riportati in Figura 5.8.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zone_1 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_2 Wavelength[%]:	0	0	-1	0	0	0	0	0	1	0	0	0	0	1	0	1
Zone_3 Wavelength[%]:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
Zone_4 Wavelength[%]:	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Zone_5 Wavelength[%]:	1	0	1	0	0	0	0	0	0	-1	0	0	0	0	0	0
Zone_6 Wavelength[%]:	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Zone_7 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_8 Wavelength[%]:	3	3	3	3	3	3	3	3	3	3	2	3	3	4	4	3
Zone_9 Wavelength[%]:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_10 Wavelength[%]:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_11 Wavelength[%]:	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
Zone_12 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_13 Wavelength[%]:	-9	-12	-10	-10	-11	-10	-10	-9	-10	-10	-11	-10	-10	-11	-10	-10
Zone_14 Wavelength[%]:	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
Zone_15 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_16 Wavelength[%]:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Zone_17 Wavelength[%]:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_18 Wavelength[%]:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_19 Wavelength[%]:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Zone_20 Wavelength[%]:	-14	-14	-15	-15	-15	-15	-15	-15	-15	-15	-13	-15	-15	-15	-15	-15
chain	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zone_1 Wavelength[%]:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Zone_2 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_3 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_4 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_5 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_6 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_7 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_8 Wavelength[%]:	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Zone_9 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_10 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_11 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_12 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_13 Wavelength[%]:	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
Zone_14 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_15 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_16 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_17 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_18 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_19 Wavelength[%]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zone_20 Wavelength[%]:	-15	-15	-15	-15	-15	-15	-15	-15	-15	-15	-15	-15	-15	-15	-15	-15

Figura 5.8: output della procedura di correzione spettrale

La tabella posta a sinistra rappresenta i 340 fattori di aggiustamento presenti al momento della creazione del file di calibrazione. La tabella posta a destra rappresenta gli stessi fattori di aggiustamento dopo la correzione. Sono state evidenziate con lo stesso colore le righe relative alle catene che hanno subito una variazione della corrente di alimentazione (catene 1,8,13 e 20).

In arancione sono stati evidenziati i fattori di aggiustamento che differiscono rispetto ai valori target.

L'output ideale del test sarebbe rappresentato da due seti di fattori identici tra loro: ciò comporterebbe una perfetta ricostruzione dello spettro.

Ciò che si osserva, invece, è che:

- Circa 88.5% dei fattori di aggiustamento (301 su 340) sono stati impostati correttamente;
- Circa 10.5% dei fattori di aggiustamento (37 su 340) si discostano dai valori corretti solamente per un valore di corrente pari a 1% in valore assoluto;
- Circa 0.5% dei fattori di aggiustamento (2 su 340) si discostano dai valori corretti per un valore di corrente di alimentazione del 2% in valore assoluto.

## CONCLUSIONI

La procedura di correzione dello spettro, sebbene impieghi un considerevole arco temporale, risulta precisa ed affidabile.

Dai dati analizzati in questo capitolo si evince che la rumorosità in fase di acquisizione ha un impatto che può essere ridotto considerevolmente tramite semplici medie aritmetiche sui valori digitali acquisiti.

Un altro punto di forza è rappresentato dal setup hardware, relativamente snello ed agevole. I dati acquisiti tramite spettrometro e tramite i fotodiodi dimostrano la robustezza della procedura automatica di correzione.

## CAPITOLO 6: CONCLUSIONI

Le procedure di calibrazione dell'uniformità spaziale e spettrale descritte nei Capitoli 4 e 5 permettono di migliorare le prestazioni di misura dell'intero sistema.

Come già descritto precedentemente, tali procedure non risultano conformi alle normative internazionali. Dunque, esse difficilmente possono essere sfruttate per produrre una certificazione e una classificazione verosimile del simulatore. Ciononostante, risulta utile applicare le procedure implementate per diversi motivi.

A titolo di esempio si consideri il grafico di Figura 6.1.

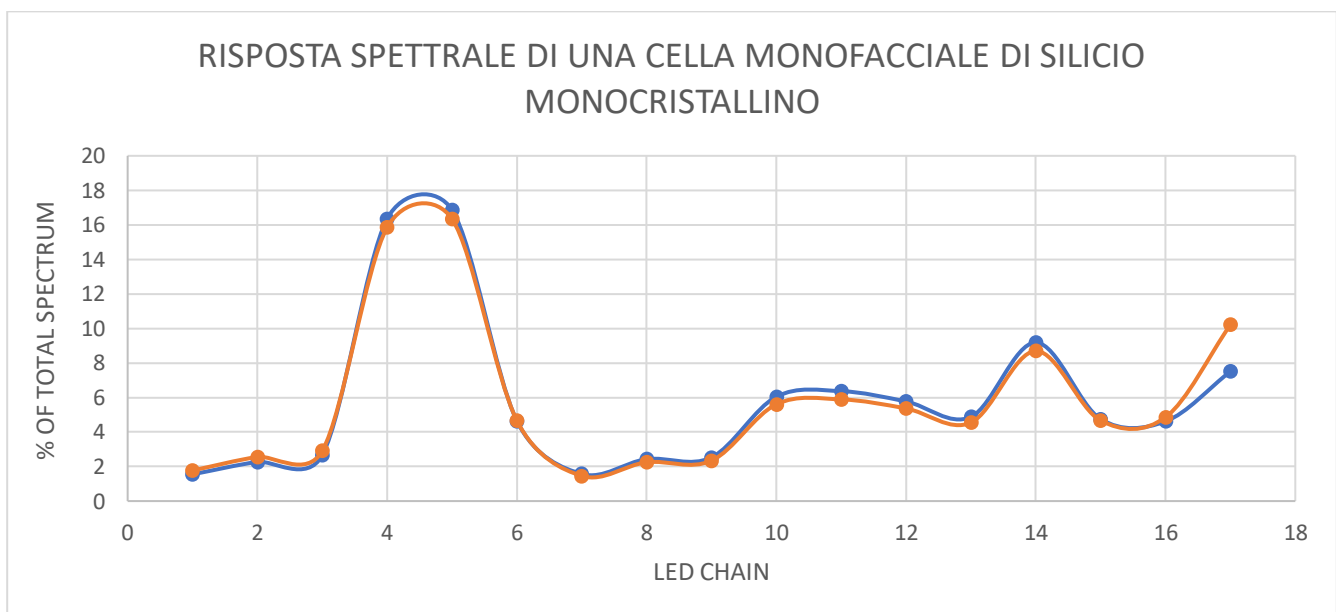


Figura 6.1: risposta spettrale di una cella fotovoltaica esposta ad uno spettro non stabile nel tempo

Il grafico rappresenta la risposta spettrale di una stessa cella al silicio esposta a due spettri leggermente diversi a causa dell'arco temporale nel quale sono stati acquisiti. In ascissa vengono riportate le 17 catene di LED della lampada del simulatore, in ordinata la percentuale di irradianza misurata accendendo singolarmente una catena.

Si può notare che le due curve si scostano reciprocamente per le catene numero 4 e 5 (luce bianca) e per la catena 17 (infrarosso).

Tali differenze equivalgono ad un assorbimento diverso di potenza. Ciò comporta la generazione di due correnti di cortocircuito diverse tra loro, sebbene la cella sotto test sia la medesima.

La procedura di correzione spettrale permette di mantenere uno spettro costante nel tempo e dunque di garantire una maggiore ripetibilità temporale di misura a parità di condizioni di una cella solare (usura e comparsa di difetti).

Per ciò che riguarda l'accuratezza di misura, è possibile affermare che un sistema correttamente calibrato permette di ottenere migliori prestazioni.

A tal proposito si consideri il grafico di Figura 6.2: esso rappresenta una serie di 30 acquisizioni di efficienza con la medesima cella al silicio monocristallino.

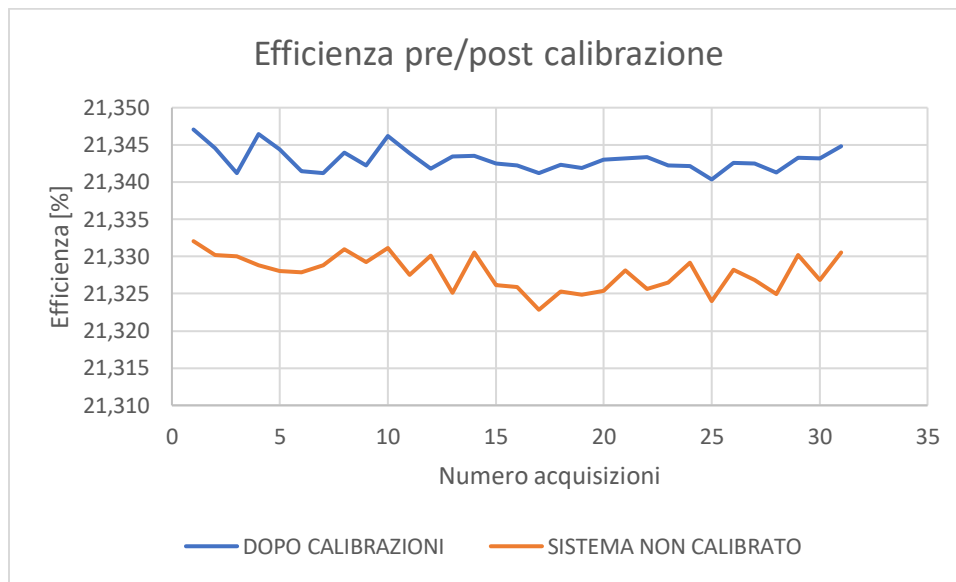


Figura 6.2: misure seriali di efficienza prima e dopo la calibrazione del sistema

La cella fotovoltaica utilizzata per il test presenta un valore di efficienza pari a 21.369% da certificato, misurata a 25°C con un livello di irradianza di 1000W/m<sup>2</sup> (condizioni standard).

	Efficienza [%]	
	pre-calibrazione	post calibrazione
<b>media</b>	21,328	21,343
<b>valore massimo</b>	21,332	21,347
<b>valore minimo</b>	21,323	21,340
<b>range di ripetibilità</b>	0,009	0,007
<b>errore assoluto</b>	0,041	0,026

Tabella 6.1: test di accuratezza prima e dopo la calibrazione

Dalla Tabella riassuntiva 6.1 è possibile concludere che il sistema sottostima l'efficienza sia prima la calibrazione che dopo. La ripetibilità rimane pressochè invariata poiché le acquisizioni sono state eseguite in serie.

Tuttavia, dopo la calibrazione in uniformità spaziale e correzione dello spettro, l'errore commesso nella misura di efficienza diminuisce di un fattore pari a 0,015%.

Ciò è dovuto al fatto che la calibrazione della lampada del simulatore permette di avvicinarsi maggiormente alle condizioni standard alle quali la cella sotto test è stata certificata.

In conclusione, è possibile affermare che le due procedure di calibrazione esposte in questo lavoro di tesi permettono di apportare importanti migliorie al sistema poiché:

1. Una buona uniformità di irradianza permette di raggiungere migliori valori di accuratezza in fase di misura dei parametri elettrici e delle figure di merito di una cella fotovoltaica;
2. Uno spettro non degradato nelle sue componenti permette di misurare correttamente parametri quali  $I_{sc}$  ed efficienza di una cella solare;
3. Una sorgente luminosa dalle caratteristiche stabili permette di ottenere una maggiore ripetibilità di misura nell'arco del tempo.



