

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Meccanica

a.a. 2023/2024



**Politecnico  
di Torino**

Tesi Magistrale

**REALIZZAZIONE DI UN DIGITAL TWIN  
METROLOGICO PER LA NANOINDENTAZIONE**

Relatori:

Prof. Gianfranco GENTA

Dott. Giacomo MACULOTTI

Candidata:

Ilaria BOLDRINO

s304954



# Abstract

I digital twin (DT) sono una tecnologia abilitante per l'industria 4.0. Essi permettono la predizione del comportamento di un sistema sulla base di dati storici, missione e acquisizione in tempo reale. Tale predizione, se evidenzia distorsioni dal comportamento ideale, permette, tramite logiche di controllo a circolo chiuso, la correzione del comportamento del sistema e la prevenzione di malfunzionamenti. In letteratura sono rari i casi di applicazioni di DT a strumenti e processi di misura. La tesi si pone l'obiettivo di studiare e implementare un DT della prova di penetrazione strumentata sulla nano scala. In particolare, si andrà ad analizzare il concetto di digital twin, evidenziando come possa migliorare la nanoindentazione. Saranno individuate le principali cause degli errori di misura, cercando di individuare il miglior metodo per identificarle e correggerle. Quindi, si andrà a sviluppare un modello in grado di creare un dataset di training da una serie di indentazioni base. Creato questo ambiente, si procederà dunque al twinning, ossia al collegamento del modello digitale alla macchina. Ad ogni nuova indentazione, il DT andrà a verificare la correttezza delle caratteristiche ricavate dalla misurazione e verificherà la correttezza della misurazione stessa, controllando l'eventuale presenza di errori nella procedura di misurazione. Il modello creato si pone l'obiettivo di individuare i problemi, implementando i calcoli in concordanza con gli standard posti dalla "Guide to the expression of Uncertainty in Measurement", in modo da creare un valido strumento metrologico.



# Indice

Indice delle Figure .....	VII
Indice delle Tabelle .....	IX
Abbreviazioni.....	X
<b>1. Introduzione.....</b>	<b>1</b>
<b>2. Il digital twin per la nanoindentazione.....</b>	<b>3</b>
2.1 Struttura generale di un DT.....	3
2.1.1 Le connessioni P2V e V2P .....	4
2.2 La nanoindentazione.....	7
2.2.1 Cenni storici .....	7
2.2.2 Procedura .....	8
2.2.3 Errori di misura.....	14
2.2.3.1 <i>Mantenimento troppo breve.....</i>	<i>14</i>
2.2.3.2 <i>Deriva termica .....</i>	<i>15</i>
2.2.3.3 <i>Vibrazioni.....</i>	<i>16</i>
2.2.3.4 <i>Usura della punta .....</i>	<i>19</i>
2.2.3.5 <i>Cattivo montaggio .....</i>	<i>20</i>
<b>3. Metodologia .....</b>	<b>22</b>
3.1 Acquisizione dati .....	23
3.2 Elaborazione dei dati: P2V.....	26
3.2.1 La divisione dell'indentazione in fasi .....	27
3.2.2 La regressione dei dati per la creazione del modello .....	29
3.3 La caratterizzazione: il calcolo delle proprietà meccaniche .....	30
3.3.1 Il calcolo di $S_m$ e $S$ .....	31
3.3.2 Calcolo della durezza e del modulo elastico d'indentazione.....	31
3.4 La definizione dell'incertezza di misura .....	33
3.4.1 Cenni teorici .....	33
3.4.1.1 <i>Il calcolo dell'incertezza composta.....</i>	<i>34</i>
3.4.1.2 <i>L'incertezza estesa.....</i>	<i>37</i>

3.4.2	Determinazione dei contributi di incertezza delle variabili in input	39
3.4.3	L'incertezza estesa per le fasi di carico e scarico.....	41
3.4.4	La varianza composta nei parametri caratteristici del materiale.....	43
3.5	Test statistici.....	45
3.5.1	Il Test d'ipotesi.....	45
3.5.2	Controllo del mantenimento primario.....	47
3.5.3	Controllo del mantenimento secondario.....	48
3.5.4	Test d'ipotesi sulle proprietà caratteristiche dei materiali calcolate.....	49
<b>4.</b>	<b>Risultati.....</b>	<b>51</b>
4.1	Divisione dei dati sperimentali .....	51
4.2	La regressione e l'intervallo di predizione .....	53
4.3	Calcolo delle proprietà meccaniche del materiale.....	57
4.4	I test d'ipotesi.....	59
4.4.1	Test sulle caratteristiche meccaniche .....	59
4.4.2	Test sui mantenimenti .....	60
<b>5.</b>	<b>Conclusioni e sviluppi futuri.....</b>	<b>62</b>
<b>6.</b>	<b>Bibliografia.....</b>	<b>65</b>
<b>7.</b>	<b>Appendice.....</b>	<b>67</b>

# Indice delle Figure

Figura 2.1 Schema generale di un Digital Twin.....	4
Figura 2.2 Schema macchina di indentazione.....	9
Figura 2.3 Grafico $F_h$ di un'indentazione.....	11
Figura 2.4 Effetto del tempo di mantenimento sulle curve di scarico .....	14
Figura 2.5 Affondamento in funzione del tempo, aggiunta del mantenimento secondario.....	16
Figura 2.6 Schema della macchina con accelerometri .....	17
Figura 2.7 Ciclo indentazione 1 .....	17
Figura 2.8 Ciclo indentazione 7 .....	18
Figura 2.9 Confronto dati dell'accelerometro tra indentazione 1 e 7.....	18
Figura 2.10 Microscopia impronta Berkovich.....	19
Figura 2.11 Effetto dell'usura della punta .....	20
Figura 2.12 Effetto del cattivo montaggio dell'indentatore.....	21
Figura 3.1 Macchina di nanoindentazione del laboratorio Mind4Lab .....	24
Figura 3.2 Ciclo completo con mantenimento secondario .....	27
Figura 3.3 Forza in funzione del tempo in un ciclo.....	28
Figura 3.4 Regressione alla Distanze Ortogonali.....	30
Figura 3.5 Schema di funzionamento della funzione Scipy.ODR.....	41
Figura 3.6 Test d'ipotesi ad una coda.....	47
Figura 3.7 Test d'ipotesi a due code.....	49
Figura 4.1 Divisione dei dati grezzi di training in fasi.....	51
Figura 4.2 Ingrandimento divisione dei dati sperimentali, focus sul mantenimento primario.....	52
Figura 4.3 Divisione della curva real-time .....	52
Figura 4.4 Regressione dati fase di carico.....	53
Figura 4.5 Ingrandimento regressione fase di carico .....	54
Figura 4.6 Regressione dei dati della fase di scarico .....	54
Figura 4.7 Ingrandimento regressione fase di scarico .....	55
Figura 4.8 Ingrandimento della fase di scarico, nota su MSE.....	56

Figura 4.9 Risultato finale regressioni dell'intero ciclo.....57



# Indice delle Tabelle

Tabella 3.1	Caratteristiche dataset preliminare .....	25
Tabella 3.2	Caratteristiche dataset definitivo .....	25
Tabella 3.3	Variazione dei valori di $\varepsilon$ in base alla geometria dell'indentatore....	32
Tabella 3.4	Taratura SiO <sub>2</sub> e calcolo dell'incertezza.....	39
Tabella 3.5	Valori e varianze parametri dei parametri di input .....	41
Tabella 4.1	Proprietà meccaniche del provino derivate dai dati di training.....	57
Tabella 4.2	Proprietà meccaniche del provino derivate dai dati di test.....	58
Tabella 4.3	Risultati dei test d'ipotesi sulle proprietà meccaniche.....	59
Tabella 4.4	Risultati dei test d'ipotesi sui mantenimenti.....	61

# Abbreviazioni

DT Digital Twin

GUM Guide to the expression of Uncertainty in Measurement

P2V Physical-to-Virtual

V2P Virtual-to-Physical

ODR Orthogonal Distance Regression

NLS Nonlinear Least Square

MSE Mean Squared Error

# 1. Introduzione

La nanoindentazione è un test meccanico non convenzionale. Permette la caratterizzazione meccanica su scala nanometrica e trova applicazione in numerosi ambiti.

Questa tecnologia permette di determinare parametri come il modulo di Young, la durezza, il creep e il rilassamento dei materiali. Tuttavia, l'accuratezza delle misurazioni può essere negativamente influenzata da fattori esterni come la deriva termica e le vibrazioni. Per aumentarne la precisione e l'affidabilità, è stato introdotto il concetto di digital twin (DT), tecnologia emergente dell'industria 4.0, che rappresenta certamente una soluzione innovativa.

I digital twin sono, come dice il loro nome, “gemelli digitali” di sistemi fisici, che integrano aggiornamenti real-time basati su dati acquisiti dal sistema reale. Questo permette un maggior controllo sui fattori ambientali apportando un miglioramento della vita del sistema e garantendo una valutazione migliore delle incertezze associate alle misurazioni. Nell'ambito della ricerca metrologica, i Digital twin stanno emergendo sempre più per integrare i dispositivi di misurazione certificati, perché permettono una correzione in tempo reale di eventuali derive che andrebbero ad inficiare la procedura di misurazione.

Questo lavoro di tesi ha, quindi, l'obiettivo di sviluppare e validare un digital twin specifico per la nanoindentazione, la cui implementazione permetterà la quantificazione dell'incertezza delle misurazioni in conformità con gli standard GUM (Guide to the expression of Uncertainty in Measurement)[1,2] andando a colmare una lacuna significativa nella ricerca metrologica.

Attualmente, infatti, non esistono procedure standard per la valutazione dell'incertezza di misura tramite questa nuova tecnologia e la letteratura disponibile su questo argomento è scarsa ed insufficiente. Il lavoro svolto si inserisce, infatti, all'interno del progetto di ricerca europeo ViDiT [3,4,5], che ha come scopo quello di sviluppare tali metodi e procedure per garantire l'affidabilità

dei digital twin in metrologia per sostenere la trasformazione digitale all'interno dell'industria 4.0 e, quindi, il Green Deal europeo.

I miglioramenti ottenuti, infatti, permetteranno di prevedere errori e difetti riducendo il tasso di scarti. Questo migliorerà l'affidabilità, l'efficienza e la velocità dei processi produttivi, andando a ridurre tempi, costi e, di conseguenza, il consumo energetico, contribuendo a diminuire l'impronta di CO2 dell'Europa.

## 2. Il digital twin per la nanoindentazione

### 2.1 Struttura generale di un DT

Il concetto di Digital Twin (DT) è attribuito a Michael Grieves e al suo lavoro con John Vickers della NASA, presentato per la prima volta in una lezione sul Product Life-Cycle Management nel 2003 [6]. A quell'epoca, le rappresentazioni virtuali dei prodotti erano ancora immature e i dati raccolti sui prodotti fisici erano limitati e spesso manuali. Grieves e Vickers immaginarono un mondo in cui un modello virtuale di un prodotto potesse fornire le basi per la gestione del ciclo di vita del prodotto. La descrizione iniziale definiva un DT come una rappresentazione virtuale di un prodotto fisico, contenente informazioni dettagliate su di esso.

Negli ultimi anni, l'interesse per il concetto di Digital Twin è cresciuto notevolmente sia in ambito industriale che accademico, è stato ampiamente investigato fino ad arrivare ad un modello simulativo che riproduce accuratamente le caratteristiche, i comportamenti e le prestazioni di un sistema fisico in un ambiente virtuale. Permettendo, in particolare, un aggiornamento dinamico bidirezionale del modello virtuale a seconda dell'osservazione dello stato del sistema fisico e un controllo dell'attività fisica utilizzando dati e simulazioni in tempo reale [7,8].

Il DT, come è schematizzato in Figura 2.1, è costituito da quattro parti principali: un'entità fisica, un'entità virtuale e le connessioni sincronizzate tra le due [9].

*L'ambiente fisico* è il mondo reale, misurabile, dove esiste l'entità fisica. Gli aspetti del mondo fisico sono misurati e riportati nell'ambiente virtuale.

*L'ambiente virtuale* è l'insieme delle simulazioni che replicano lo stato dell'ambiente fisico. Esiste in un dominio digitale e vuole essere un “gemello” del mondo reale.

Le connessioni *Physical-to-Virtual (P2V)* sono il modo in cui l'entità fisica viene trasferita e realizzata nell'ambiente virtuale. La connessione avviene tramite due fasi, una dove viene catturato lo stato dell'entità fisica e l'altra in cui viene determinata la "distanza" tra le entità fisiche e digitali, per aggiornare di conseguenza l'entità virtuale.

Le connessioni *Virtual-to-Physical(V2P)* sono il flusso di informazione e processi dall'ambiente virtuale a quello fisico. I processi virtuali individuano un insieme ottimale di parametri. Viene determinato il delta tra questi nuovi valori e lo stato attuale e viene aggiornato, di conseguenza, lo stato dell'entità fisica.

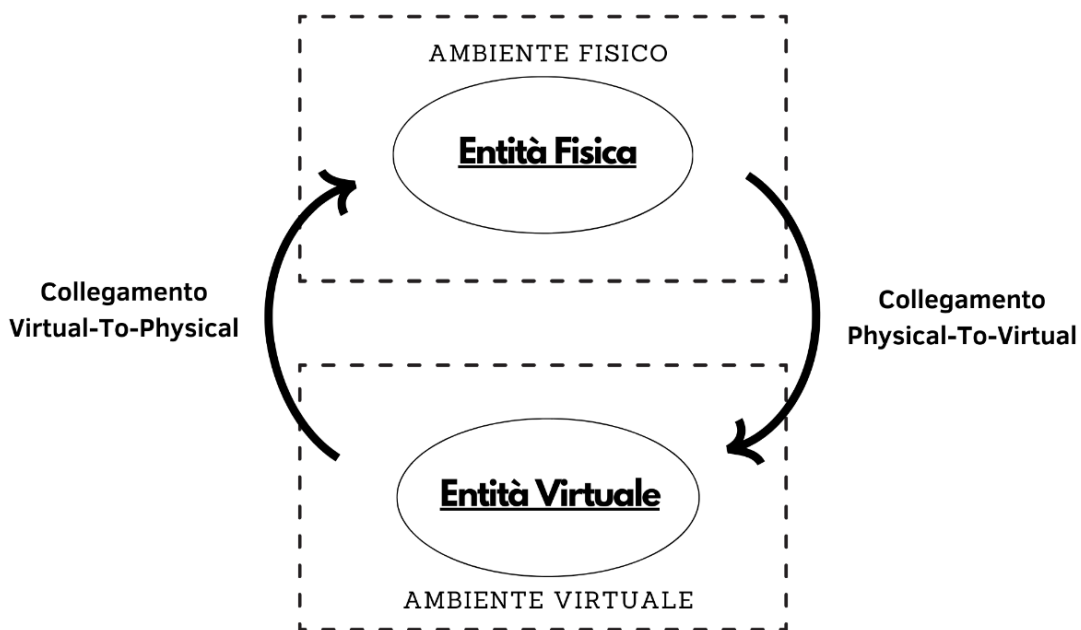


Figura 2.1 Schema generale di un Digital Twin

Riassumendo, una definizione accurata di Digital Twin può essere: un modello di simulazione che replica accuratamente i sistemi fisici in un ambiente virtuale e comprende aggiornamenti dinamici del modello virtuale, basati sullo stato osservato della sua controparte reale per ottenere il controllo fisico di quest'ultima.

### 2.1.1 Le connessioni P2V e V2P

Un aspetto cruciale del Digital Twin è l'interconnessione tra i suoi componenti fisici e virtuali. Questo collegamento bidirezionale è essenziale per assicurare che le modifiche e le informazioni fluiscano tra i due domini, permettendo al DT di aggiornare continuamente il suo stato e di fornire previsioni accurate e azioni correttive per il sistema fisico.

Le connessioni Physical-To-Virtual e Virtual-To-Physical descrivono il flusso di dati e comandi che permettono al DT di aggiornarsi continuamente e di influenzare il sistema fisico in modo efficace.

#### *Physical-to-Virtual (P2V)*

Il concetto di P2V riguarda il trasferimento di dati dal sistema fisico al gemello digitale. Questo processo coinvolge la raccolta, la trasmissione e l'elaborazione di dati provenienti dai sensori e dagli strumenti di monitoraggio che misurano vari parametri del sistema fisico, come temperatura, pressione, vibrazioni e altre condizioni operative.

I sensori posizionati sul sistema fisico raccolgono dati in tempo reale su diversi parametri operativi. Questi dati possono includere misurazioni ambientali, stato delle componenti, performance operative e altre metriche rilevanti.

Questi vengono trasmessi al modello virtuale attraverso reti di comunicazione e una volta ricevuti, vengono elaborati e integrati nel modello virtuale. Questo processo può includere la pulizia dei dati, la fusione dei dati da diverse fonti e l'aggiornamento dei modelli di simulazione per riflettere accuratamente lo stato attuale del sistema fisico.

#### *Virtual-to-Physical (V2P)*

Il concetto di V2P si riferisce all'invio di comandi e azioni dal gemello digitale al sistema fisico. Questo flusso inverso permette al DT di influenzare il comportamento del sistema fisico basandosi sulle analisi, simulazioni e previsioni effettuate nel dominio virtuale.

Utilizzando i dati aggiornati, il DT esegue simulazioni per prevedere futuri stati del sistema e per valutare l'efficacia di diverse strategie di intervento. Le analisi possono riguardare la manutenzione predittiva, l'ottimizzazione delle operazioni e la risposta a condizioni anomale.

Sulla base delle simulazioni e delle analisi, il DT genera comandi specifici che possono includere aggiustamenti operativi, avvisi di manutenzione o altre azioni correttive. Questi comandi sono progettati per ottimizzare le performance e prevenire guasti.

I comandi generati vengono trasmessi al sistema fisico, dove vengono eseguiti tramite attuatori, sistemi di controllo automatico o interventi manuali. L'esecuzione tempestiva e precisa dei comandi è essenziale per garantire che il

sistema fisico risponda in modo adeguato alle istruzioni del DT e per ottenere di conseguenza un miglioramento immediato dello stato generale dell'entità fisica.

---

L'interazione bidirezionale tra P2V e V2P è ciò che rende un DT una rappresentazione dinamica e interattiva del sistema fisico. Questo scambio continuo di dati e comandi permette al DT di essere sempre aggiornato e di reagire rapidamente ai cambiamenti nelle condizioni operative. In particolare, grazie al flusso P2V, il DT può monitorare continuamente lo stato del sistema fisico, rilevando problemi potenziali prima che si manifestino come guasti. Oppure, utilizzando i dati storici e in tempo reale, il DT può prevedere quando una componente è a rischio di fallimento e suggerire interventi preventivi, riducendo i tempi di inattività e i costi di riparazione.

In caso di condizioni operative anomale, il DT può analizzare rapidamente la situazione e inviare comandi correttivi, minimizzando i danni e mantenendo la continuità operativa [10].

In sintesi, i concetti di P2V e V2P sono fondamentali per la funzionalità e l'efficacia di un Digital Twin. L'interazione continua e bidirezionale tra il fisico e il virtuale consente di monitorare, analizzare e ottimizzare il sistema fisico in modo proattivo e reattivo, realizzando i benefici completi del gemello digitale.



## 2.2 La nanoindentazione

La nanoindentazione è un test non convenzionale utilizzato per ricavare proprietà meccaniche del materiale. La procedura consiste nell'affondare una punta, l'indentatore, in un provino, andando a registrare continuamente forza applicata e profondità di affondamento, a partire dal punto di primo contatto.

Da queste si possono ricavare le proprietà meccaniche principali del materiale di cui è costituito il provino. Le due proprietà misurate più frequentemente, utilizzando le tecniche di indentazione, sono il modulo elastico,  $E$ , e la durezza,  $H$ . In un metodo comunemente utilizzato, i dati vengono ottenuti da un ciclo completo di carico e scarico.

### 2.2.1 Cenni storici

Le prime ricerche relative alla nanoindentazione risalgono agli esperimenti di Tabor negli anni '50 [11], che studiò l'indentazione di vari metalli usando indentatori sferici induriti e, successivamente, con indentatori conici. Questi studi rivelarono che le impronte di durezza mantengono forme simili agli indentatori utilizzati, consentendo l'interpretazione dei dati di scarico elastico per determinare le proprietà meccaniche. Questi progressi furono possibili anche grazie al progressivo sviluppo di strumenti in grado di misurare continuamente forza e spostamento durante un'indentazione.

Negli anni '70, Bulychev [12] introdusse l'uso di macchine di microdurezza strumentate per ottenere dati di carico-spostamento da indentazione, stabilendo equazioni fondamentali per l'analisi dei moduli elastici. Queste innovazioni posero le basi per la comprensione della relazione tra il comportamento di carico-scarico e le proprietà elastiche dei materiali. Gli strumenti di indentazione divennero progressivamente più sofisticati, consentendo misurazioni sempre più precise e la possibilità di analizzare materiali con strutture su scala nanometrica.

Negli anni '80, l'importanza della nanoindentazione crebbe con la necessità di misurare le proprietà meccaniche di film sottili e strati superficiali, particolarmente rilevanti nell'industria dei semiconduttori e dei rivestimenti. La tecnologia avanzò rapidamente, portando allo sviluppo di strumenti per produzioni di indentazioni su scala nanometrica. In questo periodo, Oliver, Hutchings e Pethica [13] proposero un metodo basato sulla curva di carico-spostamento e sulla funzione di forma del penetratore per stimare l'area di

contatto senza bisogno di osservazioni dirette delle impronte di durezza, rappresentando un significativo avanzamento nella precisione e nella praticità delle misurazioni.

L'articolo di Oliver e Pharr del 1992 [14,15] rappresenta una pietra miliare in questo campo, offrendo un metodo migliorato per determinare la durezza e il modulo elastico dei materiali attraverso esperimenti di indentazione con sensori di carico e spostamento. Questo metodo, che consente di analizzare dettagliatamente le curve di carico-spostamento, ha permesso di ottenere dati più affidabili e precisi, facilitando la caratterizzazione delle proprietà meccaniche di materiali avanzati, inclusi quelli utilizzati in microelettronica e nanotecnologia.

Per studiare la nanoindentazione è stato necessario appoggiarsi ai precedenti studi sul contatto elastico di fine XIX secolo di Boussinesq e Hertz. Boussinesq [16] sviluppò un metodo basato sulla teoria del potenziale per calcolare gli sforzi e gli spostamenti in un corpo elastico sottoposto alla pressione di un indentatore rigido e assialsimmetrico. Questo metodo venne poi applicato per ottenere soluzioni per diverse geometrie importanti, come quelle di indentatori cilindrici e conici.

Hertz [17] analizzò il problema del contatto elastico tra due superfici sferiche, ciascuna caratterizzata da propri raggi e costanti elastiche. Le sue soluzioni classiche costituiscono la base per gran parte del lavoro sperimentale e teorico nel campo della meccanica del contatto, fornendo un quadro di riferimento che permette di includere nell'analisi anche gli effetti degli indentatori non rigidi.

Un ulteriore contributo significativo fu apportato da Sneddon [18], che derivò relazioni generali tra il carico, lo spostamento e l'area di contatto per qualsiasi punzone descritto come un solido di rivoluzione di una funzione regolare. I risultati di Sneddon hanno permesso di estendere le analisi del contatto elastico a una più vasta gamma di forme e condizioni.

Questi contributi pionieristici costituiscono le fondamenta storiche e teoriche su cui si basa la moderna tecnica della nanoindentazione.

### 2.2.2 Procedura

La nanoindentazione è una tecnica di misura che consente ai ricercatori di misurare proprietà meccaniche come modulo elastico e durezza di materiali in diverse forme, dimensioni e scale. In particolare, questa tecnica non necessita di

alcuna preparazione del campione e può misurare proprietà per vari materiali in pochi secondi, rendendola la tecnica più rapida per tali misurazioni. Si tratta di uno sviluppo significativo rispetto ai normali metodi di prova di trazione e taglio uniassiali che impiegano giorni dalla preparazione dei campioni ai risultati finali.

La nanoindentazione, la cui macchina di misura è schematizzata in Figura 2.2, è utilizzata nelle università e nelle industrie per caratterizzare film sottili in prodotti elettronici e di imballaggio, leghe avanzate per utensili da taglio, rivestimenti per barriere termiche, proprietà viscoelastiche di polimeri, microdurezza nella qualità e nel controllo industriale, resistenza ai graffi e all'usura e molto altro.

Le misurazioni di base, necessarie per eseguire la nanoindentazione, sono carico e profondità durante gli esperimenti.

La tecnica prevede un processo di attuazione per applicare un carico, rilevare gli spostamenti e quindi applicare le correzioni necessarie. L'output sono i dati grezzi che vengono calibrati per la rigidità del telaio, i test vengono eseguiti sui siti specificati con distribuzione spaziale micrometrica e le proprietà vengono successivamente calcolate. La nanoindentazione fornisce anche misure dipendenti dalla velocità, dalla temperatura e dalla plasticità, come la tenacità alla frattura. Ciò rende la nanoindentazione lo strumento più versatile disponibile per la caratterizzazione meccanica dei materiali.

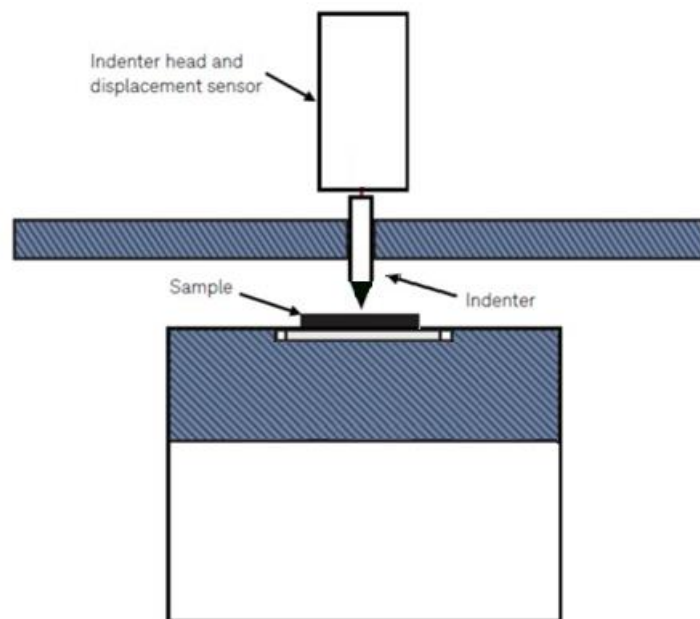


Figura 2.2 Schema macchina di indentazione

Ogni processo di indentazione è composto da tre fasi principali: carico, mantenimento e scarico [19]. Queste fasi sono essenziali per determinare le proprietà meccaniche dei materiali attraverso il test di indentazione, che fornisce informazioni su durezza, modulo elastico e comportamento viscoelastico del materiale.

La fase di carico inizia nel momento in cui la punta dell'indentatore, generalmente realizzata in diamante per la sua durezza e resistenza all'usura, entra in contatto con la superficie del provino. Durante questa fase, il carico applicato sulla punta dell'indentatore aumenta gradualmente. Questo aumento continuo del carico provoca l'affondamento progressivo dell'indentatore nel materiale del provino. La fase di carico termina una volta raggiunto il carico massimo predefinito.

La relazione, tra il carico applicato  $F$  e l'affondamento  $h$  della punta dell'indentatore durante le fasi di carico e scarico, può essere descritta da una legge esponenziale:

$$F = \alpha \cdot (h - h_0)^m$$

dove:

- $F$  è il carico applicato,
- $h$  è l'affondamento della punta,
- $h_0$  è l'affondamento iniziale, nel punto di primo contatto,
- $\alpha$  e  $m$  sono costanti specifiche del materiale e del tipo di prova, che dipendono dalle proprietà meccaniche e fisiche del materiale testato.

Una volta raggiunto il carico massimo  $F_{max}$ , inizia la fase di mantenimento. Durante questa fase, il carico massimo viene mantenuto costante per un periodo di tempo predefinito dall'operatore. L'obiettivo principale della fase di mantenimento è minimizzare gli effetti viscoelastici e di plasticità dipendenti dal tempo, come il creep e il rilassamento dello stress, che possono influenzare i risultati del test.

Infatti, durante la fase di mantenimento, materiali viscoelastici possono mostrare un comportamento dipendente dal tempo, come il creep, dove l'affondamento continua ad aumentare sotto un carico costante. Analogamente, i materiali plastici possono subire un rilassamento dello stress. La durata della fase di mantenimento è scelta per minimizzare questi effetti, garantendo che le misurazioni riflettano principalmente le proprietà elastiche del materiale.

Durante il mantenimento, la relazione tra il carico  $F$  e l'affondamento  $h$  è rappresentata da una linea retta:

$$F = F_{max}$$

In questa fase, non si verifica un ulteriore aumento del carico, ma è possibile osservare un leggero aumento dell'affondamento dovuto ai fenomeni di creep del materiale.

La fase di scarico inizia alla fine del mantenimento e continua fino al completo scarico del campione. Durante questa fase, il carico diminuisce gradualmente a zero, permettendo alla punta dell'indentatore di risalire. Anche in questa fase, la relazione tra il carico  $F$  e l'affondamento  $h$  può essere descritta dalla legge di potenza:

$$F = \alpha \cdot (h - h_f)^m$$

Dove  $h_f$  rappresenta l'affondamento residuo del materiale dopo la rimozione completa del carico.

Durante tutto il processo di indentazione, la macchina misura continuamente sia la profondità della punta sia il carico applicato. Queste misure sono fondamentali per calcolare vari parametri meccanici del materiale, come la durezza e il modulo elastico.

### *Calcolo della Rigidezza*

La rigidità del contatto  $S_m$  è definita come la derivata della curva di scarico nel punto con affondamento e carico massimi, come evidenziato in Figura 2.3.

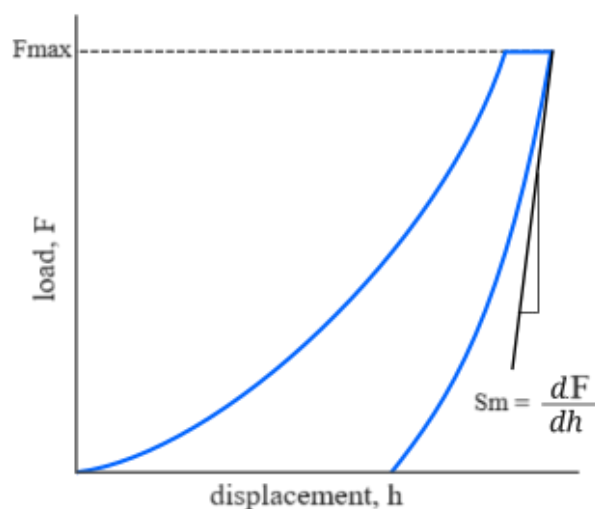


Figura 2.3 Grafico  $F-h$  di un'indentazione

Il concetto di cedevolezza si riferisce alla capacità di un sistema di deformarsi sotto l'azione di una forza, ed è strettamente legato alla rigidità, essendo il suo inverso. Non si limita al solo campione sottoposto al test, ma include anche la cedevolezza della macchina utilizzata per l'esecuzione della prova. In questo contesto, la cedevolezza della macchina comprende la deformazione di vari componenti, come il penetratore, la cella di carico e il supporto su cui è posizionato il provino. Durante un test di indentazione, le forze applicate non agiscono esclusivamente sul campione, ma coinvolgono anche la macchina stessa, provocando un aumento della profondità di indentazione registrata. Tale aumento tende a falsare i risultati, portando a una sottostima sia della durezza del materiale sia del suo modulo elastico. Di conseguenza, la cedevolezza totale  $C_t$  è determinata dalla somma della cedevolezza del campione  $C_s$  e della cedevolezza della macchina (nota anche come cedevolezza del telaio di carico)  $C_f$ .

Quindi è evidente che per calcolare la rigidità del provino

$$S = \frac{1}{C_s}$$

Sia necessario considerare la cedevolezza della macchina.

$$C_{tot} = \frac{1}{S_m} \quad C_s = \frac{1}{S} \quad \rightarrow \quad C_s = C_{tot} - C_f$$

Quindi:

$$\frac{1}{S} = \frac{1}{S_m} - C_f$$

#### *Calcolo della Durezza d'Indentazione*

La durezza d'indentazione  $H_{IT}$  si calcola, secondo la sua definizione, come il rapporto tra la forza massima  $F_{max}$  e la dimensione della proiezione della superficie di contatto tra provino e penetratore  $A_p$ :

$$H_{IT} = \frac{F_{max}}{A_p}$$

La superficie di contatto  $A_p$ , alla profondità di affondamento  $h_c$ , si definisce tramite la funzione d'area:

$$A_p = a_2 \cdot h_c^2 + a_1 \cdot h_c + a_0 \cdot h_c^{1/2}$$

I coefficienti  $a_2$ ,  $a_1$ ,  $a_0$  dipendono dalla geometria dell'indentatore. Gli indentatori più comuni sono quelli a punta di diamante a forma di piramide di

Berkovich. La corretta determinazione di questi coefficienti è fondamentale per calcolare con precisione la superficie di contatto  $A_p$  e, di conseguenza, la durezza  $H_{IT}$  e il modulo elastico  $E_{IT}$ .

L'affondamento massimo corretto  $h_c$  è definito come:

$$h_c = h_m - h_0 - \left[ C_f + \varepsilon \cdot \left( \frac{1}{S_m} - C_f \right) \right] \cdot F_{max}$$

Che si può anche scrivere:

$$h_c = h_m - h_0 - [C_f + \varepsilon \cdot C_s] \cdot F_{max}$$

dove:

- $h_0$  è l'affondamento iniziale,
- $h_m$  è l'affondamento massimo,
- $[C_f + \varepsilon \cdot C_s] \cdot F_{max}$  è il fattore correttivo che considera l'effetto della rigidità della macchina e del provino.

#### *Calcolo del Modulo Elastico*

Per quanto riguarda il calcolo del modulo elastico, si inizia definendo il modulo elastico ridotto  $E_R$ :

$$E_R = \frac{\sqrt{\pi}}{2 \cdot C_s \cdot \sqrt{A_p}} = \frac{S \cdot \sqrt{\pi}}{2 \cdot \sqrt{A_p}}$$

Quindi si può calcolare il modulo elastico di indentazione del provino  $E_{IT}$ :

$$E_{IT} = \frac{1 - \nu_s^2}{\frac{1}{E_R} - \frac{1 - \nu_i^2}{E_i}}$$

dove:

- $E_i$  e  $\nu_i$  sono rispettivamente il modulo elastico e il modulo di Poisson del materiale dell'indentatore (solitamente diamante).
- $\nu_s$  è il modulo di Poisson del provino.

Infine, la relazione tra la forza massima e la rigidezza può essere espressa come:

$$R = \frac{F_{max}}{S^2}$$

## 2.2.3 Errori di misura

Come detto precedentemente, il DT è un sistema in grado di interagire in tempo reale con il sistema e di monitorarlo e correggere eventuali errori nelle procedure.

Per iniziare la modellazione del DT per la nanoindentazione è stato quindi indispensabile isolare quelli che sono in principali errori di misura nel test di nanoindentazione.

### 2.2.3.1 Mantenimento troppo breve

Il mantenimento è la fase inserita tra carico e scarico, dove la forza viene mantenuta costante nel tempo con un valore pari a quello massimo.

È necessario per contrastare i comportamenti viscoelastici e le plasticità dipendenti dal tempo, come il creep e il rilassamento.

Quando il mantenimento non è sufficientemente lungo, o addirittura assente, l'affondamento della fase di scarico non è monotono decrescente, ma presenta una prima parte in cui i valori della profondità continuano a crescere al diminuire del carico.

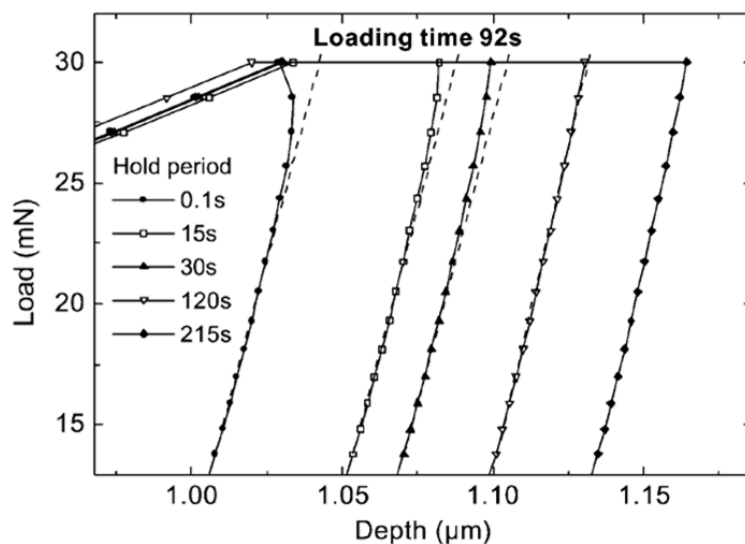


Figura 2.4 Effetto del tempo di mantenimento sulle curve di scarico

In Figura 2.4, risulta evidente questo effetto. La figura rappresenta un ingrandimento del grafico del ciclo relativo alla fase di transizione tra carico e scarico. La prima curva a sinistra, che ha come tempo di mantenimento 0.1 s, presenta la tipica forma a “naso” dovuta appunto al transitorio viscoelastico del



materiale. Si nota come, all'aumentare del tempo di mantenimento, questo effetto diminuisca sempre più fino a sparire del tutto.

È molto importante controllare che non si verifichi questo fenomeno, perché la presenza di un “naso” va ad inficiare il calcolo della rigidità di contatto, che è la derivata della forza rispetto all'affondamento nel punto iniziale della fase di scarico, inficiando, conseguentemente, il calcolo delle proprietà meccaniche del materiale.

#### *2.2.3.2 Deriva termica*

Uno dei fattori da tenere sotto controllo, che più influisce sulla correttezza della procedura di misura è lo stress termico [20]. La presenza, infatti, di una differenza di temperatura tra campione e penetratore, porta alla formazione di tensioni all'interno del provino che vanno a modificare il comportamento del materiale e la sua reazione al carico. Per contrastare queste tensioni, si inserisce nel ciclo di carico e scarico la fase di mantenimento dopo il carico, che insieme agli effetti plastici e viscoelastici, aiuta a scaricare il provino dalle tensioni dovute dalla deriva termica. Inoltre, è aggiunta una seconda fase di mantenimento durante lo scarico una volta raggiunto il 10% del carico massimo. Questa fase ha come obiettivo il contrasto degli stress termici residui all'interno del materiale.

In Figura 2.5, dove è rappresentata la variazione dell'affondamento in funzione del tempo, sono evidenti le due fasi di mantenimento. Soprattutto nella fase di mantenimento secondario, è evidente che c'è una riduzione dell'affondamento nonostante il carico sia sempre costante. Se la deriva termica non fosse presente questo tratto sarebbe perfettamente orizzontale.

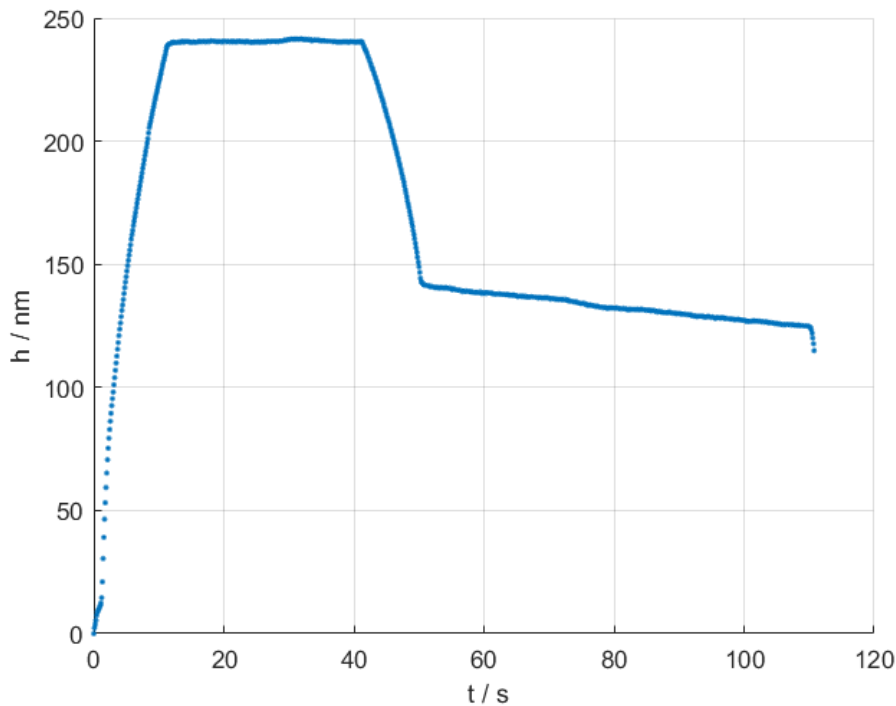


Figura 2.5 Affondamento in funzione del tempo, aggiunta del mantenimento secondario.

L'analisi del coefficiente angolare del tratto di mantenimento secondario permette, quindi, di individuare la presenza o meno della deriva termica. Inoltre, se il drift è diverso da zero, analizzandone il segno, è possibile capire la direzione del flusso termico. Se il coefficiente angolare è positivo il flusso va dal campione al penetratore, se è negativo va nella direzione opposta, dal penetratore al campione. Quest'analisi risulta indispensabile perché la presenza della deriva termica è potenzialmente pericolosa e potrebbe non essere sufficiente la presenza del mantenimento secondario per contrastarne gli effetti.

### 2.2.3.3 Vibrazioni

Per ottenere delle misure corrette è necessario porsi in un ambiente senza la presenza di vibrazioni significative.

Queste possono andare a invalidare completamente l'operazione di misura. Si possono creare dei picchi che vanno a modificare in modo essenziale la forma delle curve di indentazione.

Questo è evidente nel confronto tra Figura 2.7 e Figura 2.8. Queste sono due indentazioni effettuate all'interno di uno stesso ciclo, ossia sullo stesso provino e con lo stesso indentatore, da una macchina provvista di accelerometri in corrispondenza della piattaforma su cui è fissato il provino.

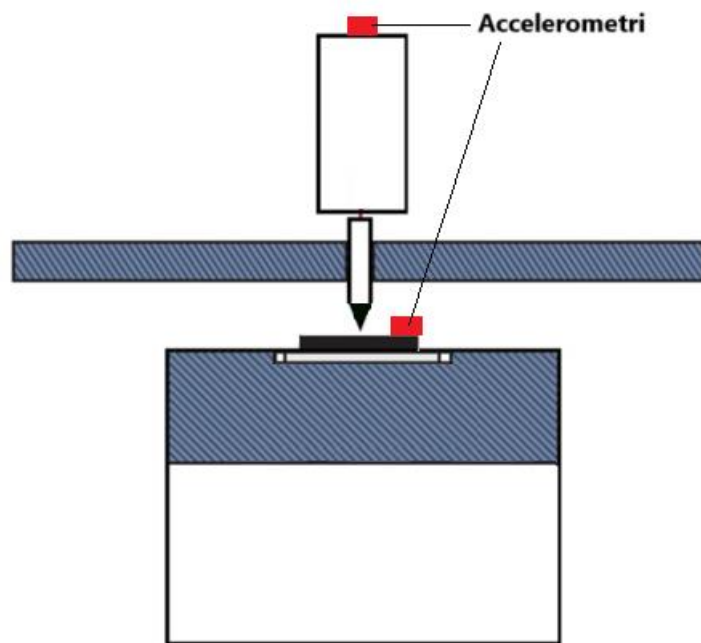


Figura 2.6 Schema della macchina con accelerometri

Come si può vedere nello schema Figura 2.6, gli accelerometri sono fissati sulla testina del penetratore e sulla piattaforma su cui è posto il provino.

Nel grafico dell'indentazione 7, in Figura 2.8, è evidente la presenza di un'anomalia che fa divergere nettamente la forma del ciclo da quello atteso, nella parte cerchiata in verde. Questo risulta ancora più chiaro dopo aver fatto il confronto con il grafico, in Figura 2.7, risultante dall'indentazione 1 dello stesso ciclo.

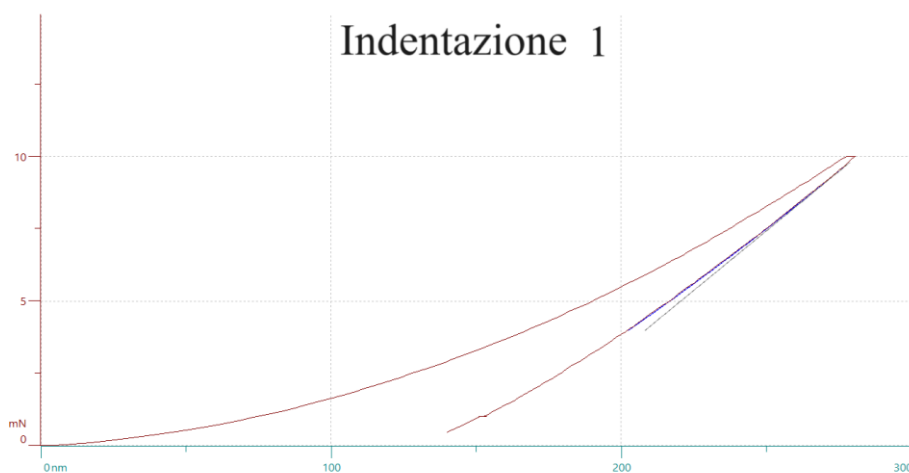


Figura 2.7 Ciclo indentazione 1

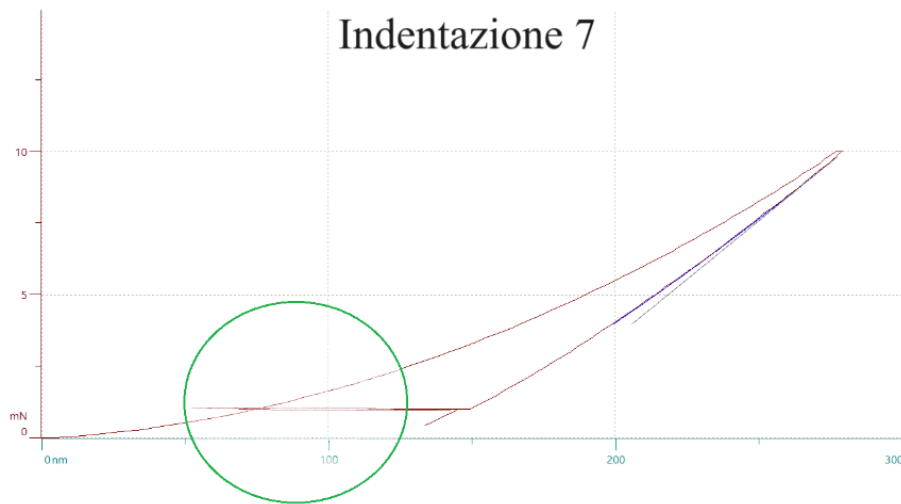


Figura 2.8 Ciclo indentazione 7

Andando ad analizzare i dati degli accelerometri relativi alle due indentazioni si estrapolano i valori per ogni istante di tempo. Andando ad analizzare i grafici ottenuti, in Figura 2.9, si nota che per l'indentazione 1 non si evidenziano accelerazioni significative, i valori rimangono infatti uniformemente distribuiti circa tra -3 g e 3 g. Per quanto riguarda l'indentazione 7 è evidente come sia presente un picco di accelerazione nella prossimità dei 1300 s e si identifica, quindi, la causa dell'instabilità riscontrata nel grafico del ciclo di indentazione.

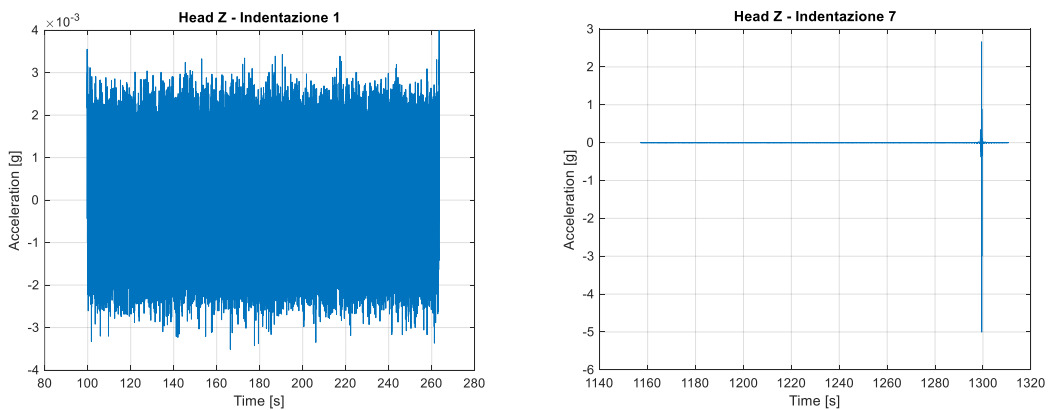


Figura 2.9 Confronto dati dell'accelerometro tra indentazione 1 e 7

Inoltre, le vibrazioni incontrollate potrebbero andare a danneggiare in modo irreparabile la punta dell'indentatore, andando a compromettere non solo la misura in corso, ma tutte le successive.

#### 2.2.3.4 Usura della punta

Un problema, che inevitabilmente si pone quando si effettuano delle misurazioni di indentazione, è quello relativo all'usura della punta.

L'indentatore usato per il processo è un indentatore Berkovich in diamante.

Questo indentatore ha forma a piramide a base triangolare. Gli angoli tra le facce della piramide sono di 65.3 gradi, l'angolo di apertura è di 142.3 gradi. L'impronta lasciata appare come in Figura 2.10. La forma piramidale permette di ottenere una penetrazione uniforme e una distribuzione omogenea dello stress sotto l'indentatore.

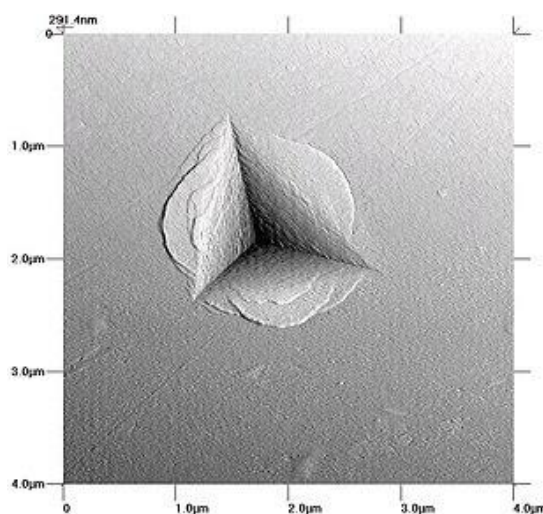


Figura 2.10 Microscopia impronta Berkovich

La profondità di penetrazione e l'area di contatto sono calcolate considerando sia la geometria dell'indentatore sia il comportamento elastico e plastico del materiale. Per cui se la punta va incontro ad un processo di usura, varierà la sua geometria; i calcoli utilizzati per individuare l'area di contatto, quindi, non saranno più adeguati e i risultati saranno conseguentemente errati.

In particolare, l'effetto osservabile sulle curve carico-affondamento è uno spostamento verso sinistra dell'intera curva. Essendosi usurato, infatti, l'indentatore raggiungerà profondità inferiori rispetto a prima.

Nel grafico in Figura 2.11 sono riportate dieci indentazioni eseguite con il medesimo indentatore sullo stesso provino. Le prime cinque, rappresentate in blu, sono state eseguite il 10 novembre e le seguenti, in verde, il 21 novembre. Tra le due sono state eseguite altre prove della stessa campagna sperimentale che hanno portato all'usura della punta. Inoltre, questi dataset sono stati creati facendo delle

misure con un indentatore conduttivo, che, attraversato dalla corrente, ha raggiunto in poco tempo un deterioramento molto evidente.