# APPENDICE A: LISTATO DELLA FUNZIONE DI AUTOCALIBRAZIONE DELL'UNIFORMITA' SPAZIALE

```
//LISTA DELLE VARIABILI COLLEGATE A INTERFACCIA GRAFICA
TYPE sSpatial_Uniformity_GUI :
STRUCT
    actual                : sSpatial_Uniformity ;
    calibrated            : sSpatial_Uniformity ;

    save                  : BOOL;
    next                  : BOOL ;
    next_index            : INT := 1;
    uniformity_measurement : BOOL := FALSE;
    uniformity_calibration : BOOL := FALSE;
    load_data              : BOOL := FALSE;
    abort                  : BOOL := FALSE;
    error_message         : STRING ;
    calibration_ended     : BOOL := TRUE;

END_STRUCT
END_TYPE

//LISTA DELLE COSTANTI UTILIZZATE NELL'ALGORITMO
{attribute 'qualified_only'}
VAR_GLOBAL CONSTANT

    number_of_pixels      : DINT := 20;
    number_of_zones       : DINT := 20;
    max_adj                : DINT := 0;
    min_adj                : DINT := -10;
    number_max_of_cycles  : DINT := 1000;
    NU_target              : DINT := 1;

END_VAR

//TIPO DI STRUTTURA UTILIZZATA PER LE ISTANZE ACTUAL E CALIBRATED
TYPE sSpatial_Uniformity :
STRUCT
    uniformity              : ARRAY[0..pSpatial_Uniformity.number_of_zones] OF LREAL;
    adj                     : ARRAY[0..pSpatial_Uniformity.number_of_zones] OF INT;
    NU                      : LREAL;
    irradiance_AVG         : LREAL;

END_STRUCT
END_TYPE

// LISTA DEGLI STATI UTILIZZATI NELLA PROCEDURA
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE eSpatial_Uniformity :
(
    idle_state := 0,
    open_U_file,
    read_U_file,
    convert_U_file,
    close_U_file,
    open_W_file,
    read_W_file,
    convert_W_file,
    close_W_file,
    acquisition_U_vector,
    acquisition_W_matrix,
    flash_W,
    W_measure,
    write_W_file,
    open_W_file_write,
    write_body_W,
    flash_U,
    U_measure,
    write_U_file,
    open_U_file_write,
    write_body_U,
    adjs_algorithm_check_NU,
    reset_variables_for_algorithm,
    calculate_actual_parameters,
    reset_alarms,
    set_adj_to_lamp,
    real_time_simulation,
    close_U_file_write

);
END_TYPE
```

```

//HEADER DELLA PROCEDURA, DICHIARAZIONE VARIABILI
FUNCTION_BLOCK FB_Spatial_Uniformity
VAR
weight
ARRAY[0..pSpatial_Uniformity.number_of_pixels,0..pSpatial_Uniformity.number_of_zones] OF LREAL;
adj
: ARRAY[0..pSpatial_Uniformity.number_of_zones] OF INT;
adj_temp
: ARRAY[0..pSpatial_Uniformity.number_of_zones] OF INT;
uniformity
: ARRAY[0..pSpatial_Uniformity.number_of_zones] OF LREAL;
uniformity_temp
: ARRAY[0..pSpatial_Uniformity.number_of_zones] OF LREAL;
NU
: ARRAY[0..pSpatial_Uniformity.number_max_of_cycles] OF LREAL;
NU_temp
: ARRAY[0..2*pSpatial_Uniformity.number_of_zones] OF LREAL;
i
: DINT;
j
: DINT;
l
: DINT;
q
: DINT;
number_of_cycles
: DINT;
best_zone
: DINT;
NU_temp_min
: LREAL;
sum
: LREAL;
irr_max
: LREAL;
irr_min
: LREAL;
irr_avg_after_calibration
: LREAL;
Calibration_step
: eSpatial_Uniformity;
current_zone
: INT;
number_of_samples
: INT;
irr_sum
: LREAL;
offset2
: DINT;
irr_avg
: LREAL;

// File management
fbNtGetTime
: NT_GetTime;
curTime
: Timestruct;
fbFormatString
: FB_FormatString;
fbFileOpen
: FB_FileOpen;
fileHandle
: UINT;
fbFilePuts
: FB_FilePuts;
fbFileClose
: FB_FileClose;
fbFileGets
: FB_FileGets;

fbGetNumFromString
: FB_EnumStringNumbers; (* Get a number from string *)
_numAsString: T_MaxString;
indx: INT;
enumCmd : INT;
end_pixel
: BOOL ;
dark_irr_avg
: LREAL;
light_irr_avg: LREAL;

fbFilewrite
: FB_Filewrite;
tmpMaxString
: T_MaxString;
offset
: INT;
m
: INT;
MeasureToBeSaved: BOOL;

_numMeasure: DINT;
END_VAR

//CORPO DELLA PROCEDURA
CASE Calibration_step OF
//reset delle variabili di stato utilizzate e degli allarmi visualizzati a interfaccia
eSpatial_Uniformity.idle_state:
fbFileOpen( bExecute := FALSE);
fbFileGets( bExecute := FALSE);
fbFileClose( bExecute := FALSE);
fbFilewrite( bExecute := FALSE);
GuiSpatialUniformity.abort := FALSE;
GuiSpatialUniformity.next := FALSE;
GuiSpatialUniformity.next_index := 1;
end_pixel := TRUE;
current_zone := 1;
Calibration_step := eSpatial_Uniformity.real_time_simulation;
GuiSpatialUniformity.calibration_ended := TRUE;
MeasureToBeSaved := FALSE;

IF GuiSpatialUniformity.load_data THEN
Calibration_step:=eSpatial_Uniformity.reset_alarms;
ELSIF GuiSpatialUniformity.uniformity_measurement THEN
Calibration_step:=eSpatial_Uniformity.reset_alarms;
GuiSpatialUniformity.calibration_ended := FALSE;
ELSIF GuiSpatialUniformity.uniformity_calibration THEN
Calibration_step:=eSpatial_Uniformity.reset_alarms;
GuiSpatialUniformity.calibration_ended := FALSE;
ELSIF GuiSpatialUniformity.save THEN

```

```

        m :=1;
        Calibration_step := eSpatial_Uniformity.reset_alarms;
    END_IF
eSpatial_Uniformity.reset_alarms:
    IF GuiSpatialUniformity.load_data THEN
        Calibration_step:=eSpatial_Uniformity.open_U_file;
    ELSIF GuiSpatialUniformity.uniformity_measurement THEN
        Calibration_step:=eSpatial_Uniformity.acquisition_U_vector;
        LampData.W_ZoneSelect:= 0;
        MeasureToBeSaved := TRUE;
    GuiSpatialUniformity.calibration_ended := FALSE;
    ELSIF GuiSpatialUniformity.uniformity_calibration THEN
        GuiSpatialUniformity.uniformity_calibration := FALSE;
        Calibration_step:=eSpatial_Uniformity.acquisition_W_matrix;
        LampData.W_ZoneSelect:= 0;
        GuiSpatialUniformity.calibration_ended := FALSE;
    ELSIF GuiSpatialUniformity.save THEN
        Calibration_step := eSpatial_Uniformity.set_adj_to_lamp;
        GuiSpatialUniformity.actual := GuiSpatialUniformity.calibrated;
    END_IF

    GuiSpatialUniformity.error_message := '';

//apertura,lettura,chiusura e gestione del file contenente i valori di irradianza sui vari pixel
eSpatial_Uniformity.open_U_file:
    fbFileOpen(
        sNetId      := ''
        sPathName   := 'D:\lamp\Spatial_uniformity\U.txt'
        nMode       := FOPEN_MODERead OR FOPEN_MODETEXT
        ePath       := PATH_GENERIC
        bExecute    := TRUE
        hFile       => fileHandle
    );
    IF NOT fbFileOpen.bBusy THEN
        IF fbFileOpen.bError THEN
            Calibration_step:=eSpatial_Uniformity.close_W_file;
            GuiSpatialUniformity.error_message := 'Error during reading uniformity file data';
        ELSE
            Calibration_step := eSpatial_Uniformity.read_U_file;
        END_IF
        fbFileOpen( bExecute := FALSE );
    END_IF

eSpatial_Uniformity.read_U_file:
    fbFileGets(
        sNetId := ''
        hFile  := fileHandle
        bExecute:= TRUE
        tTimeout:= T#5S
    );
    IF NOT fbFileGets.bBusy THEN
        IF fbFileGets.bError THEN
            Calibration_step:=eSpatial_Uniformity.close_W_file;
            GuiSpatialUniformity.error_message := 'Error during reading uniformity file data';
        ELSIF fbFileGets.bEOF THEN
            // All file read
            Calibration_step := eSpatial_Uniformity.close_U_file;
        ELSIF FIND( STR1 := fbFileGets.sLine, STR2 := 'PIXEL IRRADIANCE' ) > 0 THEN // Search
            first Data: "PIXEL IRRADIANCE"
            Calibration_step:= eSpatial_Uniformity.convert_U_file;
            indx := 1;
        END_IF
        fbFileGets( bExecute := FALSE );
    END_IF

eSpatial_Uniformity.convert_U_file:
    fbFileGets(
        sNetId := ''
        hFile  := fileHandle
        bExecute:= TRUE
        tTimeout:= T#5S
    );
    IF NOT fbFileGets.bBusy THEN
        IF fbFileGets.bError THEN
            Calibration_step:=eSpatial_Uniformity.close_W_file;
            GuiSpatialUniformity.error_message := 'Error during reading uniformity file data';
        ELSIF fbFileGets.bEOF OR (indx = pSpatial_Uniformity.number_of_zones + 1) THEN
            // All file read
            Calibration_step := eSpatial_Uniformity.close_U_file;
        ELSE
            // Index -> No needed
            fbGetNumFromString(
                sSearch := fbFileGets.sLine
                eCmd     := E_EnumCmdType.eEnumCmd_First
                eType    := E_NumGroupTypes.eNumGroup_Float,
                sNumber => _numAsString
            );
            IF NOT fbGetNumFromString.bEOS AND
            NOT (fbGetNumFromString.sNumber = '')
            THEN

```

```

                                GuiSpatialUniformity.actual.uniformity[indx] := STRING_TO_LREAL(
_numAsString );
                                indx := indx + 1;
                                END_IF
                                END_IF
                                fbFileGets( bExecute := FALSE );
                                END_IF
eSpatial_Uniformity.close_U_file:
fbFileClose(
    sNetId := ''
    hFile := fileHandle
    bExecute:= TRUE
);
IF NOT fbFileClose.bBusy THEN
    IF fbFileClose.bError THEN
        Calibration_step:=eSpatial_Uniformity.idle_state;
        GuiSpatialUniformity.error_message := 'Error during reading uniformity file data';
    ELSE
        Calibration_step := eSpatial_Uniformity.open_w_file;
    END_IF
    fbFileClose( bExecute := FALSE );
END_IF

//apertura,lettura,chiusura e gestione del file contenente i valori peso della matrice 20x20

eSpatial_Uniformity.open_w_file:
fbFileOpen(
    sNetId := ''
    sPathName := 'D:\lamp\Spatial_uniformity\w.txt'
    nMode := FOPEN_MODERead OR FOPEN_MODETEXT
    ePath := PATH_GENERIC
    bExecute := TRUE
    hFile => fileHandle
);
IF NOT fbFileOpen.bBusy THEN
    IF fbFileOpen.bError THEN
        Calibration_step:=eSpatial_Uniformity.close_w_file;
        GuiSpatialUniformity.error_message := 'Error during reading uniformity file data';
    ELSE
        Calibration_step := eSpatial_Uniformity.read_w_file;
    END_IF
    fbFileOpen( bExecute := FALSE );
END_IF

eSpatial_Uniformity.read_w_file:
fbFileGets(
    sNetId := ''
    hFile := fileHandle
    bExecute:= TRUE
    tTimeout:= T#5S
);
IF NOT fbFileGets.bBusy THEN
    IF fbFileGets.bError THEN
        Calibration_step:=eSpatial_Uniformity.close_w_file;
        GuiSpatialUniformity.error_message := 'Error during reading uniformity file data';
    ELSIF fbFileGets.bEOF THEN
        // All file read
        Calibration_step := eSpatial_Uniformity.close_w_file;
    ELSIF FIND( STR1 := fbFileGets.sLine, STR2 := 'ZONE1' ) > 0 THEN // Search first Data:
        Calibration_step:= eSpatial_Uniformity.convert_w_file;
        indx := 1;
    END_IF
    fbFileGets( bExecute := FALSE );
END_IF

eSpatial_Uniformity.convert_w_file:
fbFileGets(
    sNetId := ''
    hFile := fileHandle
    bExecute:= TRUE
    tTimeout:= T#5S
);
IF NOT fbFileGets.bBusy THEN
    IF fbFileGets.bError THEN
        Calibration_step:=eSpatial_Uniformity.close_w_file;
        GuiSpatialUniformity.error_message := 'Error during reading uniformity file data';
    ELSIF fbFileGets.bEOF OR (indx = pSpatial_Uniformity.number_of_pixels + 1) THEN
        // All file read
        Calibration_step := eSpatial_Uniformity.close_w_file;
    ELSE
        // Index -> No needed
        enumCmd := E_EnumCmdType.eEnumCmd_First;
        FOR q:=1 TO pSpatial_Uniformity.number_of_zones DO
            fbGetNumFromString(
                sSearch := fbFileGets.sLine
            ,
                eCmd := enumCmd
                eType := E_NumGroupTypes.eNumGroup_Float
                sNumber => _numAsString
            );
            IF NOT fbGetNumFromString.bEOS AND

```

```

                                NOT (fbGetNumFromString.sNumber = '')
                                THEN
                                weight[indx,q] := STRING_TO_LREAL( _numASstring );
                                END_IF
                                enumCmd := E_EnumCmdType.eEnumCmd_Next;
END_FOR
                                END_IF
                                indx := indx + 1;
                                fbFileGets( bExecute := FALSE );
END_IF

eSpatial_Uniformity.close_w_file:
    fbFileClose(
        sNetId := ''
        hFile := fileHandle
        bExecute:= TRUE
    );
    IF NOT fbFileClose.bBusy THEN
        IF fbFileClose.bError THEN
            Calibration_step:=eSpatial_Uniformity.idle_state;
            GuiSpatialUniformity.error_message := 'Error during reading uniformity file data';
        ELSE
            Calibration_step := eSpatial_Uniformity.calculate_actual_parameters;
        END_IF
        fbFileClose( bExecute := FALSE );
    END_IF

//stato per la gestione della misura di non-uniformità
eSpatial_Uniformity.acquisition_U_vector:
    IF GuiSpatialUniformity.abort=FALSE THEN
        IF GuiSpatialUniformity.next=TRUE OR GuiSpatialUniformity.next_index >
pSpatial_Uniformity.number_of_pixels THEN
            IF GuiSpatialUniformity.next_index<=pSpatial_Uniformity.number_of_pixels THEN
                Calibration_step:= eSpatial_Uniformity.flash_U;
            ELSE
                Calibration_step:= eSpatial_Uniformity.calculate_actual_parameters;
                GuiSpatialUniformity.calibration_ended := TRUE;
            END_IF
        END_IF
    ELSE
        Calibration_step:= eSpatial_Uniformity.idle_state ;
        GuiSpatialUniformity.uniformity_measurement:=FALSE ;
    END_IF

//stato per il calcolo dei parametri di non uniformità e irradianza media nei casi di misura o di calibrazione
eSpatial_Uniformity.calculate_actual_parameters:
    irr_max := 0;
    irr_min := 10000;
    IF GuiSpatialUniformity.uniformity_measurement THEN
        GuiSpatialUniformity.actual.uniformity := uniformity ;
        GuiSpatialUniformity.uniformity_measurement := FALSE;
    END_IF
    FOR l:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
        IF GuiSpatialUniformity.actual.uniformity[l]>irr_max THEN
            irr_max := GuiSpatialUniformity.actual.uniformity[l];
        END_IF
        IF GuiSpatialUniformity.actual.uniformity[l]<irr_min THEN
            irr_min := GuiSpatialUniformity.actual.uniformity[l];
        END_IF
    END_FOR
    GuiSpatialUniformity.actual.NU := ((irr_max-
irr_min)/(irr_max+irr_min))*100;

    sum :=0;
    FOR l:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
        sum := sum + GuiSpatialUniformity.actual.uniformity[l];
    END_FOR
    GuiSpatialUniformity.actual.irradiance_AVG := sum /
pSpatial_Uniformity.number_of_pixels;

    IF GuiSpatialUniformity.load_data THEN
        FOR l:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
            sum:= 0;
            FOR i:=1 TO K_LAMP_WAVELENGTH_N BY 1 DO
                sum := sum + LampData.R_ZoneIntensityAdj_ZX_WX[l][i];
            END_FOR
            GuiSpatialUniformity.actual.adj[l] :=
LREAL_TO_INT(sum/K_LAMP_WAVELENGTH_N);
        END_FOR
        GuiSpatialUniformity.load_data := FALSE;
    END_IF

    IF GuiSpatialUniformity.save OR MeasureToBeSaved THEN
        LampData.C_SetConfig := TRUE;
        Calibration_step:= eSpatial_Uniformity.open_U_file_write;
    ELSE
        Calibration_step:= eSpatial_Uniformity.idle_state;

```

```

                                END_IF

//stato per gestire il flash della lampada durante la fase di misura
eSpatial_Uniformity.flash_U:
GuiSpatialUniformity.next := FALSE;
IF BCSSun_Meas.R_MeasureBusy=FALSE THEN
    LampData.W_ZoneSelect:= 0 ;
    BCSSun_Mngr.C_bStartMeasure := TRUE ;
    Calibration_step:= eSpatial_Uniformity.U_measure;
END_IF

//stato per l'elaborazione dei dati a partire dai txt presenti nella cartella D:\log del pc botticelli. Si fa la
media dei campioni di irradianza durante la prova sink1 e viene sottratta la media dei campioni di irradianza della
prova in dark reverse (considerata dunque rumore di fondo).
Parte relativa alla misura
eSpatial_Uniformity.U_measure:
    IF BCSSun_Meas.R_MeasureBusy=FALSE THEN
        number_of_samples :=0;
        irr_sum := 0;
        _numMeasure := BCSSun_Meas.iMeas.IV[K_BCSSUN_LF1].W_NSAMPLE;
        FOR i := 0 TO BCSSun_Meas.iMeas.IV[K_BCSSUN_LF1].W_NSAMPLE -1 DO
            irr_sum :=
                irr_sum +
                BCSSun_Meas.iMeas.R_AcqResults[i,4];
            number_of_samples := number_of_samples +1;
        END_FOR
        light_irr_avg := irr_sum / number_of_samples ;

        irr_sum:= 0;
        number_of_samples :=0;
        _numMeasure := BCSSun_Meas.iMeas.IV[K_BCSSUN_DARK].W_NSAMPLE;
        offset2 := BCSSun_Meas.iMeas.IV[K_BCSSUN_LF1].W_NSAMPLE +
        BCSSun_Meas.iMeas.IV[K_BCSSUN_LF2].W_NSAMPLE;
        FOR i := offset2 TO offset2 + BCSSun_Meas.iMeas.IV[K_BCSSUN_DARK].W_NSAMPLE -1 DO
            irr_sum :=
                irr_sum +
                BCSSun_Meas.iMeas.R_AcqResults[i,4];
            number_of_samples := number_of_samples +1;
        END_FOR
        dark_irr_avg := irr_sum / number_of_samples ;
        uniformity[GuiSpatialUniformity.next_index] := light_irr_avg - dark_irr_avg;

        Calibration_step:=eSpatial_Uniformity.flash_U;
        end_pixel:=TRUE;
        GuiSpatialUniformity.next_index:= GuiSpatialUniformity.next_index +1;
        Calibration_step:=eSpatial_Uniformity.acquisition_U_vector;
    END_IF

//stato per la gestione della fase di calibrazione (dunque acquisizione matrice W 20x20)
eSpatial_Uniformity.acquisition_W_matrix:
    IF GuiSpatialUniformity.abort=FALSE THEN
        IF GuiSpatialUniformity.next=TRUE OR GuiSpatialUniformity.next_index >
pSpatial_Uniformity.number_of_pixels THEN
            IF GuiSpatialUniformity.next_index<=pSpatial_Uniformity.number_of_pixels THEN
                Calibration_step:= eSpatial_Uniformity.flash_W;
            ELSE
                Calibration_step:= eSpatial_Uniformity.reset_variables_for_alghorithm;
                GuiSpatialUniformity.calibration_ended := TRUE;
                LampData.W_ZoneSelect:= 0;
            END_IF
        END_IF
    ELSE
        Calibration_step:= eSpatial_Uniformity.idle_state ;
        GuiSpatialUniformity.uniformity_calibration:=FALSE ;
        LampData.W_ZoneSelect:= 0;
    END_IF

//gestione delle 400 flashate per la fase di calibrazione
eSpatial_Uniformity.flash_W:
GuiSpatialUniformity.next := FALSE;
IF BCSSun_Meas.R_MeasureBusy=FALSE THEN
    LampData.W_ZoneSelect:= current_zone ;
    BCSSun_Mngr.C_bStartMeasure := TRUE ;
    Calibration_step:= eSpatial_Uniformity.W_measure;
END_IF

//stato per l'elaborazione dei dati a partire dai txt presenti nella cartella D:\log del pc botticelli. Si fa la
media dei campioni di irradianza durante la prova sink1 e viene sottratta la media dei campioni di irradianza della
prova in dark reverse (considerata dunque rumore di fondo)
Parte relativa alla calibrazione
eSpatial_Uniformity.W_measure:

```

```

IF BCSSun_Meas.R_MeasureBusy=FALSE THEN
    number_of_samples :=0;
    irr_sum := 0;
    _numMeasure := BCSSun_Meas.iMeas.IV[K_BCSSUN_LF1].W_NSAMPLE;
    FOR i := 0 TO BCSSun_Meas.iMeas.IV[K_BCSSUN_LF1].W_NSAMPLE -1 DO
        irr_sum :=
            irr_sum +
            BCSSun_Meas.iMeas.R_AcqResults[i,4];
        number_of_samples := number_of_samples +1;
    END_FOR
    light_irr_avg := irr_sum / number_of_samples ;

    irr_sum:= 0;
    number_of_samples :=0;
    _numMeasure := BCSSun_Meas.iMeas.IV[K_BCSSUN_DARK].W_NSAMPLE;
    offset2 := BCSSun_Meas.iMeas.IV[K_BCSSUN_LF1].W_NSAMPLE +
    BCSSun_Meas.iMeas.IV[K_BCSSUN_LF2].W_NSAMPLE;
    FOR i := offset2 TO offset2 + BCSSun_Meas.iMeas.IV[K_BCSSUN_DARK].W_NSAMPLE -1 DO
        irr_sum :=
            irr_sum +
            BCSSun_Meas.iMeas.R_AcqResults[i,4];
        number_of_samples := number_of_samples +1;
    END_FOR
    dark_irr_avg := irr_sum / number_of_samples ;
    weight[GuiSpatialUniformity.next_index ,current_zone] := light_irr_avg -
dark_irr_avg;
    Calibration_step:=eSpatial_Uniformity.flash_W;
    current_zone := current_zone +1;
END_IF

    IF current_zone = pSpatial_Uniformity.number_of_zones+1 THEN
        end_pixel:=TRUE;
        current_zone:= 1;
        GuiSpatialUniformity.next_index:= GuiSpatialUniformity.next_index +1;
        Calibration_step:=eSpatial_Uniformity.acquisition_W_matrix;
    END_IF

//stato per la scrittura dei fattori di aggiustamento nella lampada
eSpatial_Uniformity.set_adj_to_lamp:

    FOR i := 1 TO K_LAMP_ZONE_N DO
        GuiSpatialUniformity.actual.adj[i]:= adj[i];
    END_FOR

    Calibration_step := eSpatial_Uniformity.open_w_file_write;

//stati per l'apertura, scrittura e gestione del file txt contenente la matrice 20x20. Manca lo stato relativo di
chiusura del file poiché viene sfruttato lo stato di chiusura file utilizzato in lettura
eSpatial_Uniformity.open_w_file_write:

    fbFileOpen(
        sNetId           := ''
        sPathName        := 'D:\lamp\Spatial_uniformity\w.txt'
        nMode            := FOPEN_MODEWRITE OR FOPEN_MODETEXT
        ePath            := PATH_GENERIC
        bExecute := TRUE
        hFile            => fileHandle
    );
    IF NOT fbFileOpen.bBusy THEN
        IF fbFileOpen.bError THEN
            Calibration_step:=eSpatial_Uniformity.close_w_file;
            GuiSpatialUniformity.error_message := 'Error during saving uniformity data';
        ELSE
            Calibration_step := eSpatial_Uniformity.write_w_file;
        END_IF
        fbFileOpen( bExecute := FALSE );
    END_IF

eSpatial_Uniformity.write_w_file:

tmpMaxString := 'ZONE1' ;
FOR i:=2 TO pSpatial_Uniformity.number_of_zones BY 1 DO
    tmpMaxString := concat( STR1 := tmpMaxString, STR2 := concat(' ZONE',DINT_TO_STRING( i )) );
END_FOR

    fbFormatString(
        sFormat := '%s$n',
        arg1 := F_STRING(tmpMaxString)
        sout => tmpMaxString
    );

    fbFilewrite(
        sNetId := ''
        hFile := fileHandle
    );
    pwriteBuff := ADR( tmpMaxString )
    cbwriteLen := INT_TO_UINT( LEN( tmpMaxString ) ),
    bExecute := TRUE);

    IF NOT fbFilewrite.bBusy THEN
        IF fbFilewrite.bError THEN
            Calibration_step:=eSpatial_Uniformity.close_w_file;
            GuiSpatialUniformity.error_message := 'Error during saving uniformity data';
        ELSE
            Calibration_step:=eSpatial_Uniformity.write_body_w;
        END_IF
    END_IF