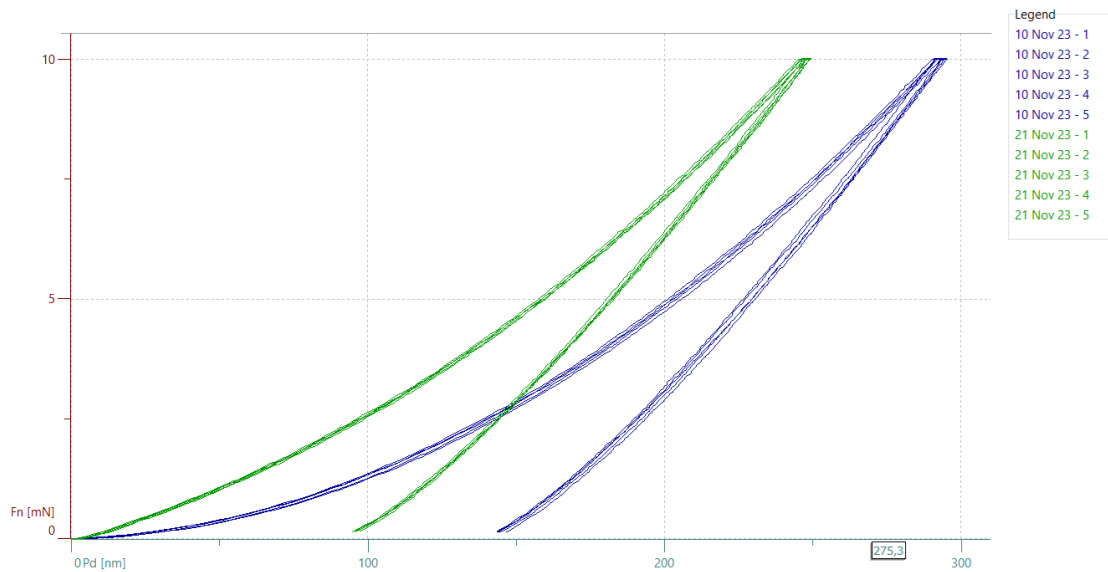


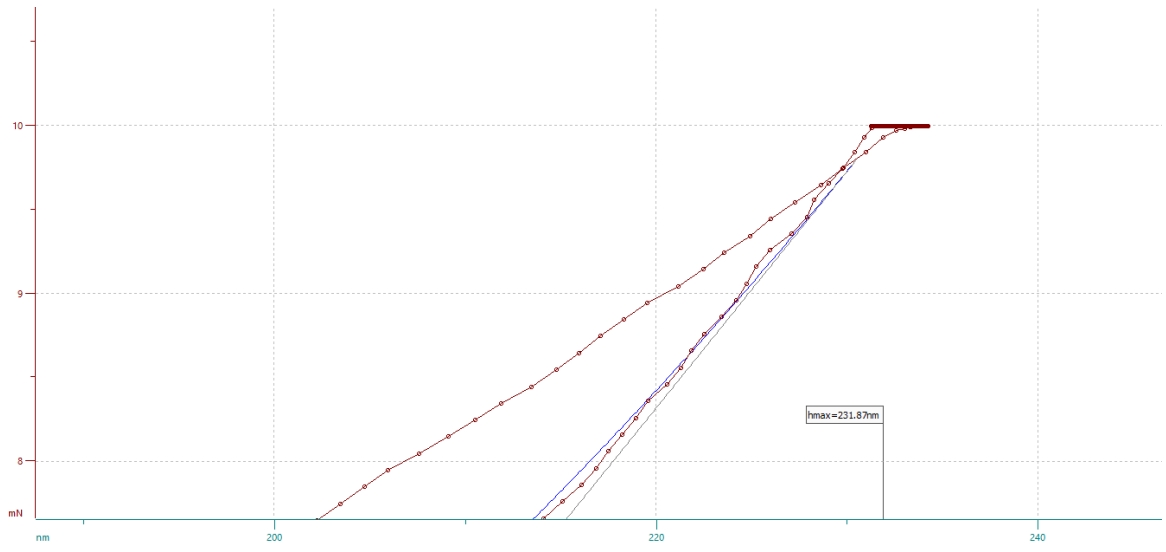
Figura 2.11 Effetto dell'usura della punta

Una volta identificato il danneggiamento, sarà necessario tarare nuovamente l'indentatore, in modo da ottenere una nuova funzione d'area e tornare a risultati accurati, oppure sostituire direttamente la punta qualora questa abbia totalmente perso la sua forma piramidale avvicinandosi sempre più ad una forma sferica.

### 2.2.3.5 Cattivo montaggio

L'ultima causa di errori di misura riscontrata durante le campagne sperimentali è certamente l'errore umano nel montaggio dell'indentatore. Se la punta non è posizionata correttamente, si possono verificare comportamenti anomali delle curve, che non rendono possibile effettuare le operazioni di misura e calcolo nel modo corretto.

Una causa immediata del cattivo montaggio della punta è l'inversione dell'andamento della fase di mantenimento, nella quale apparentemente l'affondamento torna "indietro" e diminuisce improvvisamente, invece che proseguire il suo normale andamento. Come si vede in Figura 2.12, questo effetto è molto evidente e facilmente controllabile all'interno del DT.



*Figura 2.12 Effetto del cattivo montaggio dell'indentatore*

## 3. Metodologia

Lo scopo del DT è quello di evidenziare la presenza di un discostamento sistematico dalla normalità.

Il digital twin per l'indentazione presentato in questo lavoro è stato realizzato utilizzando come linguaggio di programmazione Python, nella versione 3.11.9.

Innanzitutto, il codice realizzato, riportato in Appendice, analizza un set di curve di training per costruire un modello, ovvero per stabilire il P2V twinning e costruire la virtual entity (entità virtuale).

L'algoritmo divide i punti sperimentali in carico, mantenimento primario, scarico e mantenimento secondario. Viene fatta la regressione delle fasi di carico e scarico e calcolata l'incertezza estesa sulla variabile indipendente (Forza).

In seguito, sono calcolati, i valori di  $S_m$ ,  $S$ ,  $E_{it}$ ,  $H_{it}$  e  $R$  con le relative varianze. Viene calcolato il coefficiente angolare dei tratti di mantenimento (nella funzione  $h(t)$ ) con una regressione ai minimi quadrati. Vengono calcolate le riproducibilità di  $S_m$ ,  $S$ ,  $E_{it}$ ,  $H_{it}$ ,  $R$  e del coefficiente angolare.

Dopo aver esaurito l'analisi dei dati di training e, quindi, aver costruito una prima versione del gemello digitale, si analizza una nuova curva, ipotizzando una situazione in tempo reale, per verificare che non abbia derive significative rispetto ai dati di training.

Come per i dati su cui è costruito il modello, si dividono i punti sperimentali nelle varie fasi e viene eseguita regressione su carico e scarico, calcolando l'incertezza estesa sulla forza.

Si calcolano  $S_m$ ,  $S$ ,  $E_{it}$ ,  $H_{it}$ ,  $R$  e coefficiente angolare del mantenimento secondario con le relative varianze (a cui viene aggiunto il contributo di riproducibilità calcolato precedentemente).



Viene fatto un test d'ipotesi sul mantenimento primario per verificare che l'affondamento all'inizio del mantenimento sia minore dell'affondamento alla fine del mantenimento.

Viene eseguito un test d'ipotesi sul coefficiente angolare del mantenimento secondario per verificare che sia nullo e sono implementati anche i test d'ipotesi su tutti i parametri calcolati rispetto ai valori del training.

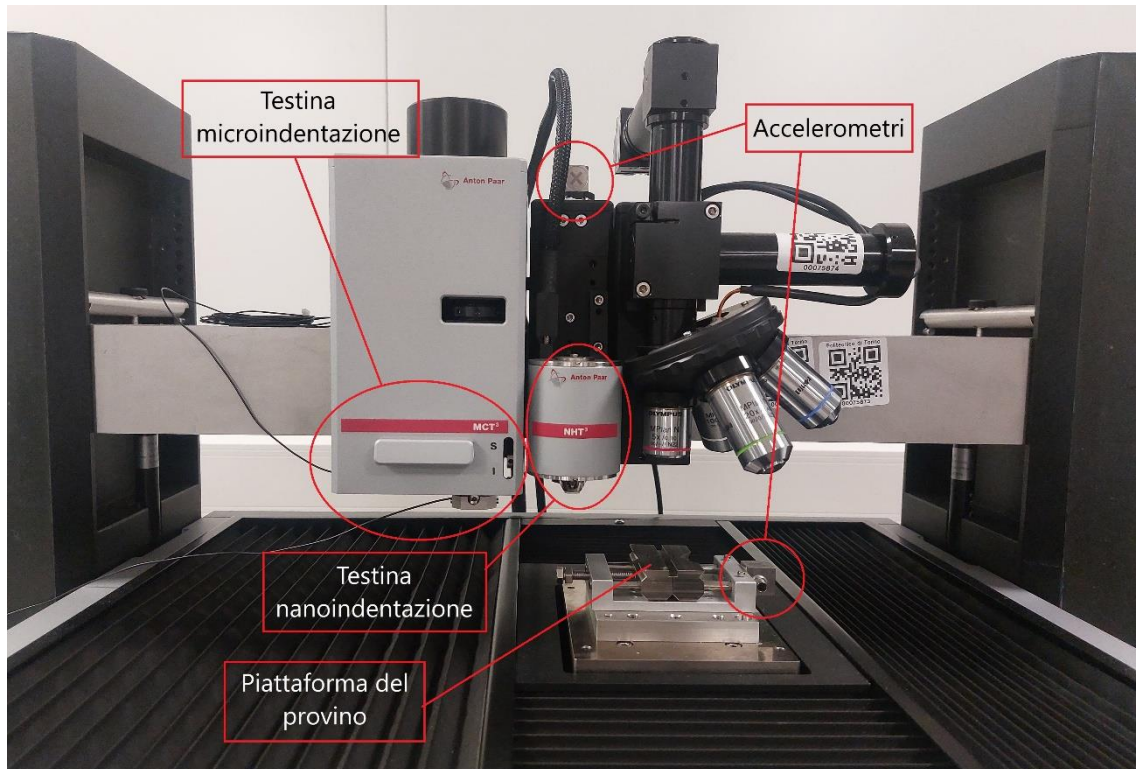
Quindi si può dire che, fino a questo punto, la costruzione del DT sia divisa in due macrofasi, la costruzione del modello, con una scelta accurata dei dati per crearlo e la prima fase di messa in atto del DT, dove viene analizzata una curva aggiuntiva, andando a simulare una poi effettiva situazione che si verrà a creare qualora il DT venga collegato alla macchina, con l'analisi della situazione attuale e il confronto con il modello precedente.

L'ultima fase è quella del V2P twinning, ossia il controllo dell'entità fisica da parte del modello digitale. In questa fase, in base ai risultati dei test effettuati ed a carte di controllo adeguatamente implementate, il DT identifica le diverse cause di eventuali errori e va a "correggere" il processo di misura.

### 3.1 Acquisizione dati

La macchina presente nel laboratorio metrologico del Mind4Lab è prodotta dall'azienda Anton Pharr. Come si vede in Figura 3.1, la macchina monta due diverse testine, la MST3 per l'indentazione a scala micrometrica e la NHT3 per la scala nanometrica. In questo progetto è stata utilizzata solamente la NHT3, poiché ci si focalizza sul processo di nanoindentazione.

Si nota la disposizione degli accelerometri, descritti nel capitolo 2.2.3.3, sulla piattaforma del provino e sul corpo della testina di nanoindentazione.



*Figura 3.1 Macchina di nanoindentazione del laboratorio Mind4Lab*

Sono stati utilizzati diversi set di indentazioni, raccolti con diverse finalità.

Inizialmente, sono state utilizzate delle campagne sperimentali effettuate per progetti precedenti. Questi dati sono costituiti da cicli convenzionali di carico, mantenimento e scarico. Sono stati utilizzati per una modellazione preliminare del sistema, in attesa di una campagna sperimentale dedicata. Per questo motivo, alcuni grafici riportati in questo lavoro saranno relativi a questi set di indentazioni. In particolare, sono stati scelti questi particolari dataset storici poiché è stato usato un indentatore conduttivo, che presentava una forte usura dopo poche indentazioni, per cui è stato possibile esaminare come variassero i parametri del DT qualora si presentasse questo tipo specifico di deriva rispetto alla normalità.

I risultati presentati sono relativi solamente al secondo dataset.

Le prime acquisizioni, di cui sono riassunte le caratteristiche in Tabella 3.1, sono state introdotte solo per completezza espositiva.

CARATTERISTICHE DATASET PRELIMINARE	
Indentatore	BC -A07
Materiale provino	SiO <sub>2</sub>
Materiale e geometria indentatore	Diamante, indentatore Berkovich
Numero di acquisizioni	3
Numero di curve per ogni acquisizione	5
Carico massimo	10 mN
Frequenza di campionamento	10 Hz
Velocità fase di carico	20 mN/min
Durata fase di mantenimento	10 s
Velocità fase di scarico	20 mN/min

*Tabella 3.1 Caratteristiche dataset preliminare*

I dati definitivi, raccolti appositamente per questo progetto, sono caratterizzati dall'aggiunta di una fase di mantenimento secondario. Questa è stata necessaria per un miglior controllo di eventuali drift termici. In particolare, per quanto riguarda le curve da testare dal DT, sono state scelte quelle appartenenti ad un set per cui, in fase di acquisizione dati, sono stati posizionati degli accelerometri per valutare le vibrazioni dell'indentatore e della piattaforma. Si aggiunge così un fattore per una migliore comprensione di eventuali derive dalla normalità dei dati.

Di seguito, in Tabella 3.2, le caratteristiche del dataset.

CARATTERISTICHE DATASET DEFINITIVO	
Indentatore	BC -A010
Materiale e geometria indentatore	Diamante, indentatore Berkovich
Materiale provino	SiO <sub>2</sub>
Numero di acquisizioni	5
Numero di curve per acquisizione	16
Carico massimo	10 mN
Frequenza di campionamento	10 Hz
Velocità di carico	60 mN/min
Durata fase di mantenimento primario	8 s
Velocità di scarico	60 mN/min
Durata fase di mantenimento secondario	60 s

*Tabella 3.2 Caratteristiche dataset definitivo*

I dati acquisiti sono stati estratti dal programma della macchina in file di testo (.TXT) dove sono contenute tutte le informazioni relative alla singola indentazione e i dati campionati dalla macchina.

Per ogni acquisizione vengono forniti i seguenti parametri:

- Time (s): l'istante di acquisizione
- Pd (nm): l'affondamento rilevato, che viene chiamato  $h$  in questo lavoro, in riferimento alla letteratura.
- Fn (mN): la forza normale, denominata  $F$  per chiarezza, in concordanza alla nomenclatura usata finora.
- SegmentID: un numero che identifica la fase dell'indentazione in cui la macchina si trova (1 carico, 2 mantenimento primario, 3 scarico, 4 mantenimento secondario, 5 scarico finale). Non verrà utilizzato in questo lavoro in quanto si è riscontrato sperimentalmente che i dati divisi tramite questo fattore non rispecchiano esattamente le curve ottenendo sovrapposizioni nei tratti di transizione tra una fase e l'altra.
- Res(ohm): Resistenza elettrica misurata al contatto tra il penetratore ed il provino [21], non utilizzato in questo lavoro.

La macchina di misura, inoltre, già fornisce un calcolo preliminare delle caratteristiche meccaniche del materiale. Però, con il DT, si vuole implementare un modello più accurato anche per il calcolo di questi fattori, per cui non ci limita ad utilizzare i dati forniti dal software.

Nell'esportazione dei dati è molto importante verificare che si stiano utilizzando i dati grezzi e non i dati corretti in automatico dalla macchina di misura. Per fare ciò è necessario azzerare manualmente il valore della cedevolezza del telaio della macchina ( $C_f$ ), che il software pone come default al valore di 0.1.

## 3.2 Elaborazione dei dati: P2V

A questo punto si entra nella modellazione del DT, in particolare della connessione Physical-to-Virtual. Dai dati grezzi si va a costruire un modello virtuale in grado di simulare adeguatamente la realtà fisica.

Si andranno quindi a dividere i dati nelle varie fasi in modo da poterle analizzare in modo approfondito separatamente e si implementerà la regressione non lineare dei valori sperimentali.

### 3.2.1 La divisione dell'indentazione in fasi

Per la divisione dei dati in base alla fase di appartenenza, andiamo ad analizzare accuratamente la loro disposizione attesa nel grafico della forza in funzione della profondità.

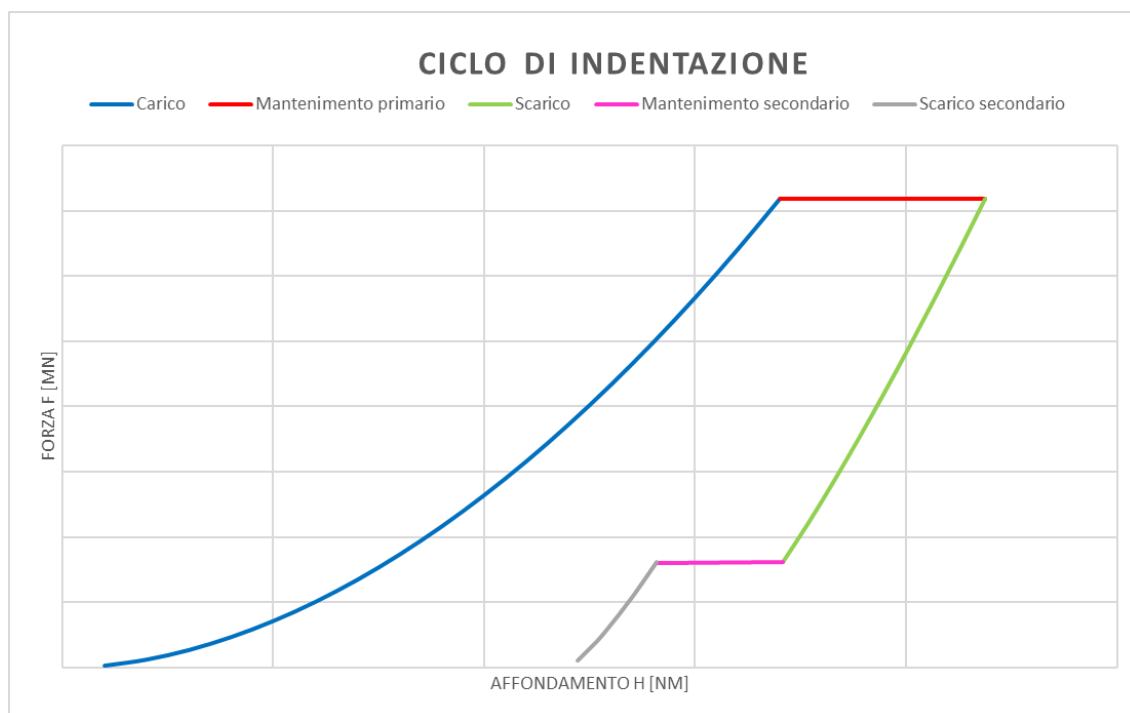


Figura 3.2 Ciclo completo con mantenimento secondario

Come si vede in Figura 3.2, il ciclo è diviso in una prima fase di carico, un mantenimento primario al carico massimo, una fase di scarico principale, un mantenimento secondario e una breve fase di scarico finale. Si rende noto che in questa immagine le fasi di mantenimento risultano molto ampliate solo a scopi grafici, nella realtà sono di lunghezza praticamente trascurabile.

Per l'elaborazione dei dati e la corretta implementazione del digital twin, l'analisi della seconda fase di scarico (in grigio) non si rivela necessaria. Quindi si vogliono individuare i dati appartenenti ai primi quattro tratti.

Per dividere i dati si utilizzano sia il tempo di campionamento che la forma attesa delle curve. In particolare, si parte isolando le fasi di mantenimento primario e secondario.

Per fare ciò si analizza la durata di ogni parte del ciclo, che viene impostata manualmente alla macchina, in modo da individuare dei tratti in cui si è certi di essere in una certa fase.

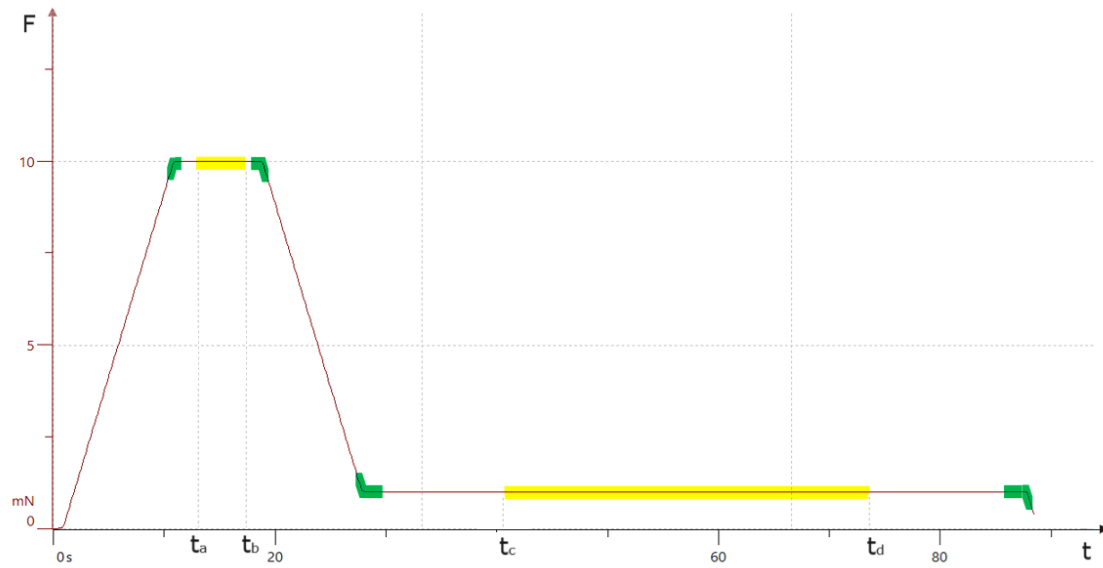


Figura 3.3 Forza in funzione del tempo in un ciclo

Andando ad analizzare il grafico in Figura 3.3, dove è rappresentata la variazione del carico in funzione del tempo, le fasi per cui si deve effettuare un controllo sono i segmenti evidenziati in verde, dove c'è un cambio di pendenza che evidenzia il passaggio dalla fase di carico e di scarico a quella di mantenimento.

Ci si va a posizionare, quindi, nei segmenti evidenziati in giallo dove si è statisticamente certi di essere nella fase di mantenimento andando a isolare gli intervalli di tempo corrispondenti.

$$t_{test1} \in [t_a, t_b] \quad t_{test2} \in [t_c, t_d]$$

Con  $t_{test1}$  e  $t_{test2}$  si selezionano i dati appartenenti agli intervalli di tempo statisticamente certi, a cui corrispondono le forze:

$$F_{test1} \in [F_a, F_b] \quad F_{test2} \in [F_c, F_d]$$

Individuati i dati di interesse, si vanno a calcolare la media e lo scarto tipo del valore della forza.

$$\bar{F}_1 = \frac{\sum_{i=a}^b F_i}{(t_b - t_a) \cdot f} \quad \bar{F}_2 = \frac{\sum_{i=c}^d F_i}{(t_d - t_c) \cdot f}$$

$$s(F_1) = \sqrt{\frac{\sum_{i=a}^b (F_i - \bar{F}_1)^2}{(t_b - t_a) \cdot f - 1}} \quad s(F_2) = \sqrt{\frac{\sum_{i=c}^d (F_i - \bar{F}_2)^2}{(t_d - t_c) \cdot f - 1}}$$

- $f$  indica la frequenza di acquisizione dei dati
- Le sommatorie sono fatte tra tutti i dati appartenenti agli intervalli  $F_{test1}$  e  $F_{test2}$ .

- $\bar{F}$  e  $s(F)$  si riferiscono rispettivamente media e scarto tipo degli intervalli di test.

I dati considerati appartenenti alla fase in esame sono quelli per cui il carico sarà all'interno dell'intervallo i cui estremi sono dati dal valore della media a cui viene addizionato e sottratto due volte lo scarto tipo.

Una volta individuate le fasi di mantenimento, si passa ad individuare le fasi di carico e scarico primario, andando ad utilizzare la sequenzialità delle fasi e la loro monotonia, crescente per la fase di carico e decrescente per la fase di scarico.

Riassumendo, i dati si dividono nel seguente modo, con un rischio di errore pari al 5%:

- Se  $(\bar{F}_1 - 2 \cdot s(F_1)) < F_n < (\bar{F}_1 + 2 \cdot s(F_1))$  allora il punto sperimentale apparterrà alla fase di mantenimento primario.
- Se  $(\bar{F}_2 - 2 \cdot s(F_2)) < F_n < (\bar{F}_2 + 2 \cdot s(F_2))$  allora il punto sperimentale apparterrà alla fase di mantenimento secondario.
- Se  $F_n > F_{n-1}$  e  $F_n < F_a$  allora il punto apparterrà alla fase di carico.
- Se  $F_n < F_{n-1}$  e  $F_n < F_c$  allora il punto apparterrà alla fase di scarico principale.
- Se il punto appartiene a nessuna delle fasi appena analizzati appartiene ai punti dello scarico finale e non sarà quindi analizzato.

### 3.2.2 La regressione dei dati per la creazione del modello

In seguito alla divisione dei dati, si procede alla regressione dei dati per ricondurre i punti sperimentali alle formulazioni di Oliver e Pharr.

Per quanto riguarda i dati di carico e scarico, si effettua una regressione non lineare con il metodo delle minime distanze ortogonali (ODR). Quindi, invece di usare una regressione "classica" dei minimi quadrati non lineare (NLS) che minimizza le distanze rispetto alla variabile indipendente, dette residui, si minimizzano le distanze normali. Questa scelta è stata fatta perché nelle fasi di carico e scarico la macchina non prosegue in controllo di forza ma in controllo di velocità di carico e scarico, per cui una regressione canonica rispetto alla forza, non terrebbe conto del fatto della variabilità di quest'ultima a livello sperimentale.

Come si può vedere in Figura 3.4, anche se applicata ad un caso lineare, è evidente la differenza tra ODR e NLS. La regressione con il metodo dei minimi quadrati minimizza le distanze verticali dei punti sperimentali, rispetto alla curva attesa,

in nero continuo, mentre la regressione con il metodo dei minimi quadrati ortogonali, minimizza le distanze normali (in rosso tratteggiato).

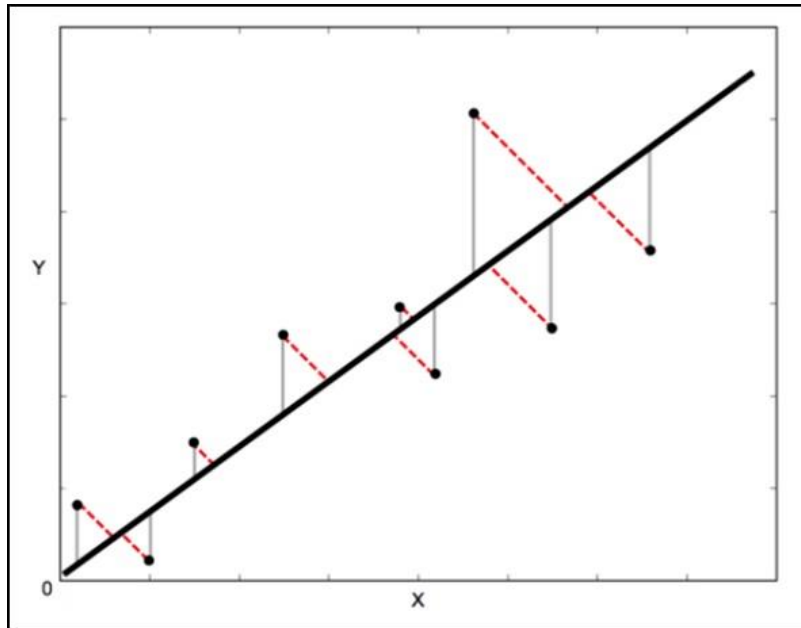


Figura 3.4 Regressione alla Distanze Ortogonali

La funzione attesa per la regressione non lineare dei dati di carico e scarico è quella ricavata da Oliver e Pharr:

$$F = \alpha \cdot (h - h_0)^m \quad F = \alpha \cdot (h - h_f)^m$$

Per quanto riguarda il mantenimento, la funzione è una retta orizzontale con intercetta pari alla media dei punti sperimentali. Per questa retta non viene fatta una regressione.

### 3.3 La caratterizzazione: il calcolo delle proprietà meccaniche

Lo scopo della nanoindentazione è la determinazione di alcune proprietà meccaniche del materiale del provino. In particolare, il modulo elastico e la durezza. È quindi fondamentale implementare il calcolo di tali proprietà all'interno del digital twin.



### 3.3.1 Il calcolo di $S_m$ e $S$

Innanzitutto, la prima grandezza da identificare è la rigidità del contatto  $S_m$ . Questa è la derivata della forza rispetto allo spostamento nel punto di inizio della fase di scarico, quindi quello con forza e spostamento massimi.

Per eseguire il calcolo, abbiamo quindi ricavato matematicamente la derivata della funzione che descrive la curva di scarico.

$$S_m = \frac{\partial F}{\partial h} = \frac{\partial}{\partial h} [\alpha \cdot (h - h_f)^m] = \alpha \cdot m \cdot (h - h_f)^{m-1}$$

Si nota come sia necessario aver calcolato precedentemente i parametri  $\alpha$ ,  $h_f$  e  $m$ .

La normativa ISO 14577-1:2015 prescrive che questi siano calcolati facendo una regressione dei dati sperimentali, scartando però i punti con carico inferiore al 20% e superiore al 98% del carico massimo. Quindi scremati i dati, è stata nuovamente eseguita la regressione ODR, preferendola nuovamente alla regressione ai minimi quadrati per le stesse considerazioni fatte precedentemente.

Per i dati di training, in realtà, non è stato necessario effettuare nuovamente la regressione. Infatti, la stima ottenuta nella prima parte del codice si basa su un gran numero di dati sperimentali, ottenendo un modello molto consistente. Una regressione che si può considerare affidabile anche in quelle zone che la normativa decide di non considerare nei calcoli, poiché sono le parti in cui è più probabile commettere errori di misura.

Definito  $S_m$ , è possibile determinare la rigidezza del provino  $S$ .

Per fare ciò, bisogna individuare la cedevolezza della macchina di misura, individuabile o tramite la calibrazione della macchina, per un risultato più accurato, o semplicemente utilizzando il valore restituito dal software dello strumento.

### 3.3.2 Calcolo della durezza e del modulo elastico d'indentazione

Si passa ora al calcolo della durezza d'indentazione  $H_{IT}$  e del modulo elastico di indentazione  $E_{IT}$ . Per fare ciò è necessario calcolare la funzione d'area dell'indentatore, necessaria al calcolo della superficie di contatto sotto carico.

Come è stato detto precedentemente, la superficie di contatto  $A_p$ , alla profondità di affondamento  $h_c$ , si definisce tramite la funzione d'area:

$$A_p = a_2 \cdot h_c^2 + a_1 \cdot h_c + a_0 \cdot h_c^{1/2}$$

I coefficienti  $a_2$ ,  $a_1$ ,  $a_0$  dipendono dalla geometria dell'indentatore. Per calcolarli, si utilizzano i dati ricavati durante la procedura di taratura dell'indentatore. Si hanno una serie di valori di  $A_p$  e i corrispondenti valori di  $h_c$ , da questi è possibile, tramite una regressione ottenere i valori di  $a_2$ ,  $a_1$ ,  $a_0$ .

Per questo calcolo è stata effettuata una regressione ai minimi quadrati "classica" perché si tratta di un semplice problema di dipendenza dell'area rispetto all'affondamento e la procedura di taratura è effettuata controllando quest'ultimo.

In seguito, si passa al calcolo dell'affondamento corretto  $h_c$  del ciclo, in modo da definire completamente  $A_p$ .

Il valore di  $h_0$  è ottenuto dall'ODR della fase di carico,  $h_m$  è l'affondamento massimo e  $\varepsilon$  è determinato in base a quanto prescritto dalla normativa. Viene utilizzata la Tabella 3.3, individuando per l'indentatore Berkovich un valore pari a 0,75.

<b>Indenter geometry</b>	$\varepsilon$
Flat punch	1
Conical	$2(\pi - 2)/\pi = 0,73$
Paraboloid of revolution (includes spherical)	3/4
Berkovich, Vickers	3/4

*Tabella 3.3 Variazione dei valori di  $\varepsilon$  in base alla geometria dell'indentatore*

È quindi possibile calcolare la durezza  $H_{IT}$ , il modulo elastico ridotto  $E_R$  e il modulo elastico di indentazione del provino  $E_{IT}$ .

Infine, viene calcolata la relazione tra la forza massima e la rigidezza:

$$R = \frac{F_{max}}{S^2}$$

## 3.4 La definizione dell'incertezza di misura

Il progetto ViDiT, all'interno del quale questo lavoro è stato svolto, vuole creare un DT metrologico specifico per la nanoindentazione, la cui implementazione permetterà la quantificazione dell'incertezza delle misurazioni in conformità con gli standard della Guida all'espressione dell'incertezza di misura (GUM). Per questo motivo risulta fondamentale il calcolo dell'incertezza estesa della regressione non lineare effettuata per le fasi di carico e scarico e per i valori caratteristici del materiale calcolati dal DT.

### 3.4.1 Cenni teorici

Ci sono due approcci principali per la valutazione dell'incertezza [22]:

- Metodi Statistici (Categoria A): Questo approccio si basa sull'analisi dei dati tramite strumenti statistici. Con tecniche come il calcolo della media e della varianza su misurazioni ripetute, si ottengono contributi di incertezza definiti "contributi di categoria A".
- Metodi Non Statistici (Categoria B): Questo approccio si fonda su conoscenze preesistenti, come informazioni di letteratura, norme tecniche o esperienze pratiche nel campo specifico. I contributi di incertezza associati, detti "contributi di categoria B", sono stimati senza ricorrere a metodi statistici formali.

In entrambi gli approcci, l'obiettivo iniziale è stimare la varianza (o lo scarto tipo) associata a ciascun fattore di incertezza, definita come "incertezza tipo".

I contributi di categoria A o B vengono identificati e quantificati in base al metodo di valutazione utilizzato, indipendentemente dall'oggetto della misurazione o dalle grandezze fisiche coinvolte.

È importante sottolineare che non esiste una correlazione diretta tra le categorie A e B e la natura sistematica o casuale degli errori, anche se gli errori casuali tendono spesso ad essere valutati come contributi di categoria A. In alcuni casi, lo stesso fattore di incertezza può essere valutato sia con un metodo statistico (categoria A) che con un metodo non statistico (categoria B).

Per i contributi di categoria B, le informazioni sulla variabilità sono spesso espresse come intervalli di variabilità. In queste circostanze, è necessario calcolare

la varianza equivalente, che rappresenta la distribuzione del fattore all'interno di tale intervallo.

Per facilitare questo calcolo, si considerano comunemente tre tipi di distribuzione: uniforme, arcoseno e triangolare.

- La distribuzione uniforme, detta anche rettangolare viene utilizzata quando non si conosce nulla sul comportamento del fattore all'interno del suo intervallo di variabilità. Se i limiti dell'intervallo sono  $\pm a$ , la varianza è:

$$u^2 = \frac{a^2}{3}$$

- La distribuzione arcoseno viene applicata per fattori che variano in modo sinusoidale, come temperatura e umidità in ambienti non controllati. Si immagini di fare misure in un ambiente in cui la temperatura  $T$  varia sinusoidalmente nell'intervallo  $\pm a$  in funzione del tempo  $t$ . In questo caso, la varianza è:

$$u^2 = \frac{a^2}{2}$$

- La distribuzione triangolare è utilizzata quando si presume che i valori centrali siano più probabili, anche se le informazioni sulla distribuzione sono limitate. La varianza in questo caso è:

$$u^2 = \frac{a^2}{6}$$

Nella valutazione dell'incertezza di categoria B basata su dati numerici provenienti da manuali o schede tecniche, si può ipotizzare una variabilità pari a  $\pm 2$  unità dell'ultima cifra significativa del valore indicato.

#### *3.4.1.1 Il calcolo dell'incertezza composta*

Spesso le grandezze che si vogliono misurare sono funzioni di più grandezze  $x_j$ :

$$Y = G(X_1, \dots, X_j, \dots, X_q)$$

Quindi si va ad evidenziare la relazione che c'è tra le misure, dove  $Y$  è la variabile dipendente e gli  $x_j$  sono le variabili indipendenti. Lo stesso legame è applicabile anche alle medie delle grandezze:

$$\bar{y} = G(\bar{x}_1, \dots, \bar{x}_j, \dots, \bar{x}_q)$$

Siccome solitamente le funzioni usate possono essere riscritte con uno sviluppo in serie di Taylor del primo ordine, per cui si può riformulare nel seguente modo:

$$\delta y = \left( \frac{\partial G}{\partial x_1} \right)_{\bar{x}} \cdot \delta x_1 + \dots + \left( \frac{\partial G}{\partial x_j} \right)_{\bar{x}} \cdot \delta x_j + \dots + \left( \frac{\partial G}{\partial x_q} \right)_{\bar{x}} \cdot \delta x_q$$

Di cui i fattori, detti coefficienti di sensibilità  $c_j = \left( \frac{\partial G}{\partial x_j} \right)_{\bar{x}}$  rappresentano le derivate parziali della funzione calcolate in corrispondenza dei valori medi.

Dopo n misurazioni si può scrivere:

$$\begin{cases} \delta y_1 = c_1 \cdot \delta x_{11} + \dots + c_j \cdot \delta x_{1j} + \dots + c_q \cdot \delta x_{1q} \\ \delta y_i = c_1 \cdot \delta x_{i1} + \dots + c_j \cdot \delta x_{ij} + \dots + c_q \cdot \delta x_{iq} \\ \delta y_n = c_1 \cdot \delta x_{n1} + \dots + c_j \cdot \delta x_{nj} + \dots + c_q \cdot \delta x_{nq} \end{cases}$$

Dove i è riferito all'i-esima misura.

In forma matriciale:

$$\begin{Bmatrix} \delta y_1 \\ \dots \\ \delta y_i \\ \dots \\ \delta y_n \end{Bmatrix} = \begin{bmatrix} \delta x_{11} & \dots & \delta x_{j1} & \dots & \delta x_{q1} \\ \dots & \dots & \dots & \dots & \dots \\ \delta x_{1i} & \dots & \delta x_{ji} & \dots & \delta x_{qi} \\ \dots & \dots & \dots & \dots & \dots \\ \delta x_{1n} & \dots & \delta x_{jn} & \dots & \delta x_{qn} \end{bmatrix} \begin{Bmatrix} c_1 \\ \dots \\ c_i \\ \dots \\ c_n \end{Bmatrix}$$

$$\{\delta y\} = [\delta x]\{C\}$$

Si va così a calcolare la varianza di y:

$$s^2(y) = \frac{\sum_{i=1}^n (\delta y_i)^2}{n-1} = \frac{\{\delta y_i\}^T \{\delta y_i\}}{n-1} = \{c_j\}^T \frac{[\delta x_{ij}]^T [\delta x_{ij}]}{n-1} \{c_j\}$$

Si nota:

$$\frac{[\delta x_{ij}]^T [\delta x_{ij}]}{n-1} = \frac{1}{n-1} \begin{bmatrix} \delta x_{11} & \dots & \delta x_{1i} & \dots & \delta x_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ \delta x_{1j} & \dots & \delta x_{ij} & \dots & \delta x_{nj} \\ \dots & \dots & \dots & \dots & \dots \\ \delta x_{1q} & \dots & \delta x_{iq} & \dots & \delta x_{nq} \end{bmatrix} \begin{bmatrix} \delta x_{11} & \dots & \delta x_{1j} & \dots & \delta x_{1q} \\ \dots & \dots & \dots & \dots & \dots \\ \delta x_{i1} & \dots & \delta x_{ij} & \dots & \delta x_{iq} \\ \dots & \dots & \dots & \dots & \dots \\ \delta x_{n1} & \dots & \delta x_{nj} & \dots & \delta x_{nq} \end{bmatrix} =$$

$$= \frac{1}{n-1} \begin{bmatrix} \sum_{i=1}^n \delta x_{i1}^2 & \cdot & \sum_{i=1}^n \delta x_{i1} \delta x_{ij} & \cdot & \sum_{i=1}^n \delta x_{i1} \delta x_{iq} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \sum_{i=1}^n \delta x_{i1} \delta x_{ij} & \cdot & \sum_{i=1}^n \delta x_{ij}^2 & \cdot & \sum_{i=1}^n \delta x_{ij} \delta x_{iq} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \sum_{i=1}^n \delta x_{i1} \delta x_{iq} & \cdot & \sum_{i=1}^n \delta x_{ij} \delta x_{iq} & \cdot & \sum_{i=1}^n \delta x_{iq}^2 \end{bmatrix} = [s^2(x)]$$

La matrice varianza-covarianza  $[s^2(x)]$  è uno strumento statistico che presenta le varianze delle variabili indipendenti lungo la sua diagonale principale. Gli elementi al di fuori della diagonale sono invece le covarianze, calcolate come la somma dei prodotti misti divisa per i gradi di libertà. Una volta compreso il comportamento delle variabili indipendenti, diventa possibile stimare la varianza della variabile dipendente.

$$s^2(y) = \{c_j\}^T [s^2(x)] \{c_j\}$$

Quando si assume che le variabili indipendenti siano completamente indipendenti l'una dall'altra, si può considerare che non ci sia alcuna relazione lineare tra di esse. Questo implica che le covarianze tra le variabili possono essere considerate pari a zero, permettendo così di semplificare l'espressione:

$$s^2(y) = \{c_j\}^T [s^2(x)] \{c_j\} = \frac{1}{n-1} \begin{bmatrix} c_1 & \cdot & c_j & \cdot & c_q \end{bmatrix} \begin{bmatrix} \sum_{i=1}^n \delta x_{i1}^2 & \cdot & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \sum_{i=1}^n \delta x_{ij}^2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & \cdot & \sum_{i=1}^n \delta x_{iq}^2 \end{bmatrix} \begin{Bmatrix} c_1 \\ \cdot \\ c_j \\ \cdot \\ c_q \end{Bmatrix}$$

$$s^2(y) = c_1^2 \frac{\sum_{i=1}^n \delta x_{i1}^2}{n-1} + \dots + c_j^2 \frac{\sum_{i=1}^n \delta x_{ij}^2}{n-1} + \dots + c_q^2 \frac{\sum_{i=1}^n \delta x_{iq}^2}{n-1}$$

$$s^2(y) = \sum_{j=1}^q \left( \frac{\partial G}{\partial x_j} \right)^2 s^2(x_j) = \sum_{j=1}^q c_j^2 s^2(x_j)$$

### 3.4.1.2 L'incertezza estesa

Secondo la GUM, l'incertezza estesa è un parametro fondamentale che descrive la gamma di valori plausibili che possono essere attribuiti a una grandezza misurata, o misurando. La valutazione "ragionevole" di questa dispersione richiede una considerazione critica e un livello di fiducia che deve essere dichiarato chiaramente per garantire una quantificazione corretta dell'incertezza.

La comunicazione precisa dei risultati di una misura dipende fortemente dall'incertezza associata, che deve includere non solo il numero di gradi di libertà, ma anche un livello di fiducia specifico. Questo è essenziale per garantire che le informazioni trasmesse siano comprese e utilizzate correttamente.

Lo scarto tipo e l'incertezza tipo composta non bastano a delineare completamente l'intervallo di valori entro cui potrebbe cadere il misurando. Infatti, in una distribuzione normale, l'intervallo dato da  $\pm u$  copre solo il 68% circa dei possibili risultati, lasciando fuori più del 30%, il che dimostra i limiti di questa rappresentazione.

In ambito metrologico, l'incertezza viene spesso espressa utilizzando un livello di copertura di  $2\sigma$ , che corrisponde a un livello di fiducia del 95%. Questo metodo è più rappresentativo della dispersione effettiva dei valori possibili del misurando.

Il concetto di incertezza estesa si basa sull'utilizzo di un fattore di copertura, indicato con il simbolo  $k$ . L'incertezza estesa,  $U$ , è data da

$$U = k \cdot u$$

dove  $u$  rappresenta l'incertezza tipo. Comunemente,  $k$  è scelto uguale a 2, per un livello di fiducia del 95%. Questo metodo consente di includere un intervallo più ampio e realistico per i valori attribuibili al misurando.

La GUM specifica che, in molti casi pratici, si può assumere che la distribuzione della variabile dipendente segua una normale, soprattutto quando le stime utilizzate per derivare il misurando provengono da distribuzioni ben definite, le

incertezze tipo contribuiscono equamente all'incertezza composta, l'approssimazione lineare è adeguata e l'incertezza composta ha un numero sufficiente di gradi di libertà. In tali condizioni, il Teorema del Limite Centrale giustifica l'uso dell'incertezza composta come stima dello scarto tipo di una distribuzione normale.

Un concetto correlato è quello dell'intervallo di fiducia, che rappresenta l'intervallo centrale della distribuzione all'interno del quale si prevede che il misurando si trovi con una determinata probabilità, definita dal livello di fiducia. Questo intervallo è la manifestazione statistica dell'incertezza.

Per determinare l'incertezza estesa, si segue una serie di passaggi metodologici. Si inizia definendo il rischio d'errore  $\alpha = 1 - P$ , scegliendo tra un intervallo unilaterale o bilaterale (simmetrico o asimmetrico). In metrologia, si utilizza spesso un livello di fiducia del 95%, associato a un rischio d'errore bilaterale simmetrico. Successivamente, si determina l'intervallo di fiducia calcolando i valori massimo e minimo di  $z$  o  $t$ , che servono a definire i limiti dell'intervallo di incertezza. Il fattore di copertura  $k$  stabilito dalla GUM corrisponde al valore massimo di  $z$  calcolato in questo contesto. L'incertezza estesa  $U$  viene quindi calcolata come  $U = k \cdot u_c(y)$  e riportata nella forma  $y \pm U(y)$ .

Esistono situazioni in cui è possibile convertire un'incertezza estesa nota in un'incertezza tipo, basandosi sul livello di fiducia e sui gradi di libertà disponibili. Se un certificato di misura indica sia il livello di fiducia sia i gradi di libertà, è possibile calcolare il fattore di copertura  $k$  utilizzando la distribuzione  $t$  di Student. Se sono indicati solo i gradi di libertà, si presuppone un livello di fiducia del 95% per calcolare l'incertezza tipo. Se viene fornito solo il livello di fiducia e non i gradi di libertà, si assume che l'incertezza estesa sia stata calcolata utilizzando la distribuzione normale e si ricava il corrispondente valore di  $z$  per ottenere  $k$ .

In assenza di informazioni specifiche sul livello di fiducia e sui gradi di libertà, si adotta convenzionalmente  $k=2$ , ipotizzando un numero elevato di gradi di libertà.

Il calcolo dei gradi di libertà è cruciale e deriva dal numero di osservazioni meno i vincoli applicati durante la misura. Per i contributi di categoria B, si assume comunemente un numero elevato di gradi di libertà, spesso intorno a 100, poiché tali contributi provengono da incertezze già stabilite o da valutazioni basate su esperienze precedenti. È importante considerare come ciascun fattore influisce sui gradi di libertà dell'incertezza tipo composta, e la formula di Welch-Satterthwaite



è utile per calcolare i gradi di libertà totali dell'incertezza composta, tenendo conto della variabilità di ciascun contributo.

L'ultimo passaggio nel processo di valutazione dell'incertezza consiste nel bilanciare i contributi di incertezza, utilizzando la formula di Welch-Satterthwaite per determinare il fattore di copertura  $k$  e, di conseguenza, l'incertezza estesa  $U$ . Questo bilancio deve essere effettuato con precisione, poiché rappresenta la sintesi di tutte le incertezze associate a una misura.

Infine, è fondamentale sottolineare che la valutazione dell'incertezza richiede non solo competenze tecniche approfondite, ma anche un approccio critico e un'integrità morale, elementi che la GUM sottolinea come essenziali per una corretta pratica metrologica. L'accurata valutazione dell'incertezza non è solo un esercizio matematico, ma una responsabilità che deve essere assunta con consapevolezza, poiché le decisioni prese sulla base di queste misure possono avere conseguenze significative.

### 3.4.2 Determinazione dei contributi di incertezza delle variabili in input

Per prima cosa è necessario analizzare e determinare quale siano i contributi di incertezza da associare alle variabili che abbiamo inserito come input nel codice. Dobbiamo quindi analizzare i dati sperimentali, i fattori correttivi e i valori caratteristici del materiale. Per ogni grandezza misurata bisogna, in genere, tenere conto dei contributi dati dall'accuratezza dello strumento  $A$ , dalla riproducibilità  $R_{\text{ep}}$  legata alle condizioni di prova e dalla risoluzione di lettura  $r$ .

Nel caso dei dati sperimentali del progetto, ossia forza e affondamento di ogni indentazione, i contributi di incertezza sono stati determinati in fase di taratura. In Tabella 3.4 sono riportati tutti i contributi di incertezza, la varianza composta calcolata e lo scarto tipo percentuale.

m		Riprod (A- Norm)		Acc - B Unif		Ris - B Unif		u <sup>2</sup>		u%	
F [μN]	hm [nm]	F [μN]	hm [nm]	F [μN]	hm [nm]	F [μN]	hm [nm]	F [μN]	hm [nm]	hm	F
19998	408,531	10	2	2,00E+02	2,04E+00	0,001	0,04	1,34E+04	5,39E+00	0,57%	0,58%
15000	348,821	12	2,3	1,50E+02	1,74E+00	0,001	0,04	7,64E+03	6,30E+00	0,72%	0,58%
10000	281,572	6	1,3	1,00E+02	1,41E+00	0,001	0,04	3,37E+03	2,35E+00	0,54%	0,58%
7500	242,497	6	1,2	7,50E+01	1,21E+00	0,001	0,04	1,91E+03	1,93E+00	0,57%	0,58%
5001	194,806	5	1	5,00E+01	9,74E-01	0,001	0,04	8,59E+02	1,32E+00	0,59%	0,59%
2501	136,501	3	0,8	2,50E+01	6,83E-01	0,001	0,04	2,18E+02	7,96E-01	0,65%	0,59%

Tabella 3.4 Taratura SiO<sub>2</sub> e calcolo dell'incertezza

La riproducibilità è stata valutata come un contributo di tipo A normale, la risoluzione e l'accuratezza come contributi di tipo B con distribuzione uniforme. Segue che la varianza composta è calcolata come:

$$u^2(x) = \frac{A^2}{3} + R_{ep}^2 + \frac{r^2}{3}$$

Si è calcolato il valore dell'incertezza percentuale come rapporto tra lo scarto tipo calcolato e la misura relativa.

$$u_{h_m\%} = \frac{u_{h_m}}{h_m} \quad u_{F\%} = \frac{u_F}{F}$$

Si nota come l'incertezza percentuale sia pressoché costante, per cui si utilizzerà quest'ultima per determinare l'incertezza dei dati sperimentali.

Per cui:

$$u_h = 0.0006 \cdot h \quad u_F = 0.00058 \cdot F$$

Per il calcolo delle caratteristiche del materiale, come visto nel capitolo 3.3, abbiamo bisogno di diversi parametri in input. Questi sono la cedevolezza della macchina  $C_f$ , il modulo di Young  $E_i$  e il modulo di Poisson  $\nu_i$  del materiale di cui è composta la punta dell'indentatore, il modulo di Poisson  $\nu_s$  del provino e il fattore di correzione  $\varepsilon$ .

Per quanto riguarda la cedevolezza della macchina  $C_f$  il valore della varianza associata è dato dalla taratura della macchina.

I valori del modulo di Young e di Poisson del materiale della punta, in questo caso diamante, sono presi da valori tabellati e la varianza è stata determinata come un contributo di tipo B uniforme. Il valore della varianza è calcolato come il quadrato del doppio dell'ultima cifra significativa diviso per 3, ovvero associando una distribuzione uniforme, al contributo valutato secondo metodo PUMa.

$$u_{\nu_i}^2 = \frac{(0.01 \cdot 2)^2}{3} \quad u_{E_i}^2 = \frac{(1 \cdot 2)^2}{3} \text{ [GPa]}$$

Il valore del modulo di Poisson del provino deriva dalla taratura (nel caso di materiali di riferimento), o da valori tabellari.

Il valore di  $\varepsilon$  è tabulato nella normativa quindi si calcola, come per il modulo di Young del diamante, come un contributo di tipo B con distribuzione uniforme.

$$u_{\varepsilon}^2 = \frac{(0.01 \cdot 2)^2}{3}$$

In Tabella 3.5 sono riassunti i valori e le varianze dei parametri di input.

	misura (x)	$u^2(x)$
$C_f$ [nm/mN]	0,1	2,89E-08
$\varepsilon$ [-]	0,75	1,33E-04
$\nu_s$ [-]	0,161	1,00E-06
$\nu_i$ [-]	0,07	1,33E-04
$E_i$ [GPa]	1,14	3,00E-06

Tabella 3.5 Valori e varianze parametri dei parametri di input

### 3.4.3 L'incertezza estesa per le fasi di carico e scarico

Per quanto riguarda la modellizzazione derivata dai dati sperimentali, si vuole ottenere l'incertezza estesa della forza punto per punto. Soprattutto per quanto riguarda le fasi di carico e scarico. Per fare ciò, come è stato detto in precedenza, è stata eseguita una regressione ODR non lineare. Per eseguire questa regressione nel codice è stata utilizzata il pacchetto ODR della libreria Scipy di Python [23].

La funzione richiede in ingresso i dati sperimentali e i relativi scarti tipo, tutti sotto forma di vettori e la funzione che dovrebbe modellizzare il sistema.

Oltre a calcolare i parametri della funzione, l'*odrpac* restituisce anche la matrice varianza-covarianza relativa ai parametri. In Figura 3.5 è rappresentato uno schema esemplificativo della funzione per la fase di carico (analogamente funziona per lo scarico) con dati in input e corrispondenti output.

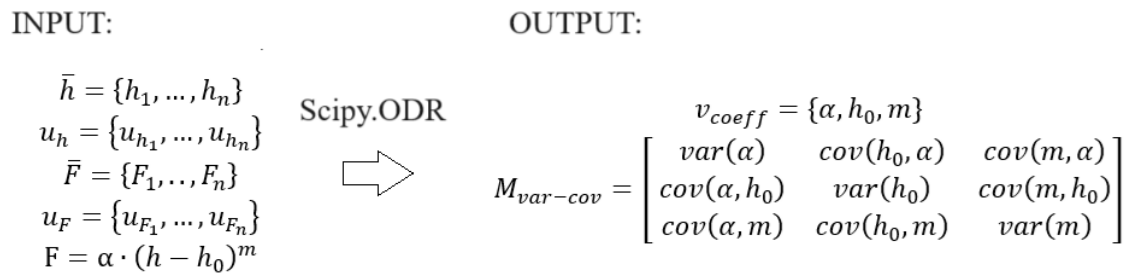


Figura 3.5 Schema di funzionamento della funzione Scipy.ODR

Per il calcolo dell'incertezza estesa dobbiamo usare la formula trovata nel capitolo 3.4.1.2:

$$U(F) = k \cdot u(F)$$

Dove il  $k$  si calcola dalla distribuzione di Student, usando come livello di fiducia il 95% e gradi di libertà pari a 100. I gradi di libertà si scelgono pari a 100 perché abbiamo un grande numero di dati sperimentali che vanno a creare un modello

molto consistente e utilizzare 100 come gradi di libertà risulta già una stima conservativa.