```

```

        END_IF
        fbFilewrite( bExecute := FALSE );
    END_IF
    m := 1;
espatial_Uniformity.write_body_W:
        fbFormatString(
            sFormat := '%f',
            arg1 := F_LREAL(weight[m,1]),
            sout => tmpMaxString );

        FOR i:= 2 TO pSpatial_Uniformity.number_of_zones BY 1 DO
            fbFormatString(
                sFormat := '%s %f',
                arg1 := F_STRING(tmpMaxString) ,
                arg2 := F_LREAL(weight[m,i]) ,
                sout => tmpMaxString );
            END_FOR

        tmpMaxString := concat( STR1 := tmpMaxString, STR2 := '$n' );

        fbFilewrite(
            sNetId := '' ,
            hFile := fileHandle ,
            pwriteBuff := ADR( tmpMaxString ) ,
            cbwriteLen := INT_TO_UINT( LEN( tmpMaxString ) ),
            bExecute := TRUE);

        IF NOT fbFilewrite.bBusy THEN
            IF fbFilewrite.bError THEN
                Calibration_step:=espatial_Uniformity.close_W_file;
                GuiSpatialUniformity.error_message := 'Error during saving uniformity data';
            ELSE
                IF m = pSpatial_Uniformity.number_of_pixels THEN
                    Calibration_step:=espatial_Uniformity.close_W_file;
                ELSE
                    Calibration_step:=espatial_Uniformity.write_body_W;
                    m := m+1;
                END_IF
            END_IF
            fbFilewrite( bExecute := FALSE );
        END_IF
    END_IF
END_IF

//stati per l'apertura, scrittura, gestione e chiusura del file txt contenente la misura di irradianza sui 20 pixel
espatial_Uniformity.open_U_file_write:
        fbFileOpen(
            sNetId := '' ,
            sPathName := 'D:\lamp\Spatial_uniformity\U.txt' ,
            nMode := FOPEN_MODEWRITE OR FOPEN_MODETEXT ,
            ePath := PATH_GENERIC ,
            bExecute := TRUE ,
            hFile => fileHandle );
        IF NOT fbFileOpen.bBusy THEN
            IF fbFileOpen.bError THEN
                Calibration_step:=espatial_Uniformity.close_U_file;
                GuiSpatialUniformity.error_message := 'Error during saving uniformity data';
            ELSE
                Calibration_step := espatial_Uniformity.write_U_file;
            END_IF
            fbFileOpen( bExecute := FALSE );
        END_IF
    END_IF

espatial_Uniformity.write_U_file:
    tmpMaxString := 'PIXEL IRRADIANCE' ;
        fbFormatString(
            sFormat := '%s$n',
            arg1 := F_STRING(tmpMaxString) ,
            sout => tmpMaxString );
        fbFilewrite(
            sNetId := '' ,
            hFile := fileHandle ,
            pwriteBuff := ADR( tmpMaxString ) ,
            cbwriteLen := INT_TO_UINT( LEN( tmpMaxString ) ),
            bExecute := TRUE);

        IF NOT fbFilewrite.bBusy THEN
            IF fbFilewrite.bError THEN
                Calibration_step:=espatial_Uniformity.close_U_file;
                GuiSpatialUniformity.error_message := 'Error during saving uniformity data';
            ELSE
                Calibration_step:=espatial_Uniformity.write_body_U;
            END_IF
            fbFilewrite( bExecute := FALSE );
        END_IF
    m := 1;
espatial_Uniformity.write_body_U:

```



```

        fbFormatString(
            sFormat := '%f',
            arg1 := F_LREAL(GuiSpatialUniformity.actual.uniformity[m]),
            sout => tmpMaxString );
    tmpMaxString := concat( STR1 := tmpMaxString, STR2 := '$n' );

    fbFilewrite(
        sNetId := '',
        hFile := fileHandle,
        pwriteBuff := ADR( tmpMaxString ),
        cbwriteLen := INT_TO_UINT( LEN( tmpMaxString ) ),
        bExecute := TRUE);

    IF NOT fbFilewrite.bBusy THEN
        IF fbFilewrite.bError THEN
            Calibration_step:=eSpatial_Uniformity.close_U_file_write;
            GuiSpatialUniformity.error_message := 'Error during saving uniformity data';
        ELSE
            IF m = pSpatial_Uniformity.number_of_pixels THEN
                Calibration_step:=eSpatial_Uniformity.close_U_file_write;
            ELSE
                Calibration_step:=eSpatial_Uniformity.write_body_U;
                m := m+1;
            END_IF
            fbFilewrite( bExecute := FALSE );
        END_IF
    END_IF
END_IF

espatial_Uniformity.close_U_file_write:
    fbFileclose(
        sNetId := '',
        hFile := fileHandle,
        bExecute:= TRUE );
    IF NOT fbFileclose.bBusy THEN
        IF fbFileclose.bError THEN
            Calibration_step:=eSpatial_Uniformity.idle_state;
            GuiSpatialUniformity.error_message := 'Error during saving uniformity data';
        ELSE
            Calibration_step := espatial_Uniformity.idle_state;
        END_IF
        fbFileclose( bExecute := FALSE );
        GuiSpatialUniformity.error_message := 'ACTUAL UNIFORMITY MAP CORRECTLY SAVED';
    END_IF

//stato per il reset di tutte le variabili utilizzate per l'ocòalgoritmo di calibrazione
espatial_Uniformity.reset_variables_for_algorithm:
    FOR i:=1 TO pSpatial_Uniformity.number_of_zones DO
        adj[i] := 0;
        adj_temp[i] := 0;
        uniformity_temp[i] := 0;
    END_FOR

    FOR i:=0 TO 2*pSpatial_Uniformity.number_of_zones DO
        NU_temp[i] := 0;
    END_FOR
    FOR i:=0 TO pSpatial_Uniformity.number_max_of_cycles DO
        NU[i] := 0;
    END_FOR
    NU[0] := 100;
    number_of_cycles := 0;
    Calibration_step := espatial_Uniformity.adjs_algorithm_check_NU;

//stato che contiene l'ocòelaborazione dei dati per il calcolo dei fattori di aggiustamento della calibrazione.
L'ocòalgoritmo varia tutti i fattori di aggiustamento con un pi-| o meno 1% cercando di minimizzare il fattore di non
uniformità. Per i dettagli guardare il diagramma di flusso a pagina 7.
espatial_Uniformity.adjs_algorithm_check_NU:
    IF NU[number_of_cycles] > pSpatial_Uniformity.NU_target THEN
        FOR j:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
            sum := 0;
            FOR i:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
                sum := sum + (weight[j,i]*(1+(INT_TO_REAL(adj[i])/100)));
            END_FOR
            uniformity[j] :=sum;
        END_FOR
        irr_max := 0;
        irr_min := 10000;
        FOR i:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
            IF uniformity[i]>irr_max THEN
                irr_max := uniformity[i];
            END_IF
            IF uniformity[i]<irr_min THEN
                irr_min := uniformity[i];
            END_IF
        END_FOR
        NU[number_of_cycles+1] := ((irr_max-irr_min)/(irr_max+irr_min))*100;
        IF NU[number_of_cycles+1]<=pSpatial_Uniformity.NU_target THEN
            Calibration_step := espatial_Uniformity.idle_state;
        END_IF
    END_IF

```

```

GuiSpatialUniformity.error_message := 'calibration ended';
GuiSpatialUniformity.calibrated.adj:= adj;
GuiSpatialUniformity.calibrated.uniformity:= uniformity;
GuiSpatialUniformity.calibrated.NU:= NU[number_of_cycles+1];
sum := 0;
FOR l:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
    sum := sum + uniformity[l];
END_FOR
GuiSpatialUniformity.calibrated.irradiance_AVG :=
sum/pSpatial_Uniformity.number_of_pixels;
ELSE
adj_temp := adj;
FOR i :=1 TO 2*pSpatial_Uniformity.number_of_zones BY 1 DO
    adj_temp := adj;
    IF i<= pSpatial_Uniformity.number_of_zones THEN
        uniformity_temp := uniformity;
        IF adj_temp[i]<pSpatial_Uniformity.max_adj THEN
            adj_temp[i] := adj_temp[i]+1;
        END_IF
        FOR j:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
            sum := 0;
            FOR l:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
                sum := sum + (weight[j,l]*(1+(INT_TO_REAL(adj_temp[l])/100)));
            END_FOR
            uniformity_temp[j] :=sum;
        END_FOR
        irr_max := 0;
        irr_min := 10000;
        FOR l:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
            IF uniformity_temp[l]>irr_max THEN
                irr_max := uniformity_temp[l];
            END_IF
            IF uniformity_temp[l]<irr_min THEN
                irr_min := uniformity_temp[l];
            END_IF
        END_FOR
        NU_temp[i] := ((irr_max-irr_min)/(irr_max+irr_min))*100;
    END_IF
    IF i> pSpatial_Uniformity.number_of_zones THEN
        uniformity_temp := uniformity;
        IF adj_temp[i-
pSpatial_Uniformity.number_of_zones]>pSpatial_Uniformity.min_adj THEN
            adj_temp[i-pSpatial_Uniformity.number_of_zones] := adj_temp[i-
pSpatial_Uniformity.number_of_zones]-1;
        END_IF
        FOR j:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
            sum := 0;
            FOR l:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
                sum := sum + (weight[j,l]*(1+(INT_TO_REAL(adj_temp[l])/100)));
            END_FOR
            uniformity_temp[j] :=sum;
        END_FOR
        irr_max := 0;
        irr_min := 10000;
        FOR l:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
            IF uniformity_temp[l]>irr_max THEN
                irr_max := uniformity_temp[l];
            END_IF
            IF uniformity_temp[l]<irr_min THEN
                irr_min := uniformity_temp[l];
            END_IF
        END_FOR
        NU_temp[i] := ((irr_max-irr_min)/(irr_max+irr_min))*100;
    END_IF
END_FOR
NU_temp_min := 100;
FOR l:=1 TO 2*pSpatial_Uniformity.number_of_zones BY 1 DO
    IF NU_temp[l]<NU_temp_min THEN
        NU_temp_min := NU_temp[l];
        best_zone := l;
    END_IF
END_FOR
IF NU_temp_min>= NU[number_of_cycles+1] THEN
    calibration_step := eSpatial_Uniformity.idle_state;
    GuiSpatialUniformity.error_message := 'calibration ended';
    GuiSpatialUniformity.calibrated.adj:= adj;
    GuiSpatialUniformity.calibrated.uniformity:= uniformity;
    GuiSpatialUniformity.calibrated.NU:= NU[number_of_cycles+1];
    sum := 0;
    FOR l:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
        sum := sum + uniformity[l];
    END_FOR
    GuiSpatialUniformity.calibrated.irradiance_AVG :=
sum/pSpatial_Uniformity.number_of_pixels;
END_IF
IF best_zone<=pSpatial_Uniformity.number_of_zones THEN
    adj[best_zone] := adj[best_zone]+1;
END_IF
IF best_zone>pSpatial_Uniformity.number_of_zones THEN
    adj[best_zone-pSpatial_Uniformity.number_of_zones] := adj[best_zone-
pSpatial_Uniformity.number_of_zones]-1;

```

```

        END_IF
        number_of_cycles := number_of_cycles+1;
        IF number_of_cycles>pSpatial_Uniformity.number_max_of_cycles THEN
            Calibration_step := eSpatial_Uniformity.idle_state;
            GuiSpatialUniformity.error_message := 'Error during calibration: calibration aborted';
        END_IF

    END_IF

ELSE
    Calibration_step := eSpatial_Uniformity.idle_state;
    GuiSpatialUniformity.error_message := 'calibration ended';
    GuiSpatialUniformity.calibrated.adj:= adj;
    GuiSpatialUniformity.calibrated.uniformity:= uniformity;
    GuiSpatialUniformity.calibrated.NU:= NU[number_of_cycles+1];
    sum := 0;
    FOR l:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
        sum := sum + uniformity[l];
    END_FOR
    GuiSpatialUniformity.calibrated.irradiance_AVG := sum/pSpatial_Uniformity.number_of_pixels;
END_IF

//stato per la simulazione real time della mappa di irradianza. Tramite un continuo switch tra tale stato e idle
state, da interfaccia è possibile editare i fattori di aggiustamento e vedere in tempo reale i risultati della
simulazione

eSpatial_Uniformity.real_time_simulation :

        FOR j:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
            sum := 0;
            FOR l:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
                sum := sum +
(weight[j,l]*(1+(INT_TO_REAL(GuiSpatialUniformity.calibrated.adj[l])/100)));
            END_FOR
            GuiSpatialUniformity.calibrated.uniformity[j] :=sum;
            END_FOR

            irr_max := 0;
            irr_min := 10000;
            FOR l:=1 TO pSpatial_Uniformity.number_of_zones BY 1 DO
                IF GuiSpatialUniformity.calibrated.uniformity[l]>irr_max THEN
                    irr_max := GuiSpatialUniformity.calibrated.uniformity[l];
                END_IF
                IF GuiSpatialUniformity.calibrated.uniformity[l]<irr_min THEN
                    irr_min := GuiSpatialUniformity.calibrated.uniformity[l];
                END_IF
            END_FOR

            IF irr_min = 0 THEN
                GuiSpatialUniformity.load_data := TRUE;
                GuiSpatialUniformity.error_message := 'Loading data...';
            ELSE
                GuiSpatialUniformity.calibrated.NU := ((irr_max-
irr_min)/(irr_max+irr_min))*100;
                sum := 0;
                FOR l:=1 TO pSpatial_Uniformity.number_of_pixels BY 1 DO
                    sum := sum +
                    adj[l] := GuiSpatialUniformity.calibrated.adj[l];
                END_FOR
                GuiSpatialUniformity.calibrated.irradiance_AVG :=
sum/pSpatial_Uniformity.number_of_pixels;
            END_IF

            Calibration_step := eSpatial_Uniformity.idle_state;

END_CASE