Per il calcolo dell'incertezza composta della forza è necessario non trascurare la presenza delle covarianze, non è possibile affermare che i dati siano tra di loro indipendenti. Si usa quindi la seguente equazione:

$$u^2(F) = \begin{pmatrix} \frac{\partial F}{\partial \alpha} & \frac{\partial F}{\partial h_f} & \frac{\partial F}{\partial m} & 1 & \frac{\partial F}{\partial h} \end{pmatrix} \begin{bmatrix} \begin{matrix} \text{matrix} \\ \text{from} \\ \text{ODR} \end{matrix} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \frac{\partial F}{\partial \alpha} \\ \frac{\partial F}{\partial h_f} \\ \frac{\partial F}{\partial m} \\ 1 \\ \frac{\partial F}{\partial h} \end{pmatrix}$$

L'equazione è scritta per la fase di scarico, per la fase di carico è la medesima sostituendo  $h_0$  a  $h_f$ .

Si nota subito la novità dell'introduzione nella formula trovata nel capitolo 3.4.1, del Mean Squared Error (MSE).

Il Mean Squared Error, o errore quadratico medio, è una misura utilizzata per quantificare la differenza tra i valori predetti da un modello e i valori effettivi (o osservati). Si calcola come la media aritmetica dei quadrati dei residui, ossia delle differenze tra i valori previsti e i valori osservati.

Formalmente, per un insieme di  $n$  coppie di valori osservati  $y_i$  e valori predetti  $\hat{y}_i$ , il MSE è definito come:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Nel caso in esame queste differenze non sono altro che la differenza tra i valori della forza campionati sperimentalmente per una certa profondità e i valori della forza "predetti" dalla regressione non lineare alla medesima profondità.

Un basso valore del MSE indica che le previsioni del modello sono, in media, vicine ai valori osservati. Un alto valore, al contrario, evidenzia che c'è una grande differenza tra le previsioni e i valori osservati.

L' MSE è ampiamente utilizzato perché penalizza maggiormente gli errori grandi (a causa dell'elevazione al quadrato), ed è quindi sensibile agli outlier. L'MSE è di fatto la varianza dei residui della regressione.

Quando si introduce il MSE nel calcolo della varianza composta, si sta in sostanza considerando quanto della varianza totale osservata nei dati può essere attribuita agli errori del modello rispetto a quanto è invece spiegato dalle variabili indipendenti considerate nel modello.

Si evidenzia che il calcolo della varianza composta e, di conseguenza, dell'incertezza estesa è effettuato punto per punto.

### 3.4.4 La varianza composta nei parametri caratteristici del materiale

Analizziamo ora il calcolo della varianza composta per i parametri caratteristici del materiale, ossia la durezza  $H_{IT}$  e il modulo elastico  $E_{IT}$ .

Prima di tutto va individuata la rigidità totale. Come abbiamo detto prima, per effettuare il calcolo si utilizzano i parametri calcolati nella regressione ODR della fase di scarico. Per cui anche per l'individuazione dell'incertezza estesa è necessario considerare la matrice varianza-covarianza restituita dalla funzione di Python.

Per il calcolo della varianza composta si usa la stessa equazione usata per il calcolo della varianza della forza nelle fasi di carico e scarico.

$$u^2(Sm) = \left\{ \begin{matrix} \frac{\partial Sm}{\partial \alpha} & \frac{\partial Sm}{\partial h_f} & \frac{\partial Sm}{\partial m} & \frac{\partial Sm}{\partial h} \end{matrix} \right\} \left[ \begin{matrix} [matrix] & 0 \\ from & 0 \\ ODR & 0 \\ 0 & 0 & 0 & u^2(h) \end{matrix} \right] \left\{ \begin{matrix} \frac{\partial Sm}{\partial \alpha} \\ \frac{\partial Sm}{\partial h_f} \\ \frac{\partial Sm}{\partial m} \\ \frac{\partial Sm}{\partial h} \end{matrix} \right\}$$

L'unica differenza sta ovviamente nel calcolo delle derivate parziali.

Una volta individuata l'incertezza di  $Sm$  si passa al calcolo dei coefficienti della funzione d'area. Questi, come analizzato in precedenza, si ottengono dalla regressione non lineare ai minimi quadrati con la funzione `Scipy.curve_fit` di Python.

Anche *curve\_fit* come ODR restituisce i coefficienti della funzione e la relativa matrice varianza-covarianza che servirà in seguito per calcolare la varianza di  $A_p$ .

Prima è però necessario definire la varianza di  $h_c$ .

Ricordiamo la definizione di  $h_c$ :

$$h_c = h_m - h_0 - [C_f + \varepsilon \cdot C_s] \cdot F_{max}$$

Con

$$C_s = \frac{1}{S_m} - C_f$$

La cui varianza è calcolata con i seguenti passaggi:

$$u(C_{tot}) = C_{tot} \cdot u(S_m)_{\%}, \quad \text{con} \quad u(S_m)_{\%} = \frac{u(S_m)}{S_m}$$

$$u^2(C_s) = u^2(C_f) + u^2(C_{tot})$$

Inoltre, si calcola in questa sede anche la varianza di  $S = 1/C_s$

$$u^2(S) = (S \cdot u(C_s)_{\%})^2, \quad \text{con} \quad u(C_s)_{\%} = \frac{u(C_s)}{C_s}$$

Tornando al calcolo della varianza di  $h_c$ , si può scrivere:

$$\begin{aligned} u^2(h_c) = & \left( \frac{\partial h_c}{\partial h_m} \right)^2 \cdot u^2(h_m) + \left( \frac{\partial h_c}{\partial h_0} \right)^2 \cdot u^2(h_0) + \left( \frac{\partial h_c}{\partial C_f} \right)^2 \cdot u^2(C_f) + \left( \frac{\partial h_c}{\partial \varepsilon} \right)^2 \cdot u^2(\varepsilon) \\ & + \left( \frac{\partial h_c}{\partial C_s} \right)^2 \cdot u^2(C_s) + \left( \frac{\partial h_c}{\partial F_{max}} \right)^2 \cdot u^2(F_{max}) \end{aligned}$$

È stato utilizzato il calcolo semplificato della varianza in quanto tutte le variabili si possono considerare indipendenti tra loro.

Si nota che la varianza di  $F_{max}$  è quella calcolata nel capitolo 3.4.3 per il valore massimo della forza.

È possibile quindi calcolare la varianza di  $A_p$ .

$$u^2(A_p) = \left\{ \frac{\partial A_p}{\partial a_2} \quad \frac{\partial A_p}{\partial a_1} \quad \frac{\partial A_p}{\partial a_0} \quad \frac{\partial A_p}{\partial h_c} \right\} \left[ \begin{array}{c} \text{matrix} \\ \text{from} \\ \text{curve\_fit} \\ 0 \quad 0 \quad 0 \end{array} \right] \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ u^2(h_c) \end{array} \right] \left\{ \begin{array}{c} \frac{\partial A_p}{\partial \alpha} \\ \frac{\partial A_p}{\partial h_f} \\ \frac{\partial A_p}{\partial m} \\ \frac{\partial A_p}{\partial h_c} \end{array} \right\}$$

Finalmente è possibile determinare la varianza composta della durezza d'indentazione  $H_{IT}$ , del modulo elastico ridotto  $E_R$  e quindi del modulo elastico di indentazione  $E_{IT}$  e di  $R$ .

$$u^2(H_{IT}) = \left( \frac{\partial H_{IT}}{\partial F_{max}} \right)^2 \cdot u^2(F_{max}) + \left( \frac{\partial H_{IT}}{\partial A_p} \right)^2 \cdot u^2(A_p)$$

$$u^2(E_R) = \left( \frac{\partial E_R}{\partial S} \right)^2 \cdot u^2(S) + \left( \frac{\partial E_R}{\partial A_p} \right)^2 \cdot u^2(A_p)$$

$$u^2(E_{IT}) = \left( \frac{\partial E_{IT}}{\partial E_R} \right)^2 \cdot u^2(E_R) + \left( \frac{\partial E_{IT}}{\partial E_i} \right)^2 \cdot u^2(E_i) + \left( \frac{\partial E_{IT}}{\partial v_s} \right)^2 \cdot u^2(v_s) + \left( \frac{\partial E_{IT}}{\partial v_i} \right)^2 \cdot u^2(v_i)$$

$$u^2(R) = \left( \frac{\partial R}{\partial F_{max}} \right)^2 \cdot u^2(F_{max}) + \left( \frac{\partial R}{\partial S} \right)^2 \cdot u^2(S)$$

## 3.5 Test statistici

Fino ad ora, quindi, sono stati analizzati i dati sperimentali, è stato creato il modello di training e ne sono stati calcolati i parametri caratteristici. È stata analizzata una curva “attuale” alla quale è stata applicata la stessa analisi.

A questo punto è necessario implementare dei test per verificare la presenza di errori nel modello confrontando i valori del training a quelli della curva n+1, esaminata in tempo reale. Per fare ciò sono stati implementati dei Test d'ipotesi.

### 3.5.1 Il Test d'ipotesi

Il test d'ipotesi è una procedura statistica fondamentale per decidere se accettare o rifiutare un'ipotesi di base (nota come ipotesi nulla,  $H_0$ ) rispetto a un'ipotesi alternativa ( $H_1$ ). Questo processo è cruciale in molti campi scientifici per determinare la validità delle inferenze fatte dai dati osservati. La costruzione di

un test d'ipotesi inizia con la formulazione delle due ipotesi: l'ipotesi nulla, che generalmente afferma l'assenza di un effetto o di una differenza significativa, e l'ipotesi alternativa, che è l'ipotesi contro cui si vuole testare  $H_0$ .

Il processo del test d'ipotesi prevede la raccolta di un campione di dati, la selezione di una statistica test (una funzione dei dati osservati), e la definizione di una regola decisionale basata sul confronto tra la statistica calcolata e una distribuzione di riferimento, scelta in base all'ipotesi nulla.

Il passo successivo è calcolare il valore osservato della statistica test utilizzando i dati del campione e confrontarlo con i valori critici della distribuzione teorica. Se il valore osservato cade in una regione di rifiuto, si decide di rigettare l'ipotesi nulla a favore dell'ipotesi alternativa. La regione di rifiuto viene definita dal livello di significatività  $\alpha$ , che rappresenta la probabilità di commettere un errore di tipo I, ossia rifiutare  $H_0$  quando in realtà è vera. Questo livello di significatività è scelto a priori e di solito assume valori come 0,05 o 0,01, corrispondenti a una confidenza rispettivamente del 95% o del 99%.

Il valore p è un concetto centrale nei test d'ipotesi. Esso rappresenta la probabilità di osservare un risultato almeno così estremo come quello effettivamente ottenuto, supponendo che  $H_0$  sia vera. Se il valore p è inferiore al livello di significatività  $\alpha$ , si rigetta  $H_0$ . Un valore p piccolo indica che i dati osservati sono molto improbabili sotto l'ipotesi nulla, suggerendo che l'ipotesi alternativa potrebbe essere più appropriata.

Esistono diverse tipologie di test d'ipotesi, ciascuna adatta a specifiche situazioni. I test a una coda e a due code sono tra i più comuni. Un test a una coda verifica se un parametro è maggiore o minore di un valore specifico, mentre un test a due code verifica se un parametro è diverso da un valore specifico, senza specificare la direzione della differenza. La scelta tra un test a una coda e a due code dipende dall'ipotesi alternativa specificata. Inoltre, esistono test parametrici e non parametrici. I test parametrici assumono che i dati seguano una certa distribuzione, come la distribuzione normale, mentre i test non parametrici non richiedono tali assunzioni. I test parametrici tendono ad avere maggiore potenza rispetto ai test non parametrici, ma possono essere meno robusti se le assunzioni sulla distribuzione dei dati non sono soddisfatte.



### 3.5.2 Controllo del mantenimento primario

Per quanto concerne il mantenimento primario è necessario verificare la presenza di un errore in fase di montaggio per cui il mantenimento abbia un andamento opposto a quello naturale.

Si formula quindi l'ipotesi nulla:

$$h_{0,mant} < h_{fin,mant}$$

Per cui verifichiamo che il primo punto del mantenimento non abbia affondamento maggiore dell'ultimo punto, in ordine di acquisizione.

Per verificare che l'ipotesi nulla non sia invalidata si utilizza un test t di Student ad una coda. Per calcolare il t di riferimento si assume un livello di fiducia del 95% e gradi di libertà pari a 100.

Per quanto riguarda la t di test riscriviamo la disequaglianza:

$$h_{fin,mant} - h_{0,mant} > 0$$

Per cui calcoliamo la t di test come:

$$t_{test} = \frac{h_{fin,mant} - h_{0,mant}}{\sqrt{u^2(f)_{fin,mant} + u^2(f)_{0,mant}}}$$

Come esemplificato in Figura 3.6, se la t di test è minore della t di riferimento allora l'ipotesi nulla viene rifiutata, per cui il mantenimento non avviene nel modo corretto.

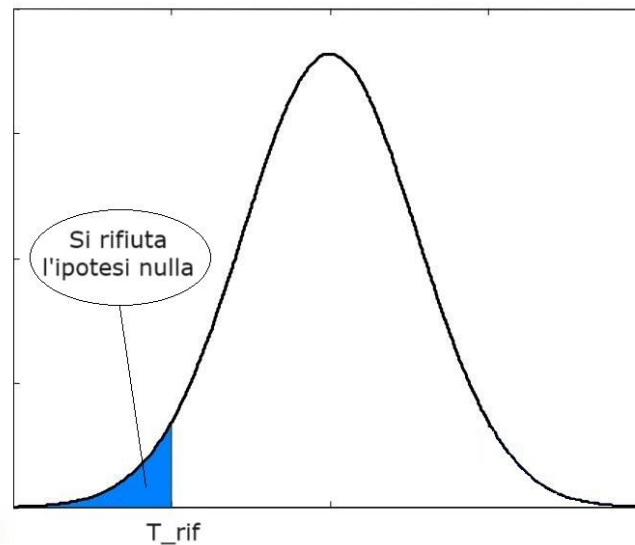


Figura 3.6 Test d'ipotesi ad una coda

### 3.5.3 Controllo del mantenimento secondario

Per verificare che non ci sia deriva termica, come scritto nel capitolo 2.2.3.2, è necessario controllare che il coefficiente angolare della funzione  $h(t)$  durante il mantenimento secondario sia nullo.

Per fare ciò è necessario calcolare il coefficiente angolare del mantenimento e la sua varianza.

Si vanno ad utilizzare nuovamente la funzione `curve_fit` per effettuare la regressione lineare dei dati del mantenimento, utilizzando come funzione attesa quella della retta:

$$h = m * t + q$$

Dove  $q$  rappresenta l'intercetta con l'asse delle ordinate e  $m$  è il coefficiente angolare, ossia il nostro obiettivo.

Come ormai noto, la funzione restituisce un vettore contenente i parametri della retta ( $m$  e  $q$ ) e la matrice varianza-covarianza associata. Per cui da questi è facilmente possibile estrarre il valore di  $m$  e la sua varianza  $u^2(m)$ .

Ora è così possibile implementare il test d'ipotesi che ha come ipotesi nulla:

$$m = 0$$

In questo caso viene utilizzato il test t di Student a due code, calcolando la t di riferimento con livello di fiducia del 95% e gradi di libertà pari al numero di dati sperimentali appartenenti alla fase di mantenimento secondario meno il numero di parametri della regressione, ossia 2.

$$DOF_{t_{rif}} = n - 2$$

La t di test è calcolata come il valore del coefficiente angolare diviso il suo scarto tipo.

$$t_{test} = \frac{m}{u(m)}$$

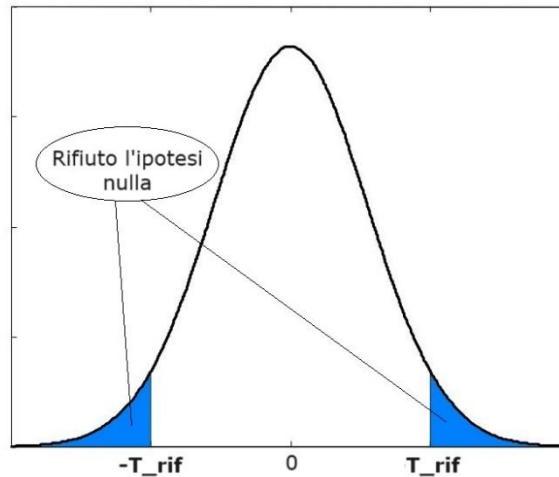


Figura 3.7 Test d'ipotesi a due code

Se, come si vede in Figura 3.7, la  $t$  di test non è compresa tra il valore della  $t$  di riferimento e il suo opposto, allora si rifiuta l'ipotesi nulla.

Qualora questo avvenga, è confermata la presenza di una deriva termica rilevante.

### 3.5.4 Test d'ipotesi sulle proprietà caratteristiche dei materiali calcolate.

Infine, si implementano dei test di ipotesi per verificare la coincidenza dei valori della curva analizzata in real-time con quelli calcolati a partire dal dataset di training.

Le ipotesi nulle sono:

$$S_{training} = S_{test}$$

$$E_{IT,training} = E_{IT,test}$$

$$H_{IT,training} = H_{IT,test}$$

$$R_{training} = R_{test}$$

Dove le grandezze di training sono quelle calcolate nell'allenamento del modello, mentre le grandezze di test sono quelle calcolate a partire dalla curva analizzata in real-time.

Per effettuare il test d'ipotesi viene utilizzato il test  $t$  di Student a due code, calcolando la  $t$  di riferimento con livello di fiducia del 95% e gradi di libertà pari a 100.

Le t di test sono calcolate nel seguente modo:

$$t_{test,S} = \frac{S_{training} - S_{test}}{\sqrt{u^2(S)_{training} + u^2(S)_{test}}}$$

$$t_{test,E_{IT}} = \frac{E_{IT,training} - E_{IT,test}}{\sqrt{u^2(E_{IT})_{training} + u^2(E_{IT})_{test}}}$$

$$t_{test,H_{IT}} = \frac{H_{IT,training} - H_{IT,test}}{\sqrt{u^2(H_{IT})_{training} + u^2(H_{IT})_{test}}}$$

$$t_{test,R} = \frac{R_{training} - R_{test}}{\sqrt{u^2(R)_{training} + u^2(R)_{test}}}$$

Se la t di test non è compresa tra il valore della t di riferimento e il suo opposto allora si rifiuta l'ipotesi nulla. In questo caso la misura in corso non può essere considerata attendibile e sicuramente saranno presenti degli errori nella procedura di misurazione.

Starà all'utente valutare se scartare o meno la misurazione in base a quali e quanti parametri tra quelli calcolati non supereranno il test d'ipotesi.

## 4. Risultati

Si analizzano di seguito i risultati ottenuti applicando il metodo utilizzato sul dataset dedicato a questo progetto.

### 4.1 Divisione dei dati sperimentali

Innanzitutto, si acquisiscono i dati dalla macchina e il codice provvede a dividerli in fasi.

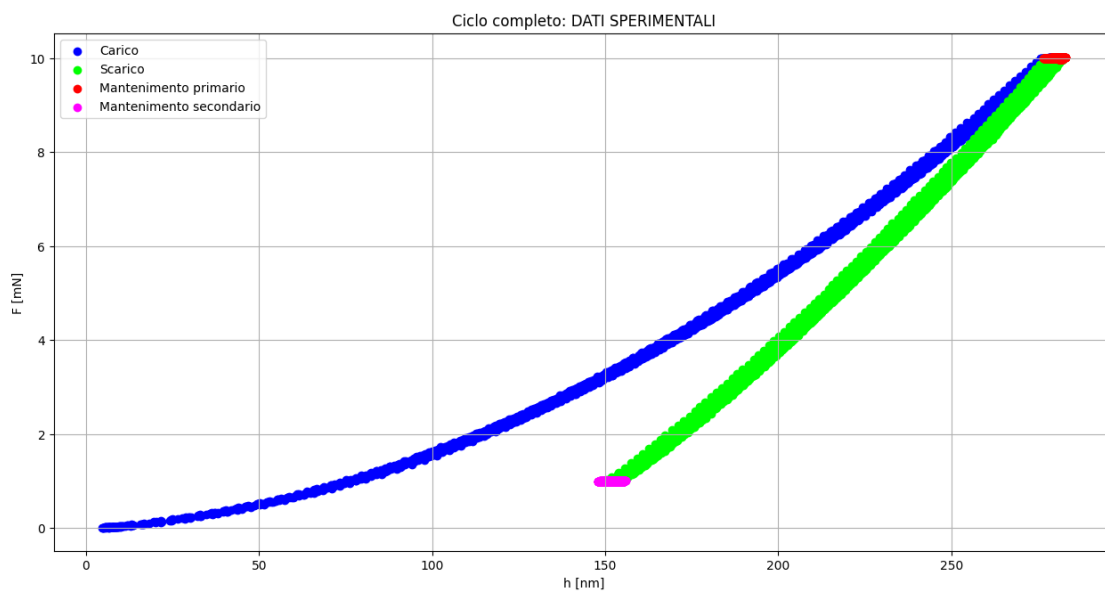


Figura 4.1 Divisione dei dati grezzi di training in fasi

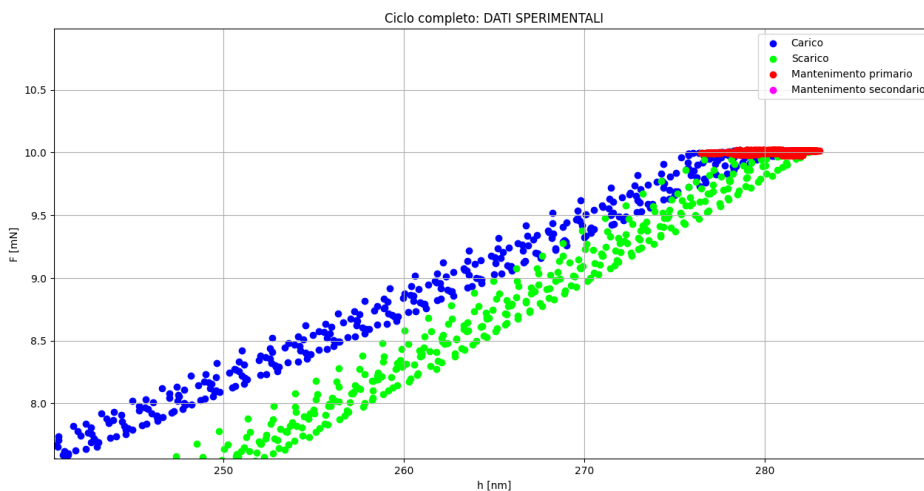


Figura 4.2 Ingrandimento divisione dei dati sperimentali, focus sul mantenimento primario

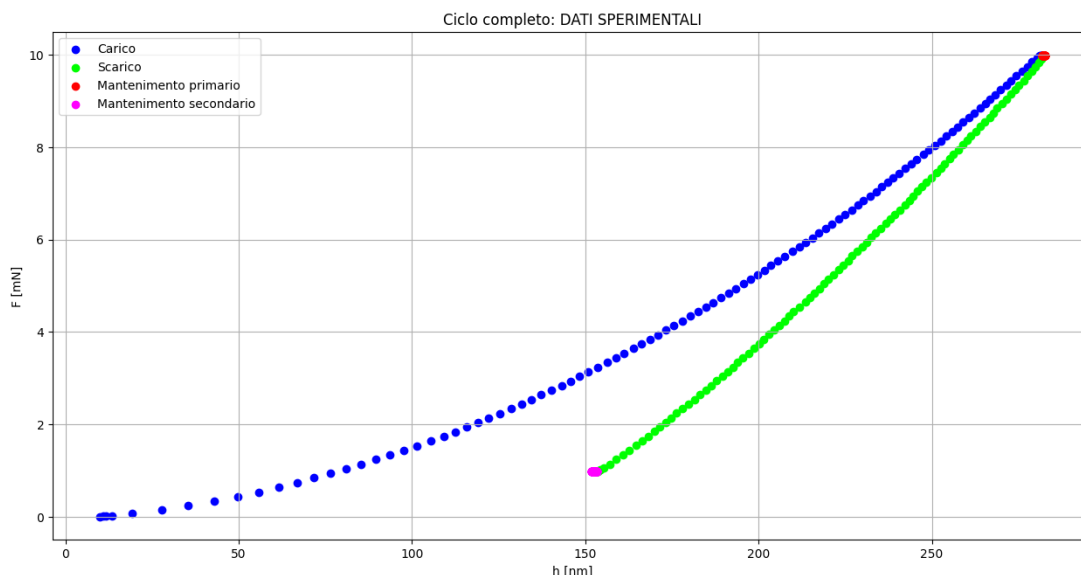


Figura 4.3 Divisione della curva real-time

In Figura 4.1 sono rappresentati i dati derivati dalle 16 curve che andranno a costituire il training del modello, mentre in Figura 4.3 è rappresentata la divisione di una singola acquisizione dati che simulerà la curva acquisita in tempo reale.

I dati sono divisi in carico, in blu, scarico, in verde, mantenimento primario, in rosso, e mantenimento secondario, in magenta. Tutti i dati acquisiti che non ricadono nelle fasi evidenziate sono stati esclusi dall'analisi in quanto superflui.

In Figura 4.2 è presente un ingrandimento dei dati divisi in corrispondenza del mantenimento primario, per apprezzare meglio la divisione in corrispondenza di questo tratto.

## 4.2 La regressione e l'intervallo di predizione

In seguito, viene effettuata la regressione ODR dei dati grezzi nelle fasi di carico e scarico in modo da poter lavorare con delle funzioni e non con i singoli dati sperimentali. Viene individuato l'intervallo di predizione del carico punto per punto.

In Figura 4.4 e Figura 4.6 sono rappresentati graficamente i risultati delle regressioni ODR accostate ai dati sperimentali che le hanno generate. In particolare, negli ingrandimenti di Figura 4.5 e Figura 4.7 si possono vedere meglio i dati ricavati sperimentalmente, rappresentati dai pallini di colore più acceso, la funzione derivata dalla regressione, indicata dalla linea tratteggiata, e l'intervallo di predizione, calcolato come  $f(F) \pm k \cdot U(F)$ , rappresentato dalle linee continue.

I primi due grafici, in blu, si riferiscono alla fase di carico, mentre gli altri due, in verde, alla fase di scarico.

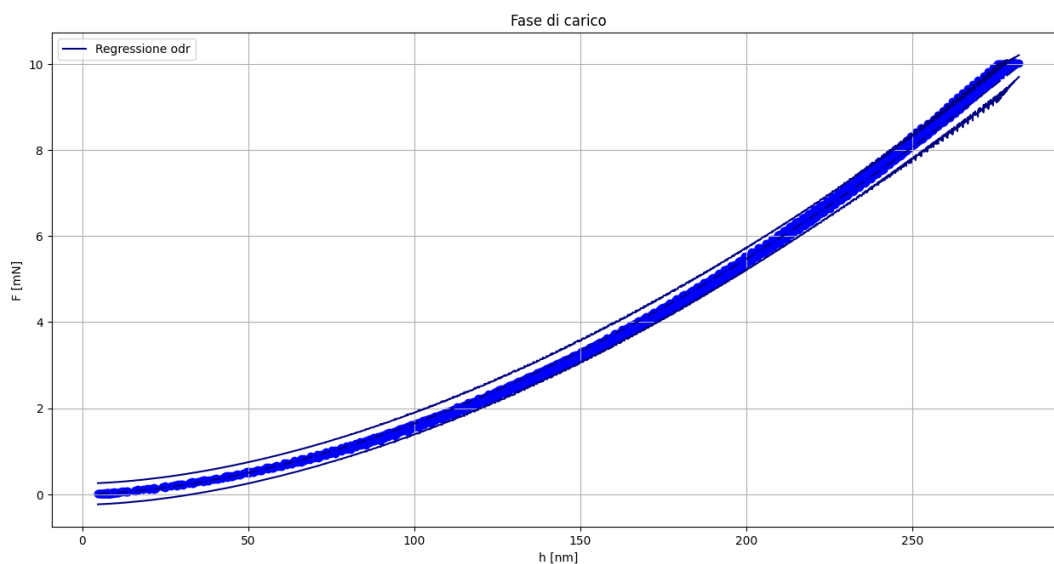


Figura 4.4 Regressione dati fase di carico

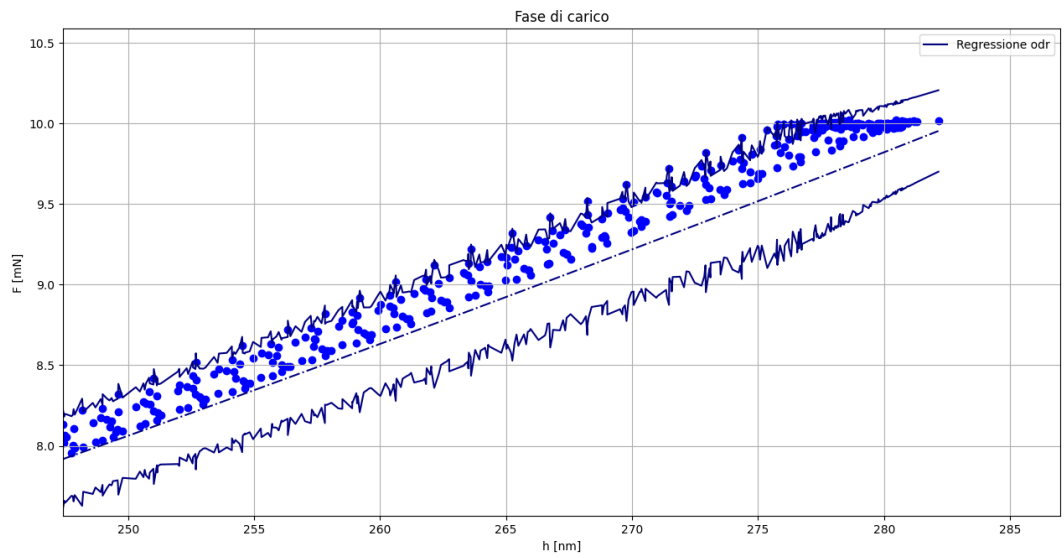


Figura 4.5 Ingrandimento regressione fase di carico

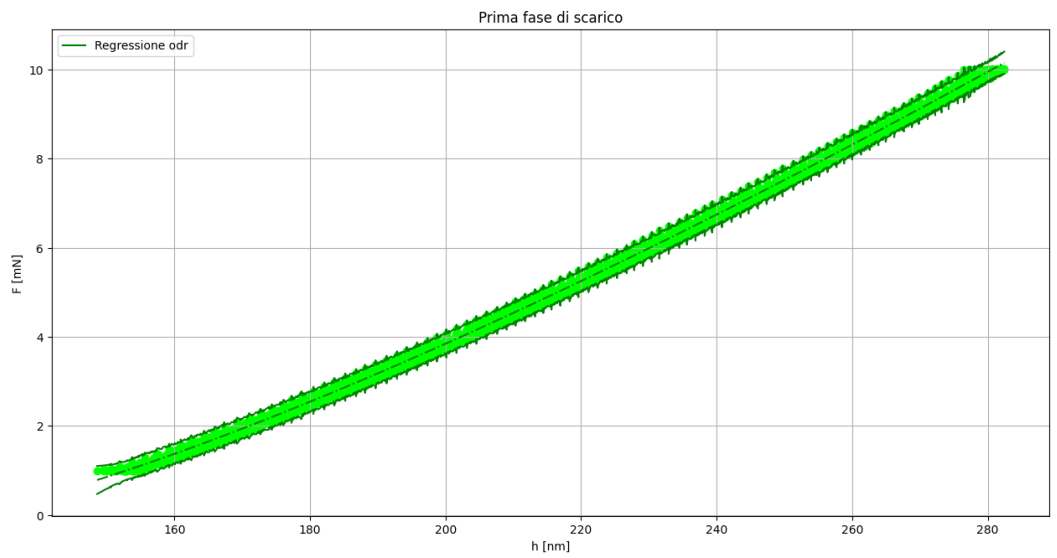


Figura 4.6 Regressione dei dati della fase di scarico



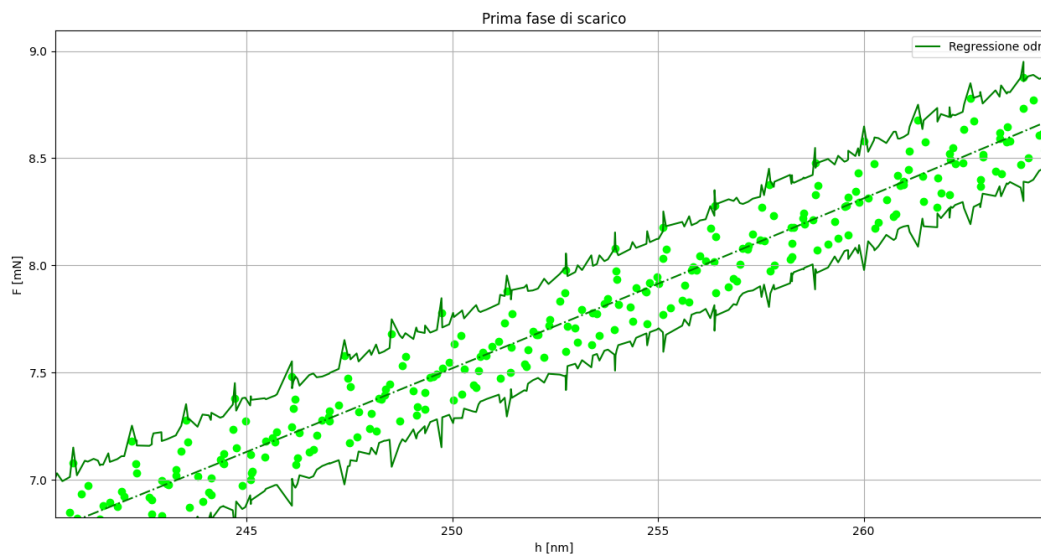


Figura 4.7 Ingrandimento regressione fase di scarico

Si può notare che le linee dell'intervallo di predizione non ha un andamento regolare. Si possono notare, infatti, dei piccoli picchi localizzati. Questi sono dovuti al fatto che l'intervallo di predizione è stato calcolato punto per punto e non si tratta quindi di una funzione come invece è il risultato della regressione.

Analizzando attentamente la Figura 4.8, si nota come tutti i dati sperimentali ricadano all'interno dell'intervallo di predizione. Ci si aspettava questo risultato, in quanto i dati rappresentati sono i dati con i quali l'intervallo stesso è stato costruito. Solo qualora avessimo rappresentato i dati senza considerare l'MSE nella stima dell'incertezza di misura, andando così a costruire l'intervallo di confidenza della nostra funzione, i dati sarebbero potuti ricadere al di fuori della banda.

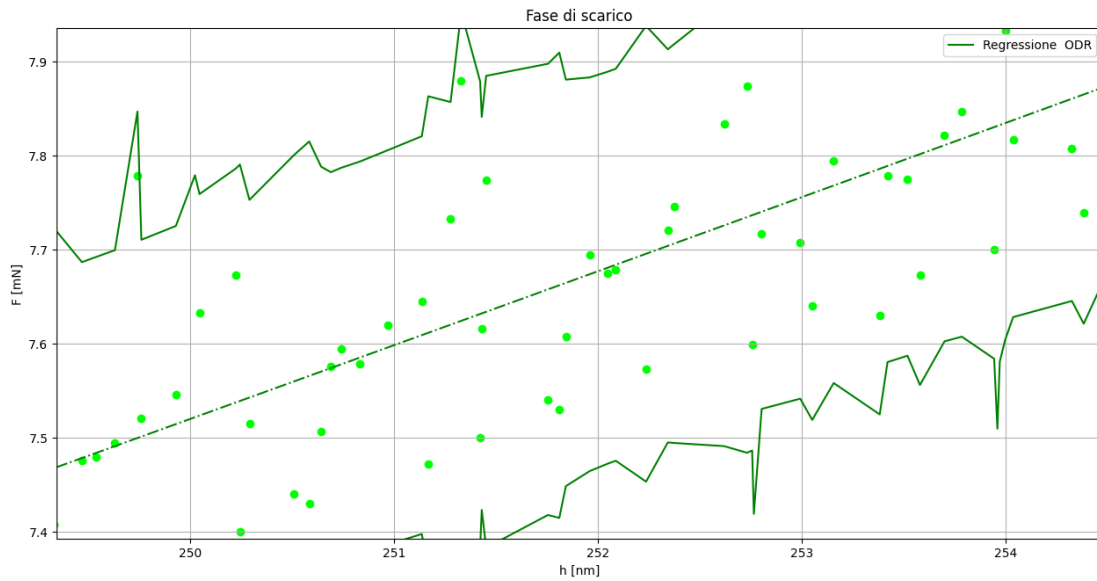


Figura 4.8 Ingrandimento della fase di scarico, nota su MSE

Dopo aver rappresentato le singole fasi è possibile mettere insieme tutte le fasi del ciclo andando a rappresentare quella che è la modellizzazione del ciclo completo effettuata dal DT.

Siccome il controllo di forma rappresenta il motivo principale per il quale è stato necessaria questa modellizzazione del ciclo, non ci si è soffermati sulla modellizzazione delle fasi di mantenimento. Quest'ultime, infatti, risultano trascurabili in tale controllo.

I controlli sulle fasi di mantenimento sono implementati per mezzo dei test d'ipotesi analizzati nei capitoli 3.5.2 e 3.5.3.

Il risultato finale è rappresentato in Figura 4.9, dove si vedono le regressioni, rappresentate con le linee tratteggiate, e i rispettivi intervalli di predizione, rappresentati dalle linee continue.

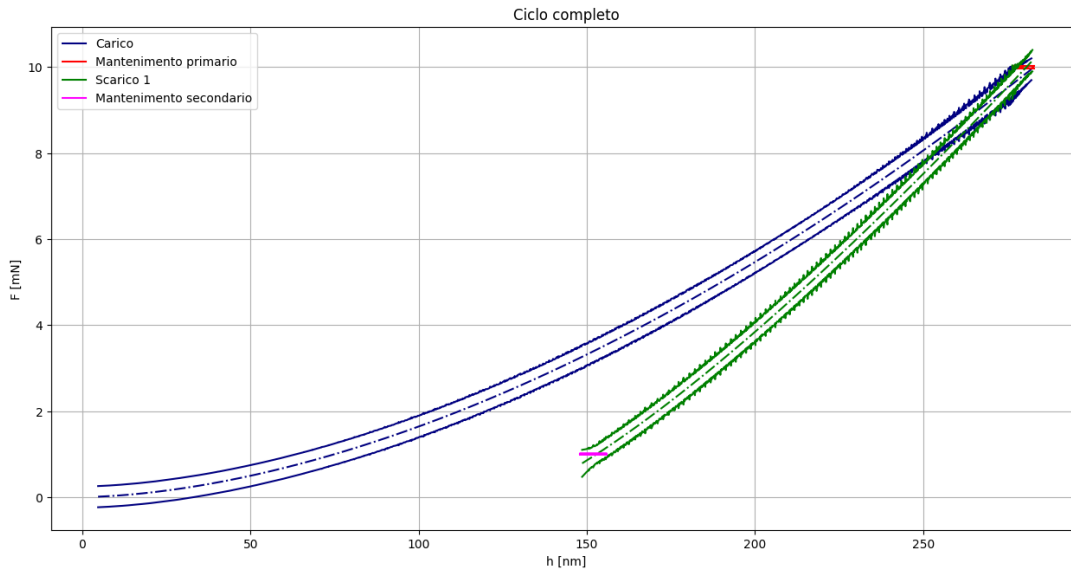


Figura 4.9 Risultato finale regressioni dell'intero ciclo

### 4.3 Calcolo delle proprietà meccaniche del materiale

Di seguito sono stati implementati i calcoli per l'individuazione delle proprietà meccaniche del materiale e della relativa incertezza di misura.

Di seguito, in Tabella 4.1, sono riportati i risultati ottenuti a partire dalle curve di training, ossia i risultati sui quali il modello si baserà per implementare successivamente i tesi d'ipotesi per validare i dati in tempo reale.

	m	u(x)	u%
HIT [GPa]	9,96E+00	3,38E-01	3,39%
EIT [GPa]	7,70E+01	2,60E+00	3,38%
Sm [mN/nm]	8,35E-02	1,23E-04	0,15%
S [mN/nm]	8,42E-02	1,25E-04	0,15%
R [nm/mN]	1,43E+03	1,82E+01	1,27%

Tabella 4.1 Proprietà meccaniche del provino derivate dai dati di training

Si può notare che i risultati sono in linea con quelli attesi e che lo scarto tipo percentuale è contenuto.

Con questa analisi finisce la costruzione del modello ed il DT è pronto all'analisi della curva in tempo reale.

I risultati ottenuti per le 16 curve testate sono riportati in Tabella 4.2.

		HIT [GPa]	EIT [GPa]	Sm [mN/nm]	S [mN/nm]	R [nm/mN]
1	m	1,00E+01	7,70E+01	8,26E-02	8,33E-02	1,44E+03
	u <sup>2</sup> (x)	1,60E-01	9,64E+00	6,18E-07	6,38E-07	7,49E+02
	u%	4,00%	4,03%	0,95%	0,96%	1,90%
2	m	9,83E+00	7,69E+01	8,34E-02	8,41E-02	1,41E+03
	u <sup>2</sup> (x)	1,52E-01	9,42E+00	6,26E-07	6,47E-07	7,25E+02
	u%	3,96%	3,99%	0,95%	0,96%	1,90%
3	m	9,95E+00	7,70E+01	8,29E-02	8,36E-02	1,43E+03
	u <sup>2</sup> (x)	1,58E-01	9,59E+00	6,20E-07	6,42E-07	7,53E+02
	u%	4,00%	4,02%	0,95%	0,96%	1,92%
4	m	9,72E+00	7,69E+01	8,39E-02	8,46E-02	1,40E+03
	u <sup>2</sup> (x)	1,46E-01	9,30E+00	6,35E-07	6,57E-07	7,05E+02
	u%	3,94%	3,97%	0,95%	0,96%	1,90%
5	m	9,93E+00	7,81E+01	8,41E-02	8,49E-02	1,39E+03
	u <sup>2</sup> (x)	1,56E-01	9,82E+00	6,41E-07	6,63E-07	7,01E+02
	u%	3,98%	4,01%	0,95%	0,96%	1,91%
6	m	9,87E+00	7,50E+01	8,12E-02	8,19E-02	1,49E+03
	u <sup>2</sup> (x)	1,54E-01	9,00E+00	5,96E-07	6,16E-07	7,98E+02
	u%	3,97%	4,00%	0,95%	0,96%	1,90%
7	m	1,02E+01	7,66E+01	8,15E-02	8,21E-02	1,48E+03
	u <sup>2</sup> (x)	1,69E-01	9,68E+00	5,99E-07	6,19E-07	7,91E+02
	u%	4,02%	4,06%	0,95%	0,96%	1,90%
8	m	9,76E+00	7,64E+01	8,32E-02	8,39E-02	1,42E+03
	u <sup>2</sup> (x)	1,48E-01	9,19E+00	6,25E-07	6,47E-07	7,25E+02
	u%	3,94%	3,97%	0,95%	0,96%	1,89%
9	m	9,72E+00	7,59E+01	8,28E-02	8,35E-02	1,43E+03
	u <sup>2</sup> (x)	1,47E-01	9,08E+00	6,19E-07	6,40E-07	7,39E+02
	u%	3,94%	3,97%	0,95%	0,96%	1,90%
10	m	9,71E+00	7,58E+01	8,28E-02	8,35E-02	1,44E+03
	u <sup>2</sup> (x)	1,46E-01	9,03E+00	6,18E-07	6,39E-07	7,41E+02
	u%	3,94%	3,96%	0,95%	0,96%	1,90%
11	m	9,84E+00	7,56E+01	8,20E-02	8,27E-02	1,46E+03
	u <sup>2</sup> (x)	1,52E-01	9,09E+00	6,08E-07	6,28E-07	7,69E+02
	u%	3,96%	3,99%	0,95%	0,96%	1,90%
12	m	9,86E+00	7,64E+01	8,27E-02	8,34E-02	1,44E+03
	u <sup>2</sup> (x)	1,53E-01	9,30E+00	6,19E-07	6,40E-07	7,44E+02
	u%	3,96%	3,99%	0,95%	0,96%	1,90%
13	m	9,53E+00	7,55E+01	8,32E-02	8,39E-02	1,42E+03
	u <sup>2</sup> (x)	1,40E-01	8,86E+00	6,26E-07	6,48E-07	7,35E+02
	u%	3,92%	3,94%	0,95%	0,96%	1,91%
14	m	9,83E+00	7,71E+01	8,36E-02	8,43E-02	1,41E+03
	u <sup>2</sup> (x)	1,51E-01	9,45E+00	6,30E-07	6,52E-07	7,18E+02
	u%	3,96%	3,99%	0,95%	0,96%	1,90%
15	m	9,82E+00	7,62E+01	8,27E-02	8,34E-02	1,44E+03
	u <sup>2</sup> (x)	1,51E-01	9,24E+00	6,17E-07	6,38E-07	7,45E+02
	u%	3,96%	3,99%	0,95%	0,96%	1,90%
16	m	9,87E+00	7,59E+01	8,22E-02	8,29E-02	1,46E+03
	u <sup>2</sup> (x)	1,53E-01	9,19E+00	6,09E-07	6,30E-07	7,60E+02
	u%	3,96%	3,99%	0,95%	0,96%	1,89%

Tabella 4.2 Proprietà meccaniche del provino derivate dai dati di test

Anche le curve di test hanno ottenuto risultati che non si discostano da quelli auspicati, senza derive individuabili con un controllo preliminare.

Si nota come l'errore percentuale dei dati in Tabella 4.2 sia leggermente più alto rispetto a quelli in Tabella 4.1. Questo è dovuto al fatto che per il modello di training l'incertezza è calcolata dall'insieme dei dati di più curve sperimentali, mentre in tempo reale si testa una curva alla volta. Il dataset su cui è allenato il modello è più consistente e conseguentemente si raggiunge un'incertezza minore nel calcolo delle proprietà meccaniche.

## 4.4 I test d'ipotesi

### 4.4.1 Test sulle caratteristiche meccaniche

Una volta calcolate le proprietà meccaniche delle curve di test, è possibile implementare i test d'ipotesi necessari a valutare la compatibilità con il modello.

Si vanno a verificare, quindi, tutti i valori appena calcolati. L'ipotesi nulla imposta è l'uguaglianza tra i valori da testare e quelli base, calcolati a partire dai dati di training.

I risultati sono riportati in Tabella 4.3.

	1	2	3	4	5	6	7	8
Test su S	✓	✓	✓	✓	✓	✗	✗	✓
Test su Hit	✓	✓	✓	✓	✓	✓	✓	✓
Test su Eit	✓	✓	✓	✓	✓	✓	✓	✓
Test su R	✓	✓	✓	✓	✓	✓	✓	✓

	9	10	11	12	13	14	15	16
Test su S	✓	✓	✓	✓	✓	✓	✓	✓
Test su Hit	✓	✓	✓	✓	✓	✓	✓	✓
Test su Eit	✓	✓	✓	✓	✓	✓	✓	✓
Test su R	✓	✓	✓	✓	✓	✓	✓	✓

✓: Non si può rifiutare l'ipotesi nulla

✗: Si rifiuta l'ipotesi nulla

Tabella 4.3 Risultati dei test d'ipotesi sulle proprietà meccaniche

Come si evince dalla Tabella 4.2, nella maggior parte delle curve le ipotesi nulle non vengono rifiutate. Le uniche eccezioni si hanno per le curve 6 e 7, dove è rifiutata l'ipotesi nulla relativa al valore di S.

Il parametro S risulta essere il più sensibile in questo tipo di modello. La pendenza del tratto iniziale della curva di scarico, dalla quale si ricava S, risulta molto variabile tra un'indentazione e l'altra. Questo giustifica il rifiuto dell'ipotesi nulla per le curve 6 e 7.

La normativa [19] attutisce parzialmente il problema, eliminando, nel calcolo di S, parte dei punti del tratto di scarico in corrispondenza della zona più sensibile alle variazioni.

Un modo per ottenere risultati più attendibili potrebbe essere eseguire i calcoli considerando insieme i dati di più curve alla volta e applicando i controlli del DT solo dopo qualche indentazione.

Per una comprensione migliore della deriva, si studia la probabilità della  $t$  calcolata per l'ipotesi nulla, per capire quanto si discosta dalla  $t$  di riferimento di Student, corrispondente ad una confidenza del 95%. Si calcola il P value per le curve 6 e 7 dove si rifiuta l'ipotesi nulla per il parametro S. Si ottiene:

$$P_{value}^{[6]} = 1\% \quad P_{value}^{[7]} = 0.4\%$$

Questi valori si discostano dal 5% necessario per la conferma del test d'ipotesi. È necessario però fare una considerazione, in questo lavoro è stato utilizzato un intervallo di fiducia del 95%, solitamente utilizzato in metrologia. Si potrebbe però lavorare anche con il controllo a sei sigma, utilizzato in ingegneria della qualità, che prevede un intervallo di fiducia del 99.73% per cui il risultato del test d'ipotesi sarebbe che non si può rifiutare l'ipotesi nulla anche per queste due curve.

#### 4.4.2 Test sui mantenimenti

Come ultimo controllo sono stati implementati i test d'ipotesi sui due mantenimenti, il primo per verificare il corretto andamento del mantenimento primario e il secondo per verificare la pendenza nulla del mantenimento secondario.

In Tabella 4.4 Risultati dei test d'ipotesi sui mantenimenti sono riportati i risultati dei test. Non sono state riscontrate criticità in nessuna delle curve.

	1	2	3	4	5	6	7	8
Test mantenimento primario	✓	✓	✓	✓	✓	✓	✓	✓
Test mantenimento secondario	✓	✓	✓	✓	✓	✓	✓	✓

	9	10	11	12	13	14	15	16
Test mantenimento primario	✓	✓	✓	✓	✓	✓	✓	✓
Test mantenimento secondario	✓	✓	✓	✓	✓	✓	✓	✓

✓: Non si può rifiutare l'ipotesi nulla

✗: Si rifiuta l'ipotesi nulla

*Tabella 4.4 Risultati dei test d'ipotesi sui mantenimenti*

## 5. Conclusioni e sviluppi futuri

Questo lavoro di tesi ha portato allo sviluppo e all'implementazione di un Digital Twin (DT) metrologico specificamente progettato per ottimizzare il processo di nanoindentazione, una tecnica fondamentale per l'analisi delle proprietà meccaniche dei materiali su scala nanometrica.

L'integrazione tra il Digital Twin e la procedura di nanoindentazione ha richiesto un'approfondita comprensione di entrambe le tecnologie, con l'obiettivo di combinare in modo sinergico le potenzialità di questi strumenti. Questa fase preliminare ha permesso di definire una solida base teorica e metodologica, che ha guidato lo sviluppo del modello.

Il processo di creazione del modello è iniziato con la selezione accurata dei dati sperimentali, seguita da una loro suddivisione in fasi analitiche, ognuna delle quali è stata studiata nei minimi dettagli. L'intero ciclo di nanoindentazione è stato modellizzato con particolare attenzione al calcolo dell'incertezza estesa, concentrandosi in particolare sulle fasi di carico e scarico. Questa modellizzazione ha consentito di determinare le proprietà meccaniche del materiale del provino con le relative varianze composte, sia per i dati impiegati nell'addestramento del modello (dati di training), essenziali per la costruzione di un ambiente virtuale affidabile, sia per quelli utilizzati nella simulazione in tempo reale del DT.

Successivamente, sono stati implementati test di ipotesi volti a verificare l'assenza di anomalie nelle curve di indentazione, confrontandole con quelle generate dal modello durante la fase di training.

I risultati ottenuti hanno mostrato una sostanziale aderenza tra le curve analizzate e il modello teorico, evidenziando una corrispondenza che, nella maggior parte dei casi, non ha mostrato scostamenti significativi.

Tra i parametri studiati, la rigidità  $S$  si è distinta come il più sensibile, essendo l'unico per cui alcuni test hanno rifiutato l'ipotesi nulla. Questo risultato, pur



confermando l'efficacia del modello, suggerisce la necessità di un ulteriore approfondimento su questo parametro specifico.

È importante notare che i risultati sono stati ottenuti in un ambiente metrologico controllato, con condizioni stabili di temperatura e monitoraggio continuo delle vibrazioni tramite accelerometri. Questo ambiente ha garantito l'assenza di interferenze esterne significative, minimizzando le variazioni indesiderate. Tuttavia, in contesti non controllati, il Digital Twin potrebbe esprimere tutto il suo potenziale, rivelando la capacità di rilevare e correggere in tempo reale deviazioni dovute a fattori esterni, quali vibrazioni non controllate, fluttuazioni termiche o malfunzionamenti meccanici.

Un elemento cruciale da considerare è che, data l'integrazione di questo lavoro nel progetto europeo ViDiT, la priorità è stata data alla costruzione del modello a partire dai dati sperimentali (P2V) per rispettare le scadenze imposte dal progetto. Questo ha limitato la possibilità di esplorare ulteriormente le capacità del Digital Twin in condizioni di fuori controllo, lasciando questa parte per futuri sviluppi.