```

# APPENDICE B: LISTATO DELLA FUNZIONE DI CALIBRAZIONE SPETTRALE

```

//SETTAGGIO DELLE COSTANTI VISUALIZZATE DA INTERFACCIA
{attribute 'qualified_only'}
VAR_GLOBAL CONSTANT
    number_of_averages          : UINT := 20;
    environment_noise_factor    : UINT := 5;
    min_ADJ_for_updating        : INT  := 5;
// added by SMD
// added by SMD
END_VAR

//HEADER DELLA PROCEDURA, DICHIARAZIONE VARIABILI
FUNCTION_BLOCK FB_Lamp
VAR_INPUT
    iDoublePulseEnabled        : BOOL;
END_VAR
VAR_OUTPUT
    oFlash1000w                : BOOL;
    oFlash500w                 : BOOL;
    oFlashOff                   : BOOL;
END_VAR
VAR_IN_OUT
    ioLampData                  : LampStruct;
END_VAR
VAR
    LampStepN                   : INT;
    endUntil                    : BOOL;
    firstCycle                   : BOOL := TRUE; // At first cycle sent configuration to lamp
    indxZx                      : INT;
    indxwx                      : INT;
    indxPx                      : INT;
    counter                     : UINT;
    ph_sum                      : ARRAY[1..K_LAMP_DRIVER_N] OF UINT;
    a                           : INT;
    print_noise                 : BOOL;
// added by SMD
// added by SMD
// added by SMD

// *****
// Check min Version
fbCheckVer                      : FB_LampCheckMinVersion;
_seqCheckVer                    : SEQUENCE_DATA;
// *****
// Get Configuration
fbGetConfig                     : FB_LampGetConfig;
_seqGetConfig                   : SEQUENCE_DATA;
// *****
// Set Configuration
fbSetConfig                     : FB_LampSetConfig;
_seqSetConfig                   : SEQUENCE_DATA;
// *****
// Flash
fbFlash                         : FB_LampFlash;
_seqFlash                       : SEQUENCE_DATA;
// *****
// Uniformity Update / Adjust
tnDelayTime                    : TON;
nZoneToFlash                   : INT; // Number of zone to involve, based on
Lamp Type
UniformityDataMatrix_Zxwx      : ARRAY[1..K_LAMP_ZONE_N,1..K_LAMP_WAVELENGTH_N] OF
ARRAY[1..K_LAMP_DRIVER_N] OF UINT;
error_reading                   : ARRAY[1..K_LAMP_ZONE_N,1..K_LAMP_WAVELENGTH_N] OF ARRAY[1..K_LAMP_DRIVER_N]
OF UINT;
minimo                         : ARRAY[1..K_LAMP_DRIVER_N] OF UINT;
massimo                        : ARRAY[1..K_LAMP_DRIVER_N] OF UINT;

fbEnumStringNumber             : FB_EnumStringNumbers;
arrNumExtracted                : ARRAY[1..50] OF UINT;
indxNum                        : INT;
photodiodeRead                 : UINT;
compensationToApply            : LREAL;

// File management
fbNtGetTime                    : NT_GetTime;
curTime                        : TIMESTRUCT;
fbFormatString                 : FB_FormatString;
fbFileOpen                     : FB_FileOpen;
fileHandle                    : UINT;
fbFilePuts                     : FB_FilePuts;
fbFileClose                    : FB_FileClose;
fbFileGets                     : FB_FileGets;
oErrorFileMng                  : ADS_ALMS_MSGS;

```

```

// *****
//      Endless Flash
tnEndlessDelay      : TON;

// *****
//      Alarm / warning retrieved by SDO
fbSdoReadAlmsMsgs: FB_LampSdoReadAlmsMsgs;

// *****
//      Lamp status ADS read      // PSR 2.1_0o
readInfoStepN      : INT;
tnTimeOutReadInfo  : TON;
fbEcCoESdoReadInfoEx : FB_ECcOESdoReadEx;
_tmpIndx           : INT;

// *****
//      Diagnostic
almIndx            : UDINT;
almStuct_NX       : ARRAY[K_AlmStartIndx..K_AlmEndIndx] OF ADS_ALMS_MSGS;
msgIndx           : UDINT;
msgStuct_NX       : ARRAY[K_MsgStartIndx..K_MsgEndIndx] OF ADS_ALMS_MSGS;

(* DELETE_ME -> *)
TestWaven        : INT := 4;
changewave       : BOOL := FALSE;

cReset           : BOOL;
(* DELETE_ME <- *)

END_VAR
VAR CONSTANT
K_SourceID              : UDINT := 30010;

K_AlmStartIndx          : UDINT := 1;
K_AlmEndIndx            : UDINT := 60;
K_MsgStartIndx          : UDINT := 61;
K_MsgEndIndx            : UDINT := 100;
K_AdsInfoReadTime      : TIME := T#0.5S;
END_VAR

//CORPO DELLA PROCEDURA
//inizializzazione delle variabili, reset della lampada
fbSdoReadAlmsMsgs.iReset := FALSE;

IF cReset THEN
    cReset := FALSE;
    readInfoStepN := 0;
    LampStepN := 100;
END_IF

IF ioLampData.C_Reset THEN

    // Reset Commands
    ioLampData.C_Reset := FALSE;
    ioLampData.C_Flash := FALSE;
    ioLampData.C_SetConfig := FALSE;
    ioLampData.C_GetConfig := FALSE;
    ioLampData.C_UniformityGetStatus := FALSE;
    ioLampData.C_UniformityAdjust := FALSE; // PSR 2.0_01
    ioLampData.C_EndlessFlash := FALSE;

    ioLampData.do_FlashOn := FALSE;
    ioLampData.do_LoadPulse := FALSE;

    ioLampData.R_Busy := TRUE;
    ioLampData.print_report := FALSE; //added by SMD
    ioLampData.abort := FALSE; //added by SMD
    ioLampData.accept := FALSE; //added by SMD
    ioLampData.flag_noise := FALSE; //added by SMD
    ioLampData.flag_ADJ := FALSE; //added by SMD
    ioLampData.string_message:= ''; // added by SMD

    // Default select to Flash all zone, all wavelength
    ioLampData.w_ZoneSelect := 0;
    ioLampData.w_waveLengthSelect := 0;

    // Reset FBS
    fbSdoReadAlmsMsgs.iReset := TRUE;

    LampStepN := 00;

    // Reset Alarms / Messages
    FUN_ClearAlmMsgFlagExt(
        ipAlmStruct      := ADR( almStuct_NX ) ,
        iAlmBufSize      := SIZEOF( almStuct_NX ) ,
        ipMsgStruct      := ADR( msgStuct_NX ) ,
        iMsgBufSize      := SIZEOF( msgStuct_NX ) );
END_IF

```

```

REPEAT
    endUntil := TRUE;
//settaggio parametri della lampada
CASE LampStepN OF
    // *****
    // Reset
    // Check version
00:
    IF NOT _seqCheckVer.iExe AND
       NOT _seqCheckVer.oBusy AND
       _seqCheckVer.oReady
    THEN
        _seqCheckVer.iExe := TRUE;
        LampStepN := LampStepN + 1;
    END_IF
01:
    IF NOT _seqCheckVer.iExe AND
       NOT _seqCheckVer.oBusy AND
       _seqCheckVer.oReady
    THEN
        IF NOT fbCheckVer.oErrorActive THEN
            LampStepN := 10;
        ELSE
            LampStepN := LampStepN + 1; // Blocked until C_Reset ( with firmware
updated )
        END_IF
    END_IF
    // First cycle force download configuration
10:
    IF firstCycle THEN
        LampStepN := LampStepN + 1;
    ELSE
        LampStepN := 100;
    END_IF
11:
    IF NOT _seqSetConfig.iExe AND
       NOT _seqSetConfig.oBusy AND
       _seqSetConfig.oReady
    THEN
        fbSetConfig.iSetGlobalIntensity := TRUE;
        fbSetConfig.iSetZoneswaveLength := TRUE;
        fbSetConfig.iSetPulseLength := TRUE;
        _seqSetConfig.iExe := TRUE;
    END_IF
    LampStepN := LampStepN + 1;
12:
    IF NOT _seqSetConfig.iExe AND
       NOT _seqSetConfig.oBusy AND
       _seqSetConfig.oReady
    THEN
        IF NOT fbSetConfig.oErrorActive THEN
            firstCycle := FALSE;
            LampStepN := 100;
        ELSE
            LampStepN := LampStepN + 1; // Blocked until C_Reset ( with firmware
updated )
        END_IF
    END_IF
//stato di inizializzazione di tutte le variabili utilizzate in misura/calibrazione. Stato di standby in attesa di
ricevere da interfaccia il comando di creazione del file day0 e calibrazione
// *****
// Command management
100:
    LampData.R_Busy := FALSE;
    ioLampData.string_message:= '';

    (* Check for commands *)
    // Flash
    IF ioLampData.C_Flash AND
       NOT _seqFlash.iExe AND
       NOT _seqFlash.oBusy AND
       _seqFlash.oReady
    THEN
        // Reset command
        ioLampData.C_Flash := FALSE;
        ioLampData.R_Busy := TRUE;
        ioLampData.do_ZoneSelect := ioLampData.w_ZoneSelect;
        ioLampData.do_wavelengthSelect := ioLampData.w_waveLengthSelect;
        fbFlash.i500wPulseEnabled := iDoublePulseEnabled;
        // Start flash subsequence
        _seqFlash.iExe := TRUE;
        LampStepN := 200;
        endUntil := FALSE;

    // Uniformity
    ELSIF ioLampData.C_UniformityGetStatus THEN
        // Reset command
        ioLampData.string_message:= 'wait for the end of the measure';
        ioLampData.C_UniformityGetStatus := FALSE;
        ioLampData.R_Busy := TRUE;
        // Set useful information