Guardando al futuro, i prossimi passi si concentreranno sull'implementazione dei controlli necessari per l'identificazione e la correzione degli errori di misura (V2P). In primo luogo, sarà essenziale sviluppare carte di controllo per monitorare e garantire che la forma della curva di indentazione non si discosti in modo significativo da quella prevista dal modello generato durante la fase di training. Un eventuale scostamento rilevante potrebbe essere indicativo di problemi quali l'usura della punta dell'indentatore o altri fenomeni di deriva accidentale, come vibrazioni non controllate o malfunzionamenti della piattaforma di misura.

In aggiunta, un passo fondamentale sarà l'integrazione del monitoraggio in tempo reale dei dati provenienti dagli accelerometri, già presenti sulla macchina di misura all'interno del Digital Twin. Questa implementazione permetterà di correlare immediatamente eventuali errori di misura a vibrazioni anomale del sistema, migliorando così la capacità del DT di identificare e correggere deviazioni indesiderate.

Un'ulteriore integrazione sarà l'introduzione di un aggiornamento del modello continuo, andandolo ad allenare con ogni nuova curva considerata valida dal modello stesso. L'introduzione delle nuove curve nel modello di training dovrà avvenire assegnando ad ogni set di dati il giusto peso, dando più importanza ai dati più recenti.

Per completare il progetto, sarà necessario condurre ulteriori campagne sperimentali mirate a investigare singolarmente i vari fattori di deriva che possono influenzare il processo di nanoindentazione, e a valutare come il DT risponda a ciascuno di essi. Questo permetterà di raffinare ulteriormente il modello, garantendo una risposta accurata e tempestiva a qualsiasi variazione nelle condizioni operative.

In prospettiva, è possibile estendere l'applicazione del DT sviluppato a strumentazioni diverse, come la macchina di misura presente presso l'Università degli Studi di Padova (presso il laboratorio TeSi), che collabora con il Politecnico di Torino all'interno del progetto europeo. Questo consentirebbe di testare la robustezza del modello in un ambiente differente, dimostrando la sua versatilità e capacità di adattamento ed eventualmente potrebbe creare un ambiente virtuale condiviso tra più macchine.

In conclusione, l'integrazione del Digital Twin nei processi metrologici rappresenta un significativo avanzamento verso l'ottimizzazione della misura e la riduzione dell'incertezza, soprattutto quando adattato agli standard GUM. L'approccio proposto in questa tesi non solo promette di migliorare l'accuratezza e l'affidabilità delle misure di nanoindentazione, ma apre anche nuove opportunità per l'implementazione di sistemi di controllo avanzati, capaci di rispondere in tempo reale alle variazioni operative. Il DT metrologico sviluppato non rappresenta solo un progresso tecnologico, ma anche un cambiamento paradigmatico nella gestione e nell'interpretazione dei dati di misura, offrendo una piattaforma potente per l'innovazione continua in campo metrologico.

## 6. Bibliografia

- [1] ISO., I.; OIML, BIPM. *Guide to the Expression of Uncertainty in Measurement*. Aenor, 1993.
- [2] DE BIÈVRE, P. The 2012 international vocabulary of metrology: “VIM”. *Chemistry International--Newsmagazine for IUPAC*, 2012, 34.3: 26-27.
- [3] SCHMELTER, S. *Annex I –JRP Protocol: 22DIT01 ViDiT - Trustworthy virtual experiments and digital twins*, ViDiT Project, 2023.
- [4] DIDONÈ, L., SAVIO, E., MACULOTTI, G., et al. *A2.1.1: Selection of the DT model’s fidelity and the other main DT-descriptive parameters*. ViDiT Project, 2023.
- [5] MACULOTTI, G., et al. A Shared Metrological Framework for Trustworthy Virtual Experiments and Digital Twins. *Metrology*, 2024, 4.3: 337-363.
- [6] GRIEVES, M. Digital twin: manufacturing excellence through virtual factory replication.
- [7] ISO/IEC 30173:2023. *Digital twin — Concepts and terminology*. Ginevra: International Organization for Standardization, 2023.
- [8] JONES, D., et al. Characterising the Digital Twin: A systematic literature review. *CIRP journal of manufacturing science and technology*, 2020, 29: 36-52.
- [9] VANDERHORN, E.; MAHADEVAN, S. Digital Twin: Generalization, characterization and implementation. *Decision support systems*, 2021, 145: 113524.
- [10] DE KETELAERE, B., et al. Digital twins in quality engineering. *Quality Engineering*, 2022, 34.3: 404-408.

- [11] TABOR, D., e TAYLOR, G. I. 1948. A simple theory of static and dynamic hardness. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 192(1029), pp. 247-274.
- [12] BULYCHEV, S. I., et al. Determination of Young modulus by the hardness indentation diagram. [Mechanical properties as tested on some steels, Ti-Zr-Mo alloy; aluminium alloys]. *Zavodskaya Laboratoriya*, 41(9)
- [13] PETHICA, J. B., HUTCHINGS, R., e OLIVER, W. C. 1983. Hardness measurement at penetration depths as small as 20 nm. *Philosophical Magazine A*, 48(4), pp. 593–606
- [14] OLIVER, W. C.; PHARR, G. M. An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments. *Journal of materials research*, 1992, 7.6: 1564-1583.
- [15] OLIVER, W. C.; PHARR, G. M. Measurement of hardness and elastic modulus by instrumented indentation: Advances in understanding and refinements to methodology. *Journal of materials research*, 2004, 19.1: 3-20.
- [16] BOUSSINESQ, J. 1885. *Application des potentiels à l'étude de l'équilibre et du mouvement des solides élastiques*. Parigi: Gauthier-Villars.
- [17] HERTZ, H. 1882. *J. reine und angewandte Mathematik*, 92, p. 156.
- [18] HARDING, J. W., e SNEDDON, I. N. 1945. The elastic stresses produced by the indentation of the plane of a semi-infinite elastic solid. *Proceedings of the Cambridge Philosophical Society*, p. 12.
- [19] ISO 14577-1:2014, Metallic materials - Instrumented indentation test for hardness and materials parameters - Part 1: Test method. 2nd edition.
- [20] VERMA, S.; SARKAR, P.; PANT, P. Thermal drift in room temperature nanoindentation experiments: Measurement and correction. *Journal of Materials Research*, 2021, 36: 3436-3444
- [21] GALETTO, M.; KHOLKHUJAEV, J.; MACULOTTI, G. Improvement of instrumented indentation test accuracy by data augmentation with electrical contact resistance. *CIRP Annals*, 2023, 72.1: 469-472.
- [22] BARBATO, G.; GERMAK, A.; D'AGOSTINO, D. Misurare per decidere. *Progetto Leonardo, Bologna*, 2005, 229-242.
- [23] BOGGS, P. T., et al. *User's Reference Guide for ODRPACK Version 2.01 Software for Weighted Orthogonal Distance Regression*. Gaithersburg: National Institute of Standards and Technology, 1992.

## 7. Appendice

È di seguito riportato il codice Python del digital twin realizzato.

```
import numpy as np
import math
from scipy.optimize import curve_fit
from scipy.stats import t
import matplotlib.pyplot as plt
from scipy.odr import ODR, Model, RealData
from scipy.io import savemat
import os
import re

def apri_file_Torino(file_path):

    x=np.array([])
    y=np.array([])
    t=np.array([])

    # Apre il file e legge le righe
    with open(file_path, 'r') as file:
        # Salta le righe fino all'intestazione
        for line in file:
            if line.startswith("Time (s)"):
                break
        # Legge il resto del file
        for line in file:
            # Divide la riga in una lista di valori
            values = line.strip().split()

            if len(values) >= 3 and values[4] != '0':
                break

    # Legge il resto del file
    for line in file:
        # Divide la riga in una lista di valori
        values = line.strip().split()
        # Controlla se la lista ha abbastanza elementi e li salva in vettori temporanei
        if len(values) >= 3:
            y=np.append(y, float(values[2].replace(',','.')))
            x=np.append(x, float(values[1].replace(',','.')))
            t=np.append(t, float(values[0].replace(',','.')))

    return x, y, t

def divisione_mantenimento(lista_file): #divide i dati del file in carico, mantenimento
primario, scarico e mantenimento secondario

    #inizializzo i vettori
    x_carico = np.array([])
    x_mantenimento_1 = np.array([])
    x_scarico_1 = np.array([])
    y_carico = np.array([])
    y_mantenimento_1 = np.array([])
```

```

t_mantenimento_1 = np.array([])
y_scarico_1 = np.array([])
x_mantenimento_2 = np.array([])
y_mantenimento_2 = np.array([])
t_mantenimento_2 = np.array([])

for file_path in lista_file:
    # Apre il file e legge le righe

    x,y,t= apri_file_Torino(file_path)

    # Definisco un intervallo di test di mantenimento primario e secondario in base
    al tempo e alla frequenza di campionamento
    # Calcolo la media e lo scarto tipo per determinare quali dati inserire nel
    mantenimento
    test1_1= 130
    test1_2= 160
    mant_1_test=y[test1_1:test1_2]
    media_mant_1= np.mean(mant_1_test)
    std_mant_1=np.std(mant_1_test)
    sup1= media_mant_1 + 2* std_mant_1
    inf1= media_mant_1 - 2* std_mant_1
    test2_1= 350
    test2_2= 750
    mant_2_test=y[test2_1: test2_2]
    media_mant_2= np.mean (mant_2_test)
    std_mant_2=np.std(mant_2_test)
    sup2= media_mant_2 + 2* std_mant_2
    inf2= media_mant_2 - 2* std_mant_2

    #divido i dati in carico, mantenimento primario, scarico e mantenimento secondario
    for i in range(len(y)):
        if y[i]>=inf1 and y[i]<=sup1:
            x_mantenimento_1= np.append(x_mantenimento_1, x[i])
            y_mantenimento_1= np.append(y_mantenimento_1, y[i])
            t_mantenimento_1= np.append(t_mantenimento_1, t[i])

        elif i==0 or y[i]>y[i-1] and i< test1_1 :
            x_carico=np.append(x_carico, x[i])
            y_carico= np.append(y_carico, y[i])

        elif y[i]<y[i-1] and y[i]> sup2 and i<test2_1:
            x_scarico_1=np.append(x_scarico_1, x[i])
            y_scarico_1= np.append(y_scarico_1, y[i])

        elif y[i]>=inf2 and y[i]<=sup2 and i>test1_2:
            x_mantenimento_2= np.append(x_mantenimento_2, x[i])
            y_mantenimento_2= np.append(y_mantenimento_2, y[i])
            t_mantenimento_2= np.append(t_mantenimento_2, t[i])

    return x_carico, y_carico, x_mantenimento_1, y_mantenimento_1, t_mantenimento_1,
    x_scarico_1, y_scarico_1, x_mantenimento_2, y_mantenimento_2, t_mantenimento_2

def funz_interp_ODR_esp(beta, x): #funzione per la regressione di carico e scarico nel
formato per l'ODR
    alfa, b, m = beta
    return alfa * (x - b) ** m

def funz_ODR_mant(beta, t): #funzione per la regressione della regressione nel formato
per l'ODR
    q, m = beta
    return m*t + q

def funz_interp(x, alfa, b, m): #funzione interpolante di carico e scarico
    return alfa* (x-b)**m

def funz_area(h, a2, a1, a0): #funzione di area
    return a2*h**2 + a1*h +a0*h**(1/2)

def funz_mant(t, h0, m): #funzione del mantenimento

```

```

return h0 + t* m

def regressione_odr(h, f, u_h, u_f, beta0, function): #Funzione per la regressione odr

    data= RealData(h, f, sx=u_h, sy=u_f)
    model=Model(function)
    odr=ODR(data, model, beta0=beta0, sstol=1e-6)
    output=odr.run()

    #output: parametri della regressione, matrice delle covarianze e errore
    dell'affondamento
    parametri=output.beta
    mat_cov= output.cov_beta
    delta= output.delta

    return parametri, mat_cov, delta

def calcolo_incertezza(h, f, coeff, matrice_coeff, delta): # calcola l'incertezza estesa
delle fasi di carico e scarico

    # Estraggo i parametri della regressione
    alfa, b, m = coeff
    # Estraggo le varianze dei parametri
    alfa_var, b_var, m_var = np.diag(matrice_coeff)
    #Definizione della matrice per il calcolo dell'incertezza
    matrice_cov = np.zeros((5, 5), dtype=float)

    # Copia i valori dalla matrice dei parametri della regressione
    matrice_cov[:3, :3] = matrice_coeff

    #calcolo dei residui e del mse
    residui =np.array(f - funz_interp(h, alfa, b, m))

    mse= np.var(residui) #mse calcolato come varianza dei residui

    #Lo aggiungo alla matrice
    matrice_cov[3,3]= mse

    #####calcolo incertezza estesa#####

    #espando nella terza dimensione per non dover fare un ciclo
    u_h_vet = delta/np.sqrt(3)
    matrice_cov_grande = np.broadcast_to(matrice_cov, (len(h),) +
        matrice_cov.shape).copy()
    matrice_cov_grande[:,4,4] = u_h_vet**2
    dh_vet = 1/100 * u_h_vet

    #incrementi infinitesimali visti come 1% della varianza dei parametri
    dalfa= 1/100 * math.sqrt(alfa_var)
    db= 1/100 * math.sqrt(b_var)
    dm= 1/100 * math.sqrt(m_var)
    #Calcolo delle derivate parziali con la funzione incrementale
    df_dh_vet = (funz_interp(h + dh_vet, alfa, b, m)-funz_interp(h, alfa, b, m))/dh_vet
    df_da_vet = (funz_interp(h, alfa + dalfa, b, m)-funz_interp(h, alfa, b, m))/dalfa
    df_db_vet = (funz_interp(h, alfa, b + db, m)-funz_interp(h, alfa, b, m))/db
    df_dm_vet = (funz_interp(h, alfa, b, m + dm)-funz_interp(h, alfa, b, m))/dm

    der_parziali_mat = np.array([df_da_vet, df_db_vet,
        df_dm_vet,np.ones(len(h)),df_dh_vet]).transpose()

    #calcolo della varianza composta
    u_sq_f = np.einsum('ij,ijk,ik->i', der_parziali_mat, matrice_cov_grande,
        der_parziali_mat)

    #incertezza tipo
    u_f= np.sqrt(u_sq_f)

    # Calcolo dell'inversa della distribuzione t
    dof= 100
    liv_fiducia = 0.95 # Livello di fiducia del 95%

```

```

k = t.ppf(1 - (1- liv_fiducia)/2, dof)

inc_f= k * u_f #Incertezza estesa carico

return u_sq_f, inc_f

def ordina_vettori(x, y, inc, u_sq, delta): # Ordina i vettori in base a x crescente
# Ordina gli indici di x in base ai valori di x
sorted_indices = np.argsort(x)

# Ordina x e y in base agli indici ordinati
x_ok = x[sorted_indices]
y_ok = y[sorted_indices]
inc_ok = inc[sorted_indices]
u_sq_ok = u_sq[sorted_indices]
delta_ok = delta[sorted_indices]
return x_ok, y_ok, inc_ok, u_sq_ok, delta_ok

def divisione_e_regressione(lista_file): #Prende i dati dai file di output della macchina
li divide, fa la regressione ODR e calcola l'incertezza estesa, inoltre fa i plot

h1, f1, h_man_1, f_man_1, t_man_1, h2, f2, h_man_2, f_man_2, t_man_2 =
divisione_mantenimento(lista_file) #divisione dei dati
# h [nm], f [mN], t[s]

#FASE DI CARICO
#scarto tipo sui dati sperimentali, dovuti principalmente al contributo di accuratezza
u_h1= 0.0006 * h1
u_f1= 0.00058 * f1

#regressione preliminare per determinare i parametri di partenza dell'ODR
limiti_parametri= ([0, -np.inf, 1],[np.inf, np.inf, 2])
beta0_carico, matrice_coeff0_carico = curve_fit(funz_interp, h1, f1,
bounds=limiti_parametri, method= 'dogbox')

#regressione dei dati sperimentali con ODR
coeff1, mat_cov1, delta1 = regressione_odr(h1, f1, u_h1, u_f1, beta0_carico,
funz_interp_ODR_esp)

#calcolo incertezza estesa
u_sq_f1, inc_f1= calcolo_incertezza(h1, f1, coeff1, mat_cov1, delta1)

#calcolo della banda di incertezza per il plot
h1_ord, f1_ord, inc_f1_ord, u_sq_f1_ord, delta1_ord= ordina_vettori(h1,
f1,inc_f1,u_sq_f1,delta1) #ordino i dati con le x crescenti per poterli plottare
f1_up_plot= funz_interp(h1_ord, *coeff1) + inc_f1_ord
f1_d_plot= funz_interp(h1_ord, *coeff1) - inc_f1_ord

#FASE DI SCARICO
u_h2= 0.0006 * h2
u_f2= 0.00058 * f2

#regressione preliminare per determinare i parametri di partenza dell'ODR beta0
limiti_parametri= ([0, -np.inf, 1],[np.inf, np.inf, 2])
beta0_scarico, matrice_coeff0_scarico = curve_fit(funz_interp, h2, f2,
bounds=limiti_parametri, method= 'dogbox')

#regressione dei dati sperimentali con ODR
coeff2, mat_cov2, delta2= regressione_odr(h2,f2, u_h2, u_f2, beta0_scarico,
funz_interp_ODR_esp)

#calcolo incertezza estesa
u_sq_f2, inc_f2 = calcolo_incertezza(h2, f2, coeff2, mat_cov2, delta2)

#calcolo della banda di incertezza per il plot
h2_ord, f2_ord, inc_f2_ord, u_sq_f2_ord, delta2_ord= ordina_vettori(h2,f2,inc_f2,
u_sq_f2, delta2) #ordino i dati con le x crescenti per poterli plottare
f2_up_plot= funz_interp(h2_ord, *coeff2) + inc_f2_ord
f2_d_plot= funz_interp(h2_ord, *coeff2) - inc_f2_ord

```



```

#FASE DI MANTENIMENTO PRIMARIO
h_man_1_ord= np.linspace(np.min(h_man_1),np.max(h_man_1))

mean_mant_1=np.mean(f_man_1)
media_man_1 = np.ones(len(h_man_1_ord)) * mean_mant_1
u_man_1 = np.ones(len(h_man_1_ord)) * np.std(f_man_1)

#FASE DI MANTENIMENTO SECONDARIO
h_man_2_ord= np.linspace(np.min(h_man_2),np.max(h_man_2))

mean_mant_2=np.mean(f_man_2)
media_man_2 = np.ones(len(h_man_2)) * mean_mant_2
u_man_2 = np.ones(len(h_man_2)) * np.std(f_man_2)

##### PLOT DATI #####

y_n= input("Vuoi visualizzare i plot? (y/n) ")
if y_n== 'y':

    #Dati sperimentali selezionati e divisi
    plt.scatter(h1,f1, color='blue', label='Carico')
    plt.scatter(h2,f2, color='lime', label='Scarico')
    plt.scatter(h_man_1, f_man_1, color='red', label='Mantenimento primario')
    plt.scatter(h_man_2,f_man_2, color='magenta',label='Mantenimento secondario')
    plt.xlabel('h [nm]')
    plt.ylabel('F [mN]')
    plt.grid((True))
    plt.title('Ciclo completo: DATI SPERIMENTALI')
    plt.legend()
    plt.show()

    #Fase di carico
    plt.plot(h1_ord, funz_interp(h1_ord, *coeff1), color='navy', linestyle= '-. ')
    plt.plot(h1_ord, f1_up_plot, color='navy', label='Regressione ODR')
    plt.plot(h1_ord, f1_d_plot, color='navy')
    plt.scatter(h1,f1, color='blue')
    plt.xlabel('h [nm]')
    plt.ylabel('F [mN]')
    plt.grid((True))
    plt.title('Fase di carico')
    plt.legend()
    plt.show()

    #Fase di scarico primario
    plt.plot(h2_ord, funz_interp(h2_ord, *coeff2), color='green', linestyle= '-. ')
    plt.plot(h2_ord, f2_up_plot, color='green', label='Regressione ODR')
    plt.plot(h2_ord, f2_d_plot, color='green')
    plt.scatter(h2,f2, color='lime')

    plt.xlabel('h [nm]')
    plt.ylabel('F [mN]')
    plt.grid((True))
    plt.title('Fase di scarico')
    plt.legend()
    plt.show()

    # ciclo completo
    plt.plot(h1_ord, funz_interp(h1_ord, *coeff1), color='navy', linestyle= '-. ')
    plt.plot(h1_ord, f1_up_plot, color='navy', label='Carico')
    plt.plot(h1_ord, f1_d_plot, color='navy')

    plt.plot(h2_ord, funz_interp(h2_ord, *coeff2), color='green', linestyle= '-. ')
    plt.plot(h2_ord, f2_up_plot, color='green', label='Scarico')
    plt.plot(h2_ord, f2_d_plot, color='green')

    plt.plot(h_man_1_ord, media_man_1, color='red', linestyle= '-. ')
    plt.plot(h_man_1_ord, media_man_1 + 2*u_man_1, color='red', label='Mantenimento
primario')

```

```

plt.plot(h_man_1_ord, media_man_1 - 2*u_man_1, color='red')

plt.plot(h_man_2, media_man_2, color='magenta', linestyle= '-.')
plt.plot(h_man_2, media_man_2 + 2*u_man_2, color='magenta', label='Mantenimento
secondario')
plt.plot(h_man_2, media_man_2 - 2*u_man_2, color='magenta')

plt.xlabel('h [nm]')
plt.ylabel('F [mN]')
plt.grid(True)
plt.title('Ciclo completo')
plt.legend()
plt.show()

return h1_ord, f1_ord, inc_f1_ord, u_sq_f1_ord, delta1_ord, coeff1, mat_cov1, h_man_1,
f_man_1, t_man_1, h2_ord, f2_ord, inc_f2_ord, u_sq_f2_ord, delta2_ord, coeff2,
mat_cov2, h_man_2, f_man_2, t_man_2

def analisi_training(h2, parametri1, parametri2, mat_cov1, mat_cov2, u_sq_f2, delta2,
file_path_indentatore): #ATTENZIONE! h2, u_sq_f2 e delta2 devono essere prima ordinati!

    hm = h2[-1] #[nm]
    u_hm = delta2[-1]/np.sqrt(3)

    fmax = funz_interp(hm, *parametri2) #[mN]
    u_fmax= np.sqrt(u_sq_f2[-1])

    h0 = parametri1[1] #[nm]
    u_h0_sq = mat_cov1[1,1] #[nm]

    ##### CALCOLO DI SM #####
    #per il calcolo di sm utilizzo i parametri calcolati nella regressione odr della fase
di scarico
    sm = parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1) #[mN/nm]

    mat_cov_sm = np.zeros((4, 4), dtype=float)
    mat_cov_sm[:3, :3] = mat_cov2
    mat_cov_sm[3, 3] = u_hm ** 2

    # calcolo delle derivate parziali
    dhm = 1e-2 * u_hm
    da = 1e-2 * np.sqrt(mat_cov2[0, 0])
    db = 1e-2 * np.sqrt(mat_cov2[1, 1])
    dc = 1e-2 * np.sqrt(mat_cov2[2, 2])

    dsm_dhm = ((parametri2[0]*parametri2[2]*((hm+dhm)-parametri2[1]) ** (parametri2[2]-
1)) - (parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1))) / dhm
    dsm_da=((parametri2[0]+da)*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1)) -
(parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1)) / da
    dsm_db= ((parametri2[0]*parametri2[2]*(hm-(parametri2[1]+db)) ** (parametri2[2]-1)) -
(parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1)) / db
    dsm_dc=((parametri2[0]*(parametri2[2]+dc)*(hm-parametri2[1]) ** ((parametri2[2]+dc)-
1)) - (parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1))) / dc

    der_parziali_sm = np.array([dsm_da, dsm_db, dsm_dc, dsm_dhm])

    # Varianza della variabile indipendente
    u_sm_sq = np.dot(np.dot(der_parziali_sm, mat_cov_sm), der_parziali_sm.T)

    # # Incertezza tipo sm
    u_sm = np.sqrt(u_sm_sq) #[mN/nm]
    u_sm_perc = u_sm / sm

    #####
    #Funzione d'area a partire dal file dell'indentatore

    hi= []
    Ai= []

    with open(file_path_indentatore, 'r') as file:

```

```

# Salta le righe fino a trovare l'intestazione
for line in file:
    if line.startswith("hi[m]"):
        break

# Leggi le righe dei dati
for line in file:
    # Sostituisci la virgola con il punto
    line = line.replace(',', '.')
    # Divide la riga in una lista di valori
    values = line.strip().split()

    # Controlla se la lista ha abbastanza elementi
    if len(values) >= 2:
        # Converti a float e aggiungi alle liste
        hi.append(float(values[0]) * 10**9) # Conversione da metri a nanometri
        Ai.append(float(values[1]) * 10**18) # Conversione da metri quadrati a
nanometri quadrati

# Converti le liste in array numpy
hi = np.array(hi) #[nm]
Ai = np.array(Ai) #[nm^2]

# Adatta la curva
vett_a, mat_a = curve_fit(funz_area, hi, Ai)
a2, a1, a0 = vett_a
u_a2_sq, u_a1_sq, u_a0_sq = np.diag(mat_a)

cf = 0.1 #[nm/mN]
u_cf_sq = (0.0001**2)/3

ctot = 1 / sm # [mN/nm]
u_ctot = ctot * u_sm_perc

cs = ctot - cf #[mN/nm]
u_cs_sq = u_cf_sq + u_ctot ** 2
u_cs_perc = np.sqrt(u_cs_sq) /cs

s = 1 / cs #[nm/mN]
u_s = s * u_cs_perc
u_s_sq = u_s **2

nus = 0.161
u_nus_sq = (0.003/2)**2 #tarato
nui= 0.07
u_nui_sq = (0.02**2) /3 #PUMA
ei = 1141 #[GPa]
u_ei_sq = (2**2) /3 #PUMA
eps = 0.76
u_eps_sq = (0.02**2) /3 #PUMA

hc = hm - h0 - (cf + eps * cs) * fmax #[nm]

dcf = 0.01 * np.sqrt(u_cf_sq)
deps = 0.01 * np.sqrt(u_eps_sq)
dcs = 0.01 * np.sqrt(u_cs_sq)
dfmax = 0.01 * u_fmax

dhc_cf = ((hm - h0 - (cf + dcf + eps * cs) * fmax) - hc) / dcf
dhc_eps = ((hm - h0 - (cf + (eps + deps) * cs) * fmax) - hc) / deps
dhc_cs = ((hm - h0 - (cf + eps * (cs + dcs)) * fmax) - hc) / dcs
dhc_fmax = ((hm - h0 - (cf + eps * cs) * (fmax + dfmax)) - hc) / dfmax

u_hc_sq = np.sum(np.array([u_hm ** 2, u_h0_sq, u_cf_sq, u_eps_sq, u_cs_sq, u_fmax **
    2]) * np.array([1, -1, dhc_cf, dhc_eps, dhc_cs, dhc_fmax]) ** 2)

ap = a2 * hc ** 2 + a1 * hc + a0 #[nm^2]
u_ap_sq = np.sum(np.array([u_a2_sq, u_a1_sq, u_a0_sq, u_hc_sq]) * np.array([hc ** 2,
    hc, 1, 2 * a2 * hc + a1]) ** 2)