```

```

CASE ioLampData.W_LampType OF
    eLampType.LAMP_BIG          : nZoneToFlash := 32;
    eLampType.LAMP_SMALL       : nZoneToFlash := 20;
END_CASE
// Disable "Double Pulse"
fbFlash.i500wPulseEnabled := FALSE;
// Reset Data
MEMSET(
    destAddr:= ADR( UniformityDataMatrix_Zxwx )
    fillByte:= 0
    n          := SIZEOF( UniformityDataMatrix_Zxwx ) );
MEMSET(
    destAddr:= ADR( error_reading )
    fillByte:= 0
    n          := SIZEOF( error_reading ) );
LampStepN := 300;
FOR a:= 1 TO K_LAMP_DRIVER_N DO
    minimo[a]:= 4095;
    massimo[a]:=0;
END_FOR
// Endless Flash
ELIF ioLampData.C_EndlessFlash THEN
    ioLampData.R_Busy := TRUE;
    tnEndlessDelay(
        IN := FALSE
        PT := MAX( T#1S, LREAL_TO_TIME( ioLampData.W_EndleddFlashTime * 1000 ));
    LampStepN := 400;

// Get Intensity/wavelength configuration
ELIF ioLampData.C_GetConfig AND
    NOT _seqGetConfig.iExe AND
    NOT _seqGetConfig.oBusy AND
    _seqGetConfig.oReady
THEN
    ioLampData.R_Busy := TRUE;
    ioLampData.C_GetConfig := FALSE;
    _seqGetConfig.iExe := TRUE;
    LampStepN := 500;

// Set Intensity/wavelength configuration
ELIF ioLampData.C_SetConfig AND
    NOT _seqSetConfig.iExe AND
    NOT _seqSetConfig.oBusy AND
    _seqSetConfig.oReady
THEN
    ioLampData.R_Busy := TRUE;
    ioLampData.C_SetConfig := FALSE;
    ioLampData.R_ErrorLoadLampAdj := FALSE;

    // Need for reach 1460w/mq
    ioLampData.C_SetZoneInt := TRUE;

    // Start subsequence
    fbSetConfig.iSetGlobalIntensity := ioLampData.W_SetCfgGlobalInt;

    fbSetConfig.iSetZoneswaveLength := ioLampData.W_SetCfgZoneswave;
    _seqSetConfig.iExe := TRUE;
    LampStepN := 600;

// Adjust Uniformity // PSR 2.0_01
ELIF ioLampData.C_UniformityAdjust THEN
    // Reset command
    ioLampData.string_message:= 'wait for the end of the calibration';
    ioLampData.C_UniformityAdjust := FALSE;
    ioLampData.R_Busy := TRUE;
    ioLampData.flag_ADJ:= FALSE;
    // Set usefull information
    CASE ioLampData.W_LampType OF
        eLampType.LAMP_BIG          : nZoneToFlash := 32;
        eLampType.LAMP_SMALL       : nZoneToFlash := 20;
    END_CASE
    // Disable "Double Pulse"
    fbFlash.i500wPulseEnabled := FALSE;
    // Reset Data
    MEMSET(
        destAddr:= ADR( UniformityDataMatrix_Zxwx )
        fillByte:= 0
        n          := SIZEOF( UniformityDataMatrix_Zxwx ) );
    LampStepN := 700;
END_IF

//print local adjustment factors in a txt file
IF ioLampData.print_report = TRUE THEN
    LampStepN := 340;
END_IF

// stato per flash della lampada durante le normali acquisizioni I-V
200:
IF NOT _seqFlash.iExe AND
    NOT _seqFlash.oBusy AND
    _seqFlash.oReady

```

```

        THEN
            LampStepN := 100;
        END_IF

//inizio della procedura di flash di tutte le zone e di tutte le lunghezze d'onda per la creazione del file contenente
le letture dei photodiodi "day0"
300: LampStepN := 310;
// 1. Boost to nominal value ( bypassable )
310:
    // Update Intensity / wavelength correction if ioLampData.flag_ADJged
    fbSetConfig.iSetGlobalIntensity := LampData.W_UniformityResetBoost;
    fbSetConfig.iSetZoneswaveLength := LampData.W_UniformityResetBoost;
    IF LampData.W_UniformityResetBoost THEN
        MEMSET(
            destAddr:= ADR( ioLampData.W_ZoneIntensityAdj_Zx_Wx ) ,
            fillByte:= 0
        ,
            n := SIZEOF( ioLampData.W_ZoneIntensityAdj_Zx_Wx ));
    END_IF
    // Update Pulse Length to 100MS
    ioLampData.W_PulseLength := ePulseLength.LAMP_PULSE_100MS;
    fbSetConfig.iSetPulseLength := TRUE;
    // Start Update sequence
    _seqSetConfig.iExe := TRUE;
    LampStepN := LampStepN + 1;
311:// wait Download finish
    IF NOT _seqSetConfig.iExe AND
        NOT _seqSetConfig.oBusy AND
        _seqSetConfig.oReady
    THEN
        LampStepN := 320;
    END_IF

// 2. Get current Configuration value ( used to print in file )
320:
    _seqGetConfig.iExe := TRUE;
    LampStepN := LampStepN + 1;
321://wait Upload finish
    IF NOT _seqGetConfig.iExe AND
        NOT _seqGetConfig.oBusy AND
        _seqGetConfig.oReady
    THEN
        // Prepare value for next steps
        indxZx := 1;
        indxWx := 1;
        // Reset used FBs
        tnDelayTime( IN := FALSE );
        LampStepN := 330;
        // set a counter used to filter read data
        counter:=0;
    END_IF

// 3. Acquire every zone - every wavelength
330:
    ioLampData.do_ZoneSelect := indxZx;
    ioLampData.do_wavelengthSelect := indxWx;
    _seqFlash.iExe := TRUE;
    counter := counter +1;
    tnDelayTime( IN := FALSE );
    LampStepN := LampStepN + 1;
    IF LampData.abort THEN
        LampData.abort:= FALSE;
    LampStepN:= 100;
    END_IF

// questo è lo stato in cui vengono memorizzate tutte le letture dei photodiodi e dove viene calcolato il noise
factor. La struttura "error_reading" è quella che contiene l'informazione di rumorosità per ogni zona/lunghezza
d'onda/photodiodo
331:
    tnDelayTime.IN :=
        NOT _seqFlash.iExe AND
        NOT _seqFlash.oBusy AND
        _seqFlash.oReady ;
    tnDelayTime( PT := T#0.1S );
    IF tnDelayTime.Q THEN
        tnDelayTime( IN := FALSE );
        // store Photodiode value
        FOR a:= 1 TO K_LAMP_DRIVER_N DO
            ph_sum[a] := ph_sum[a] + INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]);
            IF INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]) > massimo[a] THEN
                massimo[a] := INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]) ;
            END_IF
            IF INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]) < minimo[a] THEN
                minimo[a] := INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]) ;
            END_IF
        END_FOR

        // save photodiode values and go to next wavelength or zone
        IF counter = pPhotodiodeCalibrationConstant.number_of_averages THEN
            counter:=0;
            FOR a:= 1 TO K_LAMP_DRIVER_N DO
                ph_sum[a]:= ph_sum[a]/pPhotodiodeCalibrationConstant.number_of_averages;
            END_FOR
        END_IF
    END_IF

```



```

        UniformityDataMatrix_Zxwx[indxZx,indxwx] := ph_sum;
    FOR a:= 1 TO K_LAMP_DRIVER_N DO
        error_reading[indxZx,indxwx][a] :=((massimo[a]-minimo[a])*100)/ph_sum[a];
    IF error_reading[indxZx,indxwx][a] > pPhotodiodeCalibrationConstant.environment_noise_factor THEN
        ioLampData.flag_noise:= TRUE;
    END_IF
    END_FOR
    FOR a:= 1 TO K_LAMP_DRIVER_N DO
        minimo[a]:= 4095; //impostato come valore massimo poichè il photodiode è letto
        massimo[a]:=0;
    END_FOR
    MEMSET(
        destAddr:= ADR( ph_sum )
        ,
        fillByte:= 0
        ,
        n := sizeof( ph_sum )
        );
    indxwx := indxwx + 1;
    IF indxwx > K_LAMP_WAVELENGTH_N THEN
        indxwx := 1;
        indxZx := indxZx + 1;
    END_IF
    END_IF

    // All Zone flashed
    IF indxZx > nZoneToFlash THEN
        LampStepN := 340;
    ELSIF ioLampData.flag_noise = TRUE THEN
        LampStepN:= 774;
        ioLampData.flag_noise := FALSE;
    ELSE
        LampStepN := 330;
    END_IF
    END_IF

    // 4. Save result in a file. Stati (fino a 370) per la creazione del file day0 o del report post
    //se la variabile print report è TRUE allora cambia il nome del file
    340:// Reset used FBs
    fbNtGetTime( START := FALSE );
    fbFileOpen( bExecute := FALSE );
    fbFilePuts( bExecute := FALSE );
    fbFileClose( bExecute := FALSE );

    fbNtGetTime.START := TRUE;
    LampStepN := LampStepN + 1;
    341:
    fbNtGetTime(
        NETID := ' '
        ,
        START := TRUE
        ,
        TMOUT := T#5S
        ,
        TIMESTR => curTime
        );
    IF NOT fbNtGetTime.BUSY THEN
        fbNtGetTime( START := FALSE );
        LampStepN := LampStepN + 1;
    END_IF
    IF ioLampData.print_report THEN
    342:
        fbFormatString(
            sFormat := 'D:\lamp\spectrum\LED_chains_correction_factors_%.4d%.2d%.2d-
            %.2d%.2d%.2d.txt',
            arg1 := F_WORD( curTime.wYear )
            ,
            arg2 := F_WORD( curTime.wMonth )
            ,
            arg3 := F_WORD( curTime.wDay )
            ,
            arg4 := F_WORD( curTime.wHour )
            ,
            arg5 := F_WORD( curTime.wMinute )
            ,
            arg6 := F_WORD( curTime.wSecond )
            ,
            sout => fbFileOpen.sPathName
            );
    ELSE
        fbFormatString(
            sFormat := 'C:\Plc\Data\Uniformity_Day0.txt',
            arg1 := F_WORD( curTime.wYear )
            ,
            arg2 := F_WORD( curTime.wMonth )
            ,
            arg3 := F_WORD( curTime.wDay )
            ,
            arg4 := F_WORD( curTime.wHour )
            ,
            arg5 := F_WORD( curTime.wMinute )
            ,
            arg6 := F_WORD( curTime.wSecond )
            ,
            sout => fbFileOpen.sPathName
            );
    END_IF
    LampStepN := LampStepN + 1;
    343:
    fbFileOpen(
        sNetId := ' '
        ,
        // sPathName := , Set before
        nMode := FOPEN_MODEWRITE OR FOPEN_MODETEXT,
        ePath := PATH_GENERIC
        ,
        bExecute := TRUE
        ,
        tTimeout := T#5S
        ,
        hFile => fileHandle
        );
    IF NOT fbFileOpen.bBusy THEN
        IF NOT fbFileOpen.bError THEN

```

```

                                fbFormatString(
settings --$N',                                sFormat := 'Date:$T%.4d.%.2d.%.2d - %.2d:%.2d:%.2d$N$N-- Test
                                arg1 := F_WORD( curTime.wYear )
                                arg2 := F_WORD( curTime.wMonth )
                                arg3 := F_WORD( curTime.wDay )
                                arg4 := F_WORD( curTime.wHour )
                                arg5 := F_WORD( curTime.wMinute )
                                arg6 := F_WORD( curTime.wSecond )
                                sout => fbFilePuts.sLine
);
                                fbFilePuts.bExecute := TRUE;
                                LampStepN := LampStepN + 1;
ELSE
                                oErrorFileMng.R_Active := TRUE;
                                oErrorFileMng.R_AddInfoStr := CONCAT( 'fbFileOpen, nErrId',
UDINT_TO_STRING( fbFileOpen.nErrId ));
                                LampStepN := 100;
                                END_IF
                                fbFileOpen( bExecute := FALSE );
END_IF
// Insert Settings of the test: Boost current configuration, Pulse time
344:
fbFilePuts(
                                sNetId := ''
                                hFile := fileHandle
                                // sLine := , Set before
                                bExecute:= TRUE
                                tTimeout:= T#5S
                                );
IF NOT fbFilePuts.bBusy THEN
                                IF NOT fbFilePuts.bError THEN
                                // Global Intensity
                                fbFilePuts.sLine := CONCAT( 'Global Intensity[0.1%]:$T', LREAL_TO_STRING(
ioLampData.R_LampIntensityAdj )); //
                                fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, '$N' );
                                fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, '$N' );
                                LampStepN := LampStepN + 1;
ELSE
                                oErrorFileMng.R_Active := TRUE;
                                oErrorFileMng.R_AddInfoStr := CONCAT( 'fbFilePuts, nErrId',
UDINT_TO_STRING( fbFilePuts.nErrId ));
                                LampStepN := 100;
                                END_IF
                                fbFilePuts( bExecute := FALSE );
END_IF
345:
fbFilePuts(
                                sNetId := ''
                                hFile := fileHandle
                                // sLine := , Set Before
                                bExecute:= TRUE
                                tTimeout:= T#5S
                                );
IF NOT fbFilePuts.bBusy THEN
                                IF NOT fbFilePuts.bError THEN
                                indxZx := 1;
                                LampStepN := LampStepN + 1;
ELSE
                                oErrorFileMng.R_Active := TRUE;
                                oErrorFileMng.R_AddInfoStr := CONCAT( 'fbFilePuts, nErrId',
UDINT_TO_STRING( fbFilePuts.nErrId ));
                                LampStepN := 100;
                                END_IF
                                fbFilePuts( bExecute := FALSE );
END_IF
346:
// Compose Zone wavelength Adjustment string
fbFilePuts.sLine := CONCAT( 'Zone_', INT_TO_STRING( indxZx ));
fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, ' wavelength[%]:');
FOR indxWx := 1 TO K_LAMP_WAVELENGTH_N DO
                                fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, CONCAT( '$T', LREAL_TO_STRING(
ioLampData.R_ZoneIntensityAdj_Zx_Wx[indxZx][indxWx] )));
END_FOR
                                fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, '$N' );
                                LampStepN := LampStepN + 1;
347:
fbFilePuts(
                                sNetId := ''
                                hFile := fileHandle
                                // sLine := , Set Before
                                bExecute:= TRUE
                                tTimeout:= T#5S
                                );
IF NOT fbFilePuts.bBusy THEN
                                IF NOT fbFilePuts.bError THEN
                                indxZx := indxZx + 1;
                                IF indxZx > K_LAMP_ZONE_N THEN
                                // Saved all information
                                LampStepN := 350;
ELSE
                                // Save next information
                                LampStepN := LampStepN - 1;
END_IF
END_IF

```

```

ELSE
    oErrorFileMng.R_Active := TRUE;
    oErrorFileMng.R_AddInfoStr := CONCAT( 'fbFilePuts, nErrId',
UDINT_TO_STRING( fbFilePuts.nErrId ));
    LampStepN := 100;
END_IF
fbFilePuts( bExecute := FALSE );
END_IF

IF ioLampData.print_report AND indxZx > K_LAMP_ZONE_N THEN
    LampStepN:= 350;
    ioLampData.print_report := FALSE;
END_IF
// Insert Data retrieved
350:
fbFilePuts(
    sNetId := ' ',
    hFile := fileHandle ,
    sLine := '$N-- Test Result --
$NZoneN$TwaveLength$TPhoto1$TPhoto2$TPhoto3$TPhoto4$TPhoto5$TPhoto6$TPhoto7$TPhoto8$TPhoto9$TPhoto10$N' ,
    bExecute:= TRUE
    tTimeout:= T#5S
);
IF NOT fbFilePuts.bBusy THEN
    IF NOT fbFilePuts.bError THEN
        indxZx := 1;
        indxWx := 1;
        LampStepN := LampStepN + 1;
    ELSE
        oErrorFileMng.R_Active := TRUE;
        oErrorFileMng.R_AddInfoStr := CONCAT( 'fbFilePuts, nErrId',
UDINT_TO_STRING( fbFilePuts.nErrId ));
        LampStepN := 100;
    END_IF
    fbFilePuts( bExecute := FALSE );
END_IF
351:// Compose string to insert
fbFilePuts.sLine := INT_TO_STRING( indxZx );
fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, CONCAT( '$T', INT_TO_STRING( indxWx )));
IF print_noise THEN
    print_noise:= FALSE;
    FOR indxPx := 1 TO K_LAMP_DRIVER_N DO
        fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, CONCAT(
'$T', UINT_TO_STRING( error_reading[indxZx,indxWx][indxPx])) );
    END_FOR
ELSE
    FOR indxPx := 1 TO K_LAMP_DRIVER_N DO
        fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, CONCAT(
'$T', UINT_TO_STRING( UniformityDataMatrix_zxwx[indxZx,indxWx][indxPx])) );
    END_FOR
END_IF
fbFilePuts.sLine := CONCAT( fbFilePuts.sLine, '$N' );
LampStepN := LampStepN + 1;
352:
fbFilePuts(
    sNetId := ' ',
    hFile := fileHandle ,
    sLine := , Set Before
    bExecute:= TRUE
    tTimeout:= T#5S
);
IF NOT fbFilePuts.bBusy THEN
    IF NOT fbFilePuts.bError THEN
        // Save next information
        indxWx := indxWx + 1;
        IF indxWx > K_LAMP_WAVELENGTH_N THEN
            indxWx := 1;
            indxZx := indxZx + 1;
        END_IF

        IF indxZx > nZoneToFlash THEN
            // All information saved
            LampStepN := 355;
        ELSE
            LampStepN := 351;
        END_IF
    ELSE
        oErrorFileMng.R_Active := TRUE;
        oErrorFileMng.R_AddInfoStr := CONCAT( 'fbFilePuts, nErrId',
UDINT_TO_STRING( fbFilePuts.nErrId ));
        LampStepN := 100;
    END_IF
    fbFilePuts( bExecute := FALSE );
END_IF
355://close file
fbFileClose(
    sNetId := ' ',
    hFile := fileHandle ,
    bExecute:= TRUE
    tTimeout:= T#5S
);
IF NOT fbFileClose.bBusy THEN
    IF NOT fbFileClose.bError THEN
        LampStepN := 100;
    ELSE
        oErrorFileMng.R_Active := TRUE;
        oErrorFileMng.R_AddInfoStr := CONCAT( 'fbFileClose, nErrId',
UDINT_TO_STRING( fbFileClose.nErrId ));
        LampStepN := 100;

```

```

                                END_IF
                                fbFileClose( bExecute := FALSE );
                                END_IF
//stati già esistenti, gestione lampada
// *****
//      Endless Flash
400:
    tnEndlessDelay( IN := TRUE );
    // Command removed
    IF NOT ioLampData.C_EndlessFlash THEN
        LampStepN := 100;

    ELSIF // Time to Flash
            tnEndlessDelay.Q AND
            // Ready to Flash
            NOT _seqFlash.iExe          AND
            NOT _seqFlash.oBusy        AND
            _seqFlash.oReady
    THEN
        tnEndlessDelay( IN := FALSE );
        ioLampData.do_ZoneSelect := ioLampData.W_ZoneSelect;
        ioLampData.do_WavelengthSelect := ioLampData.W_waveLengthSelect;
        _seqFlash.iExe := TRUE;
    END_IF

// *****
//      Get Configuration
500:
    IF      NOT _seqGetConfig.iExe          AND
            NOT _seqGetConfig.oBusy        AND
            _seqGetConfig.oReady
    THEN
        LampStepN := 100;
    END_IF

// *****
//      Set Configuration
600:
    IF      NOT _seqSetConfig.iExe          AND
            NOT _seqSetConfig.oBusy        AND
            _seqSetConfig.oReady
    THEN
        ioLampData.R_ErrorLoadLampAdj := fbSetConfig.oErrorActive;
        LampStepN := 100;
    END_IF

// stati per la procedura di calibrazione (fino a 770). La parte di gestione lampada, lettura file è stata solamente
// debuggata poiché non funzionante in principio. La parte di calcolo è stata riscritta ex-novo.

// *****
//      Adjust Uniformity // PSR 2.0_01
// 1. Get current compensation
// 2. Set flash duration to 100[ms]
// 3. Open "Day0" file
// 4. Get "Day0" photodiode reading
// 5. Flash, Compensate
// 6. Close "Day0" file

700: LampStepN := 710;

// 1. Get current compensation
710:
    _seqGetConfig.iExe := TRUE;
    LampStepN := LampStepN + 1;
711:
    IF      NOT _seqGetConfig.iExe  AND
            NOT _seqGetConfig.oBusy AND
            _seqGetConfig.oReady
    THEN
        LampStepN := 720;
    END_IF

// 2. Set flash duration to 100[ms]
720:
    fbSetConfig.iSetPulseLength := TRUE;
    _seqSetConfig.iExe := TRUE;
    LampStepN := LampStepN + 1;
721:
    IF      NOT _seqSetConfig.iExe  AND
            NOT _seqSetConfig.oBusy AND
            _seqSetConfig.oReady
    THEN
        LampStepN := 730;
    END_IF
// 3. Open "Day0" file
730:// Reset used FBS
    fbFileOpen( bExecute := FALSE );
    fbFilePuts( bExecute := FALSE );
    fbFileClose( bExecute := FALSE );
    fbFileGets( bExecute := FALSE );
    LampStepN := LampStepN + 1;
731:

```

```

fbFileOpen(
    sNetId      := ''
    sPathName   := 'C:\Plc\Data\Uniformity_Day0.txt',
    nMode       := FOPEN_MODEREAD OR FOPEN_MODETEXT,
    ePath       := PATH_GENERIC
    bExecute := TRUE
    tTimeout := T#5S
    hFile       => fileHandle
);
IF NOT fbFileOpen.bBusy THEN
    IF NOT fbFileOpen.bError THEN
        LampStepN := 740;
    ELSE
        almStuct_NX[56].R_Active := TRUE;
        almStuct_NX[56].R_AddInfoStr := CONCAT( 'fbFileOpen, nErrId',
UDINT_TO_STRING( fbFileOpen.nErrId ));
        LampStepN := 100;
    END_IF
    fbFileOpen( bExecute := FALSE );
END_IF

// 4. Get "Day0" photodiode reading
740:
// Move to first Interesting position
fbFileGets(
    sNetId := ''
    hFile := fileHandle
    bExecute:= TRUE
    tTimeout:= T#5S
);
IF NOT fbFileGets.bBusy THEN
    IF NOT fbFileGets.bError AND
    NOT fbFileGets.bEOF
    THEN
        IF FIND( fbFileGets.sLine, '1$t1$t' ) > 0 THEN
            LampStepN := LampStepN + 1;
        END_IF
    ELSIF fbFileGets.bEOF THEN
        almStuct_NX[56].R_Active := TRUE;
        almStuct_NX[56].R_AddInfoStr := 'END of File reached';
        LampStepN := 770;// goto 7. Close "Day0" file
    ELSE
        almStuct_NX[56].R_Active := TRUE;
        almStuct_NX[56].R_AddInfoStr := CONCAT( 'fbFileGets, nErrId: ',
UDINT_TO_STRING( fbFileGets.nErrId ));
        LampStepN := 770;// goto 7. Close "Day0" file
    END_IF
    fbFileGets( bExecute := FALSE );
END_IF

741:
indxNum := 1;
fbEnumStringNumber(
    sSearch := fbFileGets.sLine
    eCmd := E_EnumCmdType.eEnumCmd_First
    eType := eNumGroup_Unsigned
);
WHILE NOT fbEnumStringNumber.bEOS DO
    IF fbEnumStringNumber.nPos <> 0 THEN
        arrNumExtracted[indxNum] := STRING_TO_UINT( fbEnumStringNumber.sNumber );
        indxNum := indxNum + 1;
    END_IF
    fbEnumStringNumber( eCmd := E_EnumCmdType.eEnumCmd_Next );
END_WHILE
LampStepN := 750;
counter:=0;
MEMSET(
    destAddr:= ADR( ph_sum )
    fillByte:= 0
    n := SIZEOF( ph_sum )
);

// 5. Flash, Compensate
750:
// Flash that zone / wavelength for a pphotodiodeCalibrationConstant.number_of_averages
number of time
indxZx := ioLampData.do_ZoneSelect := UINT_TO_INT(arrNumExtracted[1]);
indxwx := ioLampData.do_wavelengthSelect := UINT_TO_INT(arrNumExtracted[2]);
IF NOT _seqSetConfig.iExe AND
    NOT _seqSetConfig.oBusy AND
    _seqSetConfig.oReady
THEN
    tnDelayTime.IN := //SMDDMS
    NOT _seqFlash.iExe AND
    NOT _seqFlash.oBusy AND
    _seqFlash.oReady ;
tnDelayTime( PT := T#0.1S );
IF tnDelayTime.Q THEN
    tnDelayTime( IN := FALSE );
    _seqFlash.iExe := TRUE;
    counter := counter + 1;
    LampStepN := LampStepN + 1;
END_IF
END_IF

751:
IF // Flash done
    NOT _seqFlash.iExe AND
    NOT _seqFlash.oBusy AND
    _seqFlash.oReady
THEN