```

```

hit =10**6 * fmax / ap #[GPa]
u_hit_sq =10**12 * np.sum(np.array([u_fmax ** 2, u_ap_sq]) * np.array([1 / ap, -fmax
    / (ap ** 2)]) ** 2)

er =10**6 * (s * math.pi ** (1 / 2)) / (2 * ap ** (1 / 2)) #[GPa]

der_s = (math.pi ** (1 / 2)) / (2 * ap ** (1 / 2))
der_ap = (-s * math.pi ** (1 / 2)) / (2 * ap ** (3 / 2))

u_er_sq =10**12 * np.sum(np.array([u_s ** 2, u_ap_sq]) * np.array([der_s, der_ap])**2)

eit = (1 - nus ** 2) / (1 / er - (1 - nui ** 2) / ei) #[GPa]

dnus = 0.01 * np.sqrt(u_nus_sq)
dnui = 0.01 * np.sqrt(u_nui_sq)
der = 0.01 * np.sqrt(u_er_sq)
dei = 0.01 * np.sqrt(u_ei_sq)

deit_nus = ((1 - (nus + dnus) ** 2) / (1 / er - (1 - nui ** 2) / ei) - eit) / dnus
deit_nui = ((1 - nus ** 2) / (1 / er - (1 - (nui + dnui) ** 2) / ei) - eit) / dnui
deit_er = ((1 - nus ** 2) / (1 / (er + der) - (1 - nui ** 2) / ei) - eit) / der
deit_ei = ((1 - nus ** 2) / (1 / er - (1 - nui ** 2) / (ei + dei)) - eit) / dei

u_eit_sq = np.sum(np.array([u_nus_sq, u_er_sq, u_nui_sq, u_ei_sq]) *
    np.array([deit_nus, deit_er, deit_nui, deit_ei]) ** 2)

R = fmax / s ** 2 #nm/mN
u_R_sq = np.sum(np.array([u_fmax ** 2, u_s ** 2]) * np.array([1 / s ** 2, -2 * fmax /
    s ** 3]) ** 2)

return hit, u_hit_sq, eit, u_eit_sq, sm, u_sm_sq, s, u_s_sq, R, u_R_sq

def analisi_new(h2, f2, parametril, mat_cov1, u_sq_f2, delta2, file_path_indentatore):
#ATTENZIONE! h2, f2, u_sq_f2 e delta2 devono essere quelli ordinati!

    hm = h2[-1]
    u_hm = delta2[-1]

    fmax = f2[-1]
    u_fmax= np.sqrt(u_sq_f2[-1])

    h0 = parametril[1]
    u_h0_sq = mat_cov1[1,1]

    ##### CALCOLO DI SM #####
    limiti_parametri= ([0 , -np.inf , 1],[np.inf , np.inf , 2])
    beta0_sm, matrice_coeff0_sm = curve_fit(funz_interp, h2, f2, bounds=limiti_parametri,
        method= 'dogbox')

    start = np.argwhere(f2 < 0.98*f2[-1])[0].item()
    finish = np.argwhere(f2 < 0.2*f2[-1])[0].item()
    if finish == 0:
        finish = np.size(f2)

    h2_sm = h2[start:finish+1]
    f2_sm = f2[start:finish+1]
    u_h2_sm=0.0006 * h2_sm
    u_f2_sm= 0.00058 * f2_sm

    #regressione dei dati sperimentali con ODR
    parametri2, mat_cov2_sm, delta2_sm= regressione_odr(h2_sm, f2_sm, u_h2_sm, u_f2_sm,
        beta0_sm, funz_interp_ODR_esp)

    sm = parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1)

    mat_cov_sm = np.zeros((4, 4), dtype=float)
    mat_cov_sm[:3, :3] = mat_cov2_sm
    mat_cov_sm[3, 3] = u_hm ** 2

```

```

dhm = 1e-2 * u_hm
da = 1e-2 * np.sqrt(mat_cov2_sm[0, 0])
db = 1e-2 * np.sqrt(mat_cov2_sm[1, 1])
dc = 1e-2 * np.sqrt(mat_cov2_sm[2, 2])

dsm_dhm = ((parametri2[0]*parametri2[2]*(hm+dhm)-parametri2[1]) ** (parametri2[2]-1)) - (parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1)) / dhm
dsm_da=((parametri2[0]+da)*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1)) - (parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1)) / da
dsm_db= ((parametri2[0]*parametri2[2]*(hm-(parametri2[1]+db)) ** (parametri2[2]-1)) - (parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1)) / db
dsm_dc=((parametri2[0]*(parametri2[2]+dc)*(hm-parametri2[1]) ** ((parametri2[2]+dc)-1)) - (parametri2[0]*parametri2[2]*(hm-parametri2[1]) ** (parametri2[2]-1))) / dc

der_parziali_sm = np.array([dsm_da, dsm_db, dsm_dc, dsm_dhm])

# Varianza della variabile indipendente
u_sm_sq = np.dot(np.dot(der_parziali_sm, mat_cov_sm), der_parziali_sm.T)

# # Incertezza tipo sm
u_sm = np.sqrt(u_sm_sq)
u_sm_perc = u_sm / sm

#####
#Funzione d'area a partire dal file dell'indentatore

hi= []
Ai= []

with open(file_path_indentatore, 'r') as file:
    # Salta le righe fino a trovare l'intestazione
    for line in file:
        if line.startswith("hi[m]"):
            break

    # Leggi le righe dei dati
    for line in file:
        # Sostituisci la virgola con il punto
        line = line.replace(',', '.')
        # Divide la riga in una lista di valori
        values = line.strip().split()

        # Controlla se la lista ha abbastanza elementi
        if len(values) >= 2:
            # Converti a float e aggiungi alle liste
            hi.append(float(values[0]) * 10**9) # Conversione da metri a nanometri
            Ai.append(float(values[1]) * 10**18) #Conversione da metri^2 a nanometri^2

# Converti le liste in array numpy
hi = np.array(hi) #[nm]
Ai = np.array(Ai) #[nm^2]

# Adatta la curva
vett_a, mat_a = curve_fit(funz_area, hi, Ai)
a2, a1, a0 = vett_a
u_a2_sq, u_a1_sq, u_a0_sq = np.diag(mat_a)

cf = 0.1 #[nm/mN]
u_cf_sq = 2.89E-8

ctot = 1 / sm # [mN/nm]
u_ctot = ctot * u_sm_perc

cs = ctot - cf #[mN/nm]
u_cs_sq = u_cf_sq + u_ctot ** 2
u_cs_perc = np.sqrt(u_cs_sq) /cs

s = 1 / cs #[nm/mN]
u_s = s * u_cs_perc
u_s_sq = u_s **2

```

```

nus = 0.161
u_nus_sq = (0.003/2)**2 #tarato
nui= 0.07
u_nui_sq = (0.02**2) /3 #PUMA
ei = 1141 #[GPa]
u_ei_sq = (2**2) /3 #PUMA
eps = 0.76
u_eps_sq = (0.02**2) /3 #PUMA

hc = hm - h0 - (cf + eps * cs) * fmax #[nm]

dcf = 0.01 * np.sqrt(u_cf_sq)
deps = 0.01 * np.sqrt(u_eps_sq)
dcs = 0.01 * np.sqrt(u_cs_sq)
dfmax = 0.01 * u_fmax

dhc_cf = ((hm - h0 - (cf + dcf + eps * cs) * fmax) - hc) / dcf
dhc_eps = ((hm - h0 - (cf + (eps + deps) * cs) * fmax) - hc) / deps
dhc_cs = ((hm - h0 - (cf + eps * (cs + dcs)) * fmax) - hc) / dcs
dhc_fmax = ((hm - h0 - (cf + eps * cs) * (fmax + dfmax)) - hc) / dfmax

u_hc_sq = np.sum(np.array([u_hm ** 2, u_h0_sq, u_cf_sq, u_eps_sq, u_cs_sq, u_fmax **
2]) * np.array([1, -1, dhc_cf, dhc_eps, dhc_cs, dhc_fmax])) ** 2)

ap = a2 * hc ** 2 + a1 * hc + a0 #[nm^2]
u_ap_sq = np.sum(np.array([u_a2_sq, u_a1_sq, u_a0_sq, u_hc_sq]) * np.array([hc ** 2,
hc, 1, 2 * a2 * hc + a1])) ** 2)

hit =10**6 * fmax / ap #[GPa]
u_hit_sq =10**12 * np.sum(np.array([u_fmax ** 2, u_ap_sq]) * np.array([1 / ap, -fmax
/ (ap ** 2)])) ** 2)

er =10**6 * (s * math.pi ** (1 / 2)) / (2 * ap ** (1 / 2)) #[GPa]

der_s = (math.pi ** (1 / 2)) / (2 * ap ** (1 / 2))
der_ap = (-s * math.pi ** (1 / 2)) / (2 * ap ** (3 / 2))

u_er_sq =10**12 * np.sum(np.array([u_s ** 2, u_ap_sq]) * np.array([der_s, der_ap])**2)

eit = (1 - nus ** 2) / (1 / er - (1 - nui ** 2) / ei) #[GPa]

dnus = 0.01 * np.sqrt(u_nus_sq)
dnui = 0.01 * np.sqrt(u_nui_sq)
der = 0.01 * np.sqrt(u_er_sq)
dei = 0.01 * np.sqrt(u_ei_sq)

deit_nus = ((1 - (nus + dnus) ** 2) / (1 / er - (1 - nui ** 2) / ei) - eit) / dnus
deit_nui = ((1 - nus ** 2) / (1 / er - (1 - (nui + dnui) ** 2) / ei) - eit) / dnui
deit_er = ((1 - nus ** 2) / (1 / (er + der) - (1 - nui ** 2) / ei) - eit) / der
deit_ei = ((1 - nus ** 2) / (1 / er - (1 - nui ** 2) / (ei + dei)) - eit) / dei

u_eit_sq = np.sum(np.array([u_nus_sq, u_er_sq, u_nui_sq, u_ei_sq]) *
np.array([deit_nus, deit_er, deit_nui, deit_ei])) ** 2)

R = fmax / s ** 2 #nm/mN
u_R_sq = np.sum(np.array([u_fmax ** 2, u_s ** 2]) * np.array([1 / s ** 2, -2 * fmax /
s ** 3])) ** 2)

return hit, u_hit_sq, eit, u_eit_sq, sm, u_sm_sq, s, u_s_sq, R, u_R_sq

def riproducibilità(lista_file, file_path_indentatore): #Prende i dati dai file di
training e calcola i vari parametri(EIT, HIT, Sm, ecc..) per ogni curva, per calcolare la
riproducibilità

hit_ripr = np.array([])
eit_ripr = np.array([])
sm_ripr = np.array([])
s_ripr = np.array([])
R_ripr = np.array([])

```

```

r2_ripr = np.array([])

for file_path in lista_file:

    h1, f1, h_man_1, f_man_1, t_man_1, h2, f2, h_man_2, f_man_2, t_man_2 =
        divisione_mantenimento([file_path]) #divisione dei dati
    # h [nm], f [mN], t[s]

    #FASE DI CARICO
    #scarto tipo sui dati sperimentali, dovuti principalmente al contributo di
        accuratezza
    u_h1= 0.0006 * h1
    u_f1= 0.00058 * f1

    #regressione preliminare per determinare i parametri di partenza dell'ODR
    limiti_parametri= ([0 , -np.inf , 1],[np.inf , np.inf , 2])
    beta0_carico, matrice_coeff0_carico = curve_fit(funz_interp, h1, f1,
        bounds=limiti_parametri, method= 'dogbox')

    #regressione dei dati sperimentali con ODR
    coeff1, mat_cov1, delta1 = regressione_odr(h1, f1, u_h1, u_f1, beta0_carico,
        funz_interp_ODR_esp)

    #calcolo incertezza estesa
    u_sq_f1, inc_f1= calcolo_incertezza(h1, f1, coeff1, mat_cov1, delta1)

    #FASE DI SCARICO
    u_h2= 0.0006 * h2
    u_f2= 0.00058 * f2

    #regressione preliminare per determinare i parametri di partenza dell'ODR beta0
    limiti_parametri= ([0 , -np.inf , 1],[np.inf , np.inf , 2])
    beta0_scarico, matrice_coeff0_scarico = curve_fit(funz_interp, h2, f2,
        bounds=limiti_parametri, method= 'dogbox')

    #regressione dei dati sperimentali con ODR
    coeff2, mat_cov2, delta2= regressione_odr(h2,f2, u_h2, u_f2, beta0_scarico,
        funz_interp_ODR_esp)
    u_sq_f2, inc_f2 = calcolo_incertezza(h2, f2, coeff2, mat_cov2, delta2)
    h2_ord, f2_ord, inc_f2_ord, u_sq_f2_ord, delta2_ord= ordina_vettori(h2, f2,
        inc_f2, u_sq_f2, delta2)

    hit, u_hit_sq, eit, u_eit_sq, sm, u_sm_sq, s, u_s_sq, R, u_R_sq= analisi_new(h2_ord,
    f2_ord, coeff1, mat_cov1, u_sq_f2_ord, delta2_ord, file_path_indentatore)

    coeff_man_2, mat_cov_man_2 = curve_fit(funz_mant, t_man_2, h_man_2)
    _, r2 = coeff_man_2
    _, u_sq_r2= np.diag(mat_cov_man_2)

    hit_ripr= np.append(hit_ripr, hit)
    eit_ripr= np.append(eit_ripr, eit)
    sm_ripr= np.append(sm_ripr, sm)
    s_ripr= np.append(s_ripr, s)
    R_ripr= np.append(R_ripr, R)
    r2_ripr= np.append(r2_ripr, r2)

    u_ripr_hit = np.var(hit_ripr)
    u_ripr_eit = np.var(eit_ripr)
    u_ripr_sm = np.var(sm_ripr)
    u_ripr_s = np.var(s_ripr)
    u_ripr_R = np.var(R_ripr)
    u_ripr_r2 = np.var(r2_ripr)

    return u_ripr_hit, u_ripr_eit, u_ripr_sm, u_ripr_s, u_ripr_R, u_ripr_r2

def salva_matlab(coeff1, mat_cov1, coeff2, mat_cov2, r2, u_sq_r2, sm, u_sm_sq, s, u_s_sq,
hit, u_hit_sq, eit, u_eit_sq, R, u_R_sq,nome_file): #ATTENZIONE! Il nome del file deve
essere già in formato.mat
    dati_da_salvare= {

```

```

    'Vett_coeff_carico' : coeff1,
    'Mat_cov_carico' : mat_cov1,
    'Vett_coeff_scarico' : coeff2,
    'Mat_cov_scarico' : mat_cov2,
    'slope2' : r2,
    'var_slope2' : u_sq_r2,
    'sm' : sm,
    'var_sm' : u_sm_sq,
    's' : s,
    'var_s' : u_s_sq,
    'hit' : hit,
    'var_hit' : u_hit_sq,
    'eit' : eit,
    'var_eit' : u_eit_sq,
    'r' : R,
    'var_r' : u_R_sq,
}
savemat(nome_file, dati_da_salvare)

print("Dati salvati")
print()

def main():

    lista_file = [r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_1.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_2.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_3.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_4.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_5.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_6.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_7.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_8.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_9.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_10.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_12.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_11.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_13.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_14.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_15.TXT",
                  r"C:\Users\Ila\Desktop\codici\indentazione se holding\BCA10_16.TXT"] #
    Aggiungi qui i tuoi file, metto r davanti per far leggere il nome con caratteri speciali

    print()
    print("DATI TRAINING:")
    print()
    h1_ord, f1_ord, inc_f1_ord, u_sq_f1_ord, delta1_ord, coeff1, mat_cov1, h_man_1, f_man_1,
    t_man_1, h2_ord, f2_ord, inc_f2_ord, u_sq_f2_ord, delta2_ord, coeff2, mat_cov2, h_man_2,
    f_man_2, t_man_2 = divisione_e_regressione(lista_file)

    file_path_indentatore = r"C:\Users\Ila\Desktop\codici\indentazione_se
    holding\Berkovich BC-A10 10052024.tsv"
    hit, u_hit_sq, eit, u_eit_sq, sm, u_sm_sq, s, u_s_sq, R, u_R_sq =
    analisi_training(h2_ord, coeff1, coeff2, mat_cov1, mat_cov2, u_sq_f2_ord,
    delta2_ord, file_path_indentatore)

    print("HIT =", hit, "GPa +- ", np.sqrt(u_hit_sq), "*k GPa")
    print("EIT =", eit, "GPa +- ", np.sqrt(u_eit_sq), "*k GPa")
    print("Sm =", sm, "mN/nm +- ", np.sqrt(u_sm_sq), "*k mN/nm")
    print("S =", s, "mN/nm +- ", np.sqrt(u_s_sq), "*k mN/nm")
    print("R =", R, "nm/mN +- ", np.sqrt(u_R_sq), "*k nm/mN")

    print()

    # Calcolo coefficiente angolare Mantenimento
    coeff_man_1, mat_cov_man_1 = curve_fit(funz_mant, t_man_1, h_man_1)
    _, r1 = coeff_man_1
    _, u_sq_r1 = np.diag(mat_cov_man_1)

    coeff_man_2, mat_cov_man_2 = curve_fit(funz_mant, t_man_2, h_man_2)
    _, r2 = coeff_man_2

```



```

_ , u_sq_r2= np.diag(mat_cov_man_2)

u_ripr_hit_sq, u_ripr_eit_sq, u_ripr_sm_sq, u_ripr_s_sq, u_ripr_R_sq, u_ripr_r2_sq=
    riproducibilita(lista_file, file_path_indentatore)

#####
print()
print("CURVA n+1:")
print()

file_new=[r"C:\Users\Ila\Desktop\codici\indentazione se holding\n+1
\BC_A10_test_VIDIT_16.TXT"]

h1_ord_new, f1_ord_new, inc_f1_ord_new, u_sq_f1_ord_new, delta1_ord_new, coeff1_new,
    mat_cov1_new, h_man_1_new, f_man_1_new, t_man_1_new, h2_ord_new, f2_ord_new,
    inc_f2_ord_new, u_sq_f2_ord_new, delta2_ord_new, coeff2_new, mat_cov2_new,
    h_man_2_new, f_man_2_new, t_man_2_new = divisione_e_regressione(file_new)

file_path_indentatore = r"C:\Users\Ila\Desktop\codici\indentazione se holding
\Berkovich BC-A10 10052024.tsv"
hit_new, var_hit_new, eit_new, var_eit_new, sm_new, var_sm_new, s_new, var_s_new,
    R_new, var_R_new= analisi_new(h2_ord_new, f2_ord_new, coeff1_new, mat_cov1_new,
    u_sq_f2_ord_new, delta2_ord_new, file_path_indentatore)

u_hit_sq_new= var_hit_new + u_ripr_hit_sq *(hit_new/hit)**2
u_eit_sq_new= var_eit_new + u_ripr_eit_sq *(eit_new/eit)**2
u_sm_sq_new= var_sm_new + u_ripr_sm_sq *(sm_new/sm)**2
u_s_sq_new= var_s_new + u_ripr_s_sq *(s_new/s)**2
u_R_sq_new= var_R_new + u_ripr_R_sq *(R_new/R)**2

print("HIT =", hit_new, "GPa +-", np.sqrt(u_hit_sq_new), "*k GPa")
print("EIT =", eit_new, "GPa +-", np.sqrt(u_eit_sq_new), "*k GPa")
print("Sm =", sm_new, "mN/nm +-", np.sqrt(u_sm_sq_new), "*k mN/nm")
print("S =", s_new, "mN/nm +-", np.sqrt(u_s_sq_new), "*k mN/nm")
print("R =", R_new, "nm/mN +-", np.sqrt(u_R_sq_new), "*k nm/mN")
print()

# Calcolo coefficiente angolare Mantenimento
coeff_man_1_new, mat_cov_man_1_new = curve_fit(funz_mant, t_man_1_new, h_man_1_new)
_ , r1_new = coeff_man_1_new
_ , u_sq_r1_new= np.diag(mat_cov_man_1_new)

coeff_man_2_new, mat_cov_man_2_new = curve_fit(funz_mant, t_man_2_new, h_man_2_new)
_ , r2_new = coeff_man_2_new
_ , var_r2_new= np.diag(mat_cov_man_2_new)
u_sq_r2_new= var_r2_new + u_ripr_r2_sq *(r2_new/r2)**2

#Controllo h_max > h0_mant
h_max_new = h2_ord_new[-1]
h0_mant_new = h1_ord_new[-1]
u_sq_h_max_new = delta2_ord_new[-1]**2 /3
u_sq_h0_mant_new = delta1_ord_new[-1]**2 /3
dh_mant_new = h_max_new - h0_mant_new
u_dh_new= np.sqrt(u_sq_h_max_new + u_sq_h0_mant_new)
t1_new = dh_mant_new / u_dh_new
dof_test_1_new= 100
liv_fiducial_new= 0.95
t_rif1_new= t.ppf(1 - liv_fiducial_new, dof_test_1_new)

if(t1_new >= t_rif1_new):
    print("Test h_max > h0_mant: NON SI PUO' RIFIUTARE L'IPOTESI NULLA ")
    print()

else:
    print("Test hmax > h0_mant: SI RIFIUTA L'IPOTESI NULLA")
    print()

#Controllo Slope 2

```

```

t_slope2_new= r2_new / np.sqrt(u_sq_r2_new)
dof_test_2_new= len(h_man_2_new)-2
liv_fiducia2_new= 0.95
t_rif2_new= t.ppf(1-(1-liv_fiducia2_new)/2, dof_test_2_new)

if(t_slope2_new > -t_rif2_new and t_slope2_new < t_rif2_new):
    print("Test Slope2 = 0: NON SI PUO' RIFIUTARE L'IPOTESI NULLA")
    print()

elif( t_slope2_new <= -t_rif2_new):
    print("Test Slope2 = 0: SI RIFIUTA L'IPOTESI NULLA")
    print( "Slope2 < 0")
    print()

elif( t_slope2_new >= t_rif2_new):
    print("Test Slope2 = 0: SI RIFIUTA L'IPOTESI NULLA")
    print( "Slope2 > 0")
    print()

#Controllo Sm
t_sm = (sm_new - sm)/np.sqrt(u_sm_sq_new + u_sm_sq)
dof_test_sm = 100
liv_fiducia_sm= 0.95
t_rif_sm= t.ppf(1-(1-liv_fiducia_sm)/2, dof_test_sm)

if(t_sm >= -t_rif_sm and t_sm <= t_rif_sm):
    print("Test su Sm: NON SI PUO' RIFIUTARE L'IPOTESI NULLA")
    print()
else:
    print("Test su Sm: SI RIFIUTA L'IPOTESI NULLA")
    print()

#Controllo S
t_s = (s_new - s)/np.sqrt(u_s_sq_new + u_s_sq)
dof_test_s = 100
liv_fiducia_s= 0.95
t_rif_s= t.ppf(1-(1-liv_fiducia_s)/2, dof_test_s)

if(t_s >= -t_rif_s and t_s <= t_rif_s):
    print("Test su S: NON SI PUO' RIFIUTARE L'IPOTESI NULLA")
    print()
else:
    print("Test su S: SI RIFIUTA L'IPOTESI NULLA")
    print()

#Controllo EIT
t_eit = (eit_new - eit)/np.sqrt(u_eit_sq_new + u_eit_sq)
dof_test_eit = 100
liv_fiducia_eit= 0.95
t_rif_eit= t.ppf(1-(1-liv_fiducia_eit)/2, dof_test_eit)

if(t_eit >= -t_rif_eit and t_eit <= t_rif_eit):
    print("Test su EIT: NON SI PUO' RIFIUTARE L'IPOTESI NULLA")
    print()
else:
    print("Test su EIT: SI RIFIUTA L'IPOTESI NULLA")
    print()

#Controllo HIT
t_hit = (hit_new - hit)/np.sqrt(u_hit_sq_new + u_hit_sq)
dof_test_hit = 100
liv_fiducia_hit= 0.95
t_rif_hit= t.ppf(1-(1-liv_fiducia_hit)/2, dof_test_hit)

if(t_hit >= -t_rif_hit and t_hit <= t_rif_hit):
    print("Test su HIT: NON SI PUO' RIFIUTARE L'IPOTESI NULLA")
    print()
else:
    print("Test su HIT: SI RIFIUTA L'IPOTESI NULLA")
    print()

```

```

#Controllo R
t_R = (R_new - R)/np.sqrt(u_R_sq_new + u_R_sq)
dof_test_R = 100
liv_fiducia_R= 0.95
t_rif_R= t.ppf(1-(1-liv_fiducia_R)/2, dof_test_R)

if(t_R >= -t_rif_R and t_R <= t_rif_R):
    print("Test su R: NON SI PUO' RIFIUTARE L'IPOTESI NULLA")
    print()
else:
    print("Test su R: SI RIFIUTA L'IPOTESI NULLA")
    print()

##### SALVATAGGIO DATI (formato Matlab) #####
#Salva i dati del training nella stessa cartella dei dati di partenza:
y_n_file= input("Salva i dati di training? (y/n) ")

if y_n_file== 'y':
    # Ottenere solo il nome del file senza estensione
    base_name = os.path.basename(lista_file[0])
    file_name, _ = os.path.splitext(base_name)

    # Rimuovere l'underscore e il numero alla fine del nome del file
    modified_file_name = file_name[:-2]

    # Aggiungere _training.mat al nome del file modificato
    nome_file = modified_file_name + '_training.mat'

    # Creare il nuovo percorso del file
    nuovo_file_path = os.path.join(os.path.dirname(lista_file[0]), nome_file)

    salva_matlab(coeff1, mat_cov1, coeff2, mat_cov2, r2, u_sq_r2, sm, u_sm_sq, s,
        u_s_sq, hit, u_hit_sq, eit, u_eit_sq, R, u_R_sq, nuovo_file_path)

y_n_file_new= input("Salva i dati di test? (y/n) ")

if y_n_file_new== 'y':
    # Ottenere solo il nome del file senza estensione
    base_name_new = os.path.basename(file_new[0])
    file_name_new, _ = os.path.splitext(base_name_new)

    # Aggiungere _training.mat al nome del file modificato
    nome_file_new = file_name_new + '.mat'

    # Creare il nuovo percorso del file
    nuovo_file_path_test = os.path.join(os.path.dirname(file_new[0]), nome_file_new)

    salva_matlab(coeff1, mat_cov1, coeff2, mat_cov2, r2, u_sq_r2, sm, u_sm_sq, s,
        u_s_sq, hit, u_hit_sq, eit, u_eit_sq, R, u_R_sq, nuovo_file_path_test)

if __name__ == "__main__":
    main()

```