```

```

FOR a:= 1 TO K_LAMP_DRIVER_N DO
    ph_sum[a] := ph_sum[a] + INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]);
    IF INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]) > massimo[a] THEN
        massimo[a] := INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]) ;
    END_IF
    IF INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]) < minimo[a] THEN
        minimo[a] := INT_TO_UINT(ioLampData.ai_PhotoDiode_Dx[a]) ;
    END_IF
END_FOR

IF counter = pPhotodiodeCalibrationConstant.number_of_averages THEN
    counter:= 0;
    FOR a:= 1 TO K_LAMP_DRIVER_N DO
        ph_sum[a]:=
ph_sum[a]/pPhotodiodeCalibrationConstant.number_of_averages;
    END_FOR
    UniformityDataMatrix_Zxwx[indxZx,indxwx] := ph_sum;
    FOR a:= 1 TO K_LAMP_DRIVER_N DO
        error_reading[indxZx,indxwx][a] :=((massimo[a]-minimo[a])*100)/ph_sum[a];
    IF error_reading[indxZx,indxwx][a] > pPhotodiodeCalibrationConstant.environment_noise_factor
THEN
        ioLampData.flag_noise:= TRUE;
    END_IF
    END_FOR
    FOR a:= 1 TO K_LAMP_DRIVER_N DO
        minimo[a]:= 4095;
        massimo[a]:=0;
    END_FOR

    // Compare Photodiode ( current ) respect file ( expected ) and compensate
    indxPx := (( indxZx + 1 ) / 2) + 2;
    photodiodeRead := ph_sum[indxPx-2];
    compensationToApply := (( UINT_TO_LREAL( arrNumExtracted[indxPx] ) /
    UINT_TO_LREAL( photodiodeRead )) - 1.0 ) * 100;
    // Apply compensation
    IF ABS(compensationToApply)>
pPhotodiodeCalibrationConstant.min_ADJ_for Updating THEN
        ioLampData.flag_ADJ := TRUE;
    END_IF
    IF compensationToApply < -15 THEN
        compensationToApply := -15;
        LampStepN := 762;
    ELSIF compensationToApply > 60 THEN
        compensationToApply := 60;
        LampStepN := 762;
    END_IF
    ioLampData.w_ZoneIntensityAdj_Zx_wx[indxZx][indxwx] :=
        ioLampData.w_ZoneIntensityAdj_Zx_wx[indxZx][indxwx]
        +
        LREAL_TO_INT( compensationToApply );
    fbSetConfig.iSetGlobalIntensity := FALSE;
    fbSetConfig.iSetZonesWaveLength := FALSE;
    fbSetConfig.iSetPulseLength := TRUE;
    _seqSetConfig.iExe := TRUE;
    LampStepN := 762;

    ELSE
        LampStepN := 750;
    END_IF
END_IF
IF ioLampData.flag_noise THEN
    lampStepN:= 770;
    END_IF
762:
    fbFileGets(
        sNetId := '' ,
        hFile := fileHandle ,
        bExecute:= TRUE ,
        tTimeout:= T#5S );
    IF NOT fbFileGets.bBusy THEN
        IF NOT fbFileGets.bError AND
        NOT fbFileGets.bEOF
        THEN
            // Extract number from string
            LampStepN := 741;
        ELSIF fbFileGets.bEOF THEN
            LampStepN := 770;// goto 7. Close "Day0" file
        ELSE
            almStuct_NX[56].R_Active := TRUE;
            almStuct_NX[56].R_AddInfoStr := CONCAT( 'fbFileGets, nErrId: ',
UDINT_TO_STRING( fbFileGets.nErrId ));
            LampStepN := 770;// goto 7. Close "Day0" file
        END_IF
        fbFileGets( bExecute := FALSE );
    END_IF
// 7. Close "Day0" file
770:
    fbFileClose(
        sNetId := '' ,
        hFile := fileHandle ,
        bExecute:= TRUE ,
        tTimeout:= T#5S );
    IF NOT fbFileClose.bBusy THEN

```

```

        IF NOT fbFileClose.bError THEN
            LampStepN := 100;
        ELSE
            oErrorFileMng.R_Active := TRUE;
            oErrorFileMng.R_AddInfoStr := CONCAT( 'fbFileClose, nErrId',
UDINT_TO_STRING( fbFileClose.nErrId ));
            LampStepN := 100;
        END_IF
        fbFileClose( bExecute := FALSE );
    END_IF
    LampStepN := 771;
    IF ioLampData.flag_noise = TRUE THEN
        LampStepN:= 774;
    END_IF
    LampData.print_report:= TRUE;

//parte relativa alla gestione dei risultati ottenuti in calibrazione. Vengono printati i messaggi che
indicano se lo spettro necessita di essere corretto o meno oppure se la misura è stata troppo rumorosa.
771:
ioLampData.accept:= TRUE;
IF NOT ioLampData.flag_ADJ THEN
    ioLampData.string_message:= 'Spectrum does not need to be corrected';
    IF ioLampData.accept THEN
        ioLampData.string_message:= '';
        LampStepN:= 772;
    END_IF
ELSE
    ioLampData.string_message:= 'Spectrum needs to be corrected. Update spectrum?';
    IF ioLampData.update_spectrum = 1 THEN // 1: update adj matrix
        ioLampData.string_message:= '';
        LampStepN:= 773;
    ELSIF ioLampData.update_spectrum = 2 THEN // 2: don't update adj matrix
        ioLampData.string_message:= '';
        LampStepN:= 772;
        ioLampData.flag_ADJ:= FALSE;
    END_IF
END_IF
//ripristino i fattori di aggiustamento
772:
ioLampData.accept:= FALSE;
FOR indxZx:=1 TO K_LAMP_ZONE_N BY 1 DO
    FOR indxwx:=1 TO K_LAMP_WAVELENGTH_N BY 1 DO
        ioLampData.W_ZoneIntensityAdj_Zx_Wx[indxZx][indxwx]:=
ioLampData.R_ZoneIntensityAdj_Zx_Wx[indxZx][indxwx];
    END_FOR
END_FOR
lampdata.C_SetConfig := TRUE;
ioLampData.update_spectrum:= 0;
LampStepN:= 100;
773:
ioLampData.accept:= FALSE;
lampdata.C_SetConfig := TRUE;
ioLampData.update_spectrum:= 0;
LampStepN:=100;
//il delay di 5 secondi presente è per fare in modo che la stringa di errore si resettì da sola dopo tale time-out
774:
ioLampData.string_message:= 'Procedure interrupted: too much environmental noise';
tnDelayTime.IN := TRUE;
tnDelayTime( PT := T#5S );
IF tnDelayTime.Q THEN
    tnDelayTime( IN := FALSE );
    ioLampData.string_message:= '';
    ioLampData.flag_noise := FALSE;
    LampStepN:= 772;
END_IF

//fine del case degli stati. Parte già esistente
END_CASE
UNTIL endUntil // PAY ATTENTION: Exit loop done when == TRUE
END_REPEAT

IF ioLampData.di_FlashON THEN
    fbMeasurementsTime.mIVSetStartTime( iMeasType := K_BCSSUN_LAMP );
ELSIF BCSSun_Meas.MeasureTimeData[K_BCSSUN_LAMP,0] > 0 THEN
    fbMeasurementsTime.mIVSetEndTime( iMeasType := K_BCSSUN_LAMP );
END_IF

A_SubSequences();
A_Diagn();

```





## INDICE DELLE FIGURE

2.1	Struttura generale del simulatore solare AMAT.....	3
1.2	Scope dei segnali digitali in fase di acquisizione.....	4
1.3	Misura di efficienza di una cella in silicio cristallino.....	5
2.1	Definizione di AM.....	12
2.2	Intensità della luce solare per lo standard AM 1.5.....	12
2.3	Esempio di correzione in irradianza e temperatura di una caratteristica I-V.....	17
3.1	Modulo contenente il simulatore solare di AppliedMaterials.....	21
3.2	Modulo contenente il simulatore solare di AppliedMaterials senza pannelli protettivi.....	22
3.3	Cella di riferimento e sistema di contattazione.....	23
3.4	Carico elettronico collegato al sistema di contattazione.....	23
3.5	Singolo modulo della lampada.....	25
3.6	Pannello posteriore della lampada con collegamenti al chiller.....	25
3.7	Pannello frontale del carico elettronico.....	26
3.8	Modello circuitale semplificato per il carico elettronico connesso alla cella.....	27
3.9	Definizione della ricetta di acquisizione da interfaccia grafica.....	27
3.10	Segnali di temperatura e irradianza letti da cella di riferimento.....	28
3.11	Calibrazione del simulatore solare.....	29
3.12	Modello a due diodi della cella solare.....	31
3.13	Acquisizione I-V durante la prova di dark shunt.....	31
3.14	Calcolo della tensione di circuito aperto.....	32
3.15	Calcolo della corrente di cortocircuito.....	33
3.15	Calcolo della resistenza serie $R_s$ .....	34
4.1	Diagramma di flusso rappresentativo della procedura di calibrazione della lampada LED.....	36
4.2	Griglia di misura per uniformità spaziale.....	38
4.3	Andamento della temperatura della cella di riferimento nell'arco di 20 acquisizioni.....	39
4.4	Setup sperimentale dei test per la misura di uniformità spaziale.....	40
4.5	Diagramma di flusso dell'algorithmo di minimizzazione della non-uniformità.....	47
4.6	Simmetria dei valori di irradianza sul piano dell'area sottostante la lampada.....	48
4.7	Diagramma di flusso dell'algorithmo con vettore target.....	50
4.8	Relazione tra diminuzione di NU e ampiezza range del vettore <b>ADJ</b> .....	52

4.9	Relazione tra irradianza media e ampiezza range del vettore <b>ADJ</b> .....	54
4.10	Codice sorgente TwinCat, lista degli stati.....	56
4.11	Diagramma a stati dell'implementazione TwinCat della procedura di calibrazione di uniformità spaziale.....	57
4.12	Interfaccia grafica per l'utente per la funzione di calibrazione di uniformità spaziale.....	58
4.13	Finestra di dialogo per lo spostamento della cella di riferimento.....	59
5.1	Spettro della lampada del simulatore solare.....	64
5.2	Andamento di 5 fotodiodi durante le acquisizioni.....	65
5.3	Andamento di 5 fotodiodi durante il test per la linearità.....	67
5.4	Diagramma di flusso per acquisizione valori digitali dei fotodiodi.....	72
5.5	Diagramma degli stati per la procedura di correzione tramite fotodiodi.....	74
5.6	Sezione del file di testo di calibrazione.....	75
5.7	Spettro acquisito prima e dopo la correzione.....	77
5.8	Output della procedura di correzione spettrale.....	78
6.1	Risposta spettrale di una cella fotovoltaica esposta ad uno spettro non stabile nel tempo.....	80
6.2	Misure seriali di efficienza prima e dopo la calibrazione del sistema.....	81

## INDICE DELLE TABELLE

2.1	Classificazione di un simulatore solare in base al parametro di non-uniformità di irradianza.....	10
2.2	Spettro solare in AM 1.5 diviso in bande.....	13
2.3	Classificazione di un simulatore solare in base al matching spettrale.....	13
3.1	Composizione dello spettro della lampada.....	24
3.2	Elenco delle prove effettuate su singola cella campione.....	30
4.1	Irradianza e temperatura cella di riferimento per 20 acquisizioni.....	39
4.2	Valori di irradianza e parametro NU per tre test.....	41
4.3	Risoluzione della misura del parametro NU.....	42
4.4	Verifica applicazione sovrapposizione degli effetti.....	45
4.5	Elementi di $U$ misurati per il confronto degli algoritmi.....	51
4.6	Confronto delle prestazioni degli algoritmi con range $[-10\%, 0\%]$ .....	53
4.7	Caratteristiche delle prove per uniformità spaziale.....	60
4.8	Risultati delle prove per uniformità spaziale.....	60
5.1	Valori di ripetibilità in lettura dei fotodiodi.....	66
5.2	Linearità tra valori letti dai fotodiodi e fattori di aggiustamento.....	68
5.3	Applicazione dell'equazione (5.2) senza filtraggi.....	69
5.4	Applicazione dell'equazione (5.2) con media aritmetica su 20 acquisizioni.....	70
6.1	Test di accuratezza prima e dopo la calibrazione.....	81



## RIFERIMENTI BIBLIOGRAFICI

- [1] Comitato elettrotecnico italiano, norma italiana EN 60904-1, “Parte 1: Misura delle caratteristiche fotovoltaiche corrente-tensione”, 2008, seconda edizione, fascicolo 9297.
- [2] International electrotechnical commission, IEC 60904-9, “Photovoltaic devices – Part 9: Solar simulator performance requirements”, 2007, edition 2.0.
- [3] Christiana Honsberg and Stuart Bowden, PV education.org, “Properties of Sunlight: Air mass”, 2019 [online]. Disponibile: <https://www.pveducation.org/pvcdrom/properties-of-sunlight/%EC%97%90%EC%96%B4%EB%A7%A4%EC%8A%A4-air-mass>. [consultato in data 03 Novembre 2019].
- [4] Christiana Honsberg and Stuart Bowden, PV education.org, “Properties of Sunlight: Atmospheric-effects”, 2019 [online]. Disponibile: <https://www.pveducation.org/pvcdrom/properties-of-sunlight/%EB%8C%80%EA%B8%B0-%ED%9A%A8%EA%B3%BC-atmospheric-effects>. [consultato in data 03 Novembre 2019].
- [5] International electrotechnical commission, IEC 60891, “Photovoltaic devices – Procedures for temperature and irradiance corrections to measured I-V characteristics”, 2009-2012, edition 2.0.