

UNIVERSITÀ DEGLI STUDI DI PADOVA  
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE  
Laurea magistrale in ingegneria aerospaziale

**Studio e caratterizzazione della delaminazione in  
strutture in materiale composito tramite reti  
neurali**

Relatore:  
Prof. **Mirco Zaccariotto**

Candidato:  
**Enrico Aloisi**  
Matricola:  
**2058265**

Anno accademico 2022-2023

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
<b>2</b>	<b>Materiali Compositi</b>	<b>7</b>
2.1	Proprietà generali . . . . .	7
2.1.1	Formulazione meccanica per una lamina . . . . .	9
2.1.2	Laminati . . . . .	13
2.2	Modalità di fallimento . . . . .	15
2.2.1	Delaminazione . . . . .	18
2.3	Modellazione FEM . . . . .	19
2.3.1	Layered shell . . . . .	19
2.3.2	Stacked shell . . . . .	20
2.3.3	Ply splitting . . . . .	21
<b>3</b>	<b>Structural Health Monitoring</b>	<b>22</b>
3.1	Motivazioni . . . . .	22
3.2	Metodi attivi . . . . .	25
3.3	Metodi passivi . . . . .	26
<b>4</b>	<b>Machine learning</b>	<b>29</b>
4.1	Reti neurali . . . . .	29
4.2	Funzionamento di base . . . . .	31
4.3	Reti convoluzionali . . . . .	34

4.4	Addestramento	37
4.4.1	Gestione dei dati	37
4.4.2	Overfitting	38
4.5	Reti neurali in SHM	39
<b>5</b>	<b>Modello FEM</b>	<b>42</b>
5.1	Modello di verifica	43
5.1.1	Geometria	43
5.1.2	Materiale	43
5.1.3	Interfaccia	46
5.1.4	Criterio di rottura	49
5.1.5	Mesh	51
5.1.6	Steps	51
5.1.7	Risultati	53
5.2	Modello di lavoro	57
5.2.1	Sensori	58
5.2.2	Variabili considerate nelle simulazioni	61
5.2.3	Geometria	61
5.2.4	Materiale	63
5.2.5	Mesh	63
5.2.6	Steps	63
5.2.7	Risultati	65
<b>6</b>	<b>Risultati e discussione</b>	<b>71</b>
6.1	Architettura della rete	72
6.1.1	Reti dense	73
6.1.2	Reti convoluzionali pure	74
6.1.3	Rete convoluzionale completa	77
6.2	Impostazioni avanzate	81
6.2.1	Funzioni di attivazione	83

6.2.2 Dropout . . . . .	84
6.3 Input . . . . .	85
6.3.1 Sensori monoassiali . . . . .	85
6.3.2 Spostamenti . . . . .	87
6.4 Limitazioni e sviluppi futuri . . . . .	89
<b>7 Conclusioni</b>	<b>95</b>

# Capitolo 1

## Introduzione

Nel vasto panorama dell'industria aerospaziale, i materiali compositi hanno assunto un ruolo di primaria importanza, divenendo un elemento chiave nella progettazione e nella costruzione di nuovi velivoli. Tra le diverse tipologie di materiali compositi, i laminati in CFRP (carbon fibre reinforced plastic) hanno conquistato una posizione di spicco. Questi compositi laminati, grazie alle loro eccezionali proprietà in rapporto alla massa, hanno rivoluzionato il settore, offrendo vantaggi significativi rispetto ai tradizionali materiali metallici. Tuttavia, emerge un aspetto cruciale: la suscettibilità a specifici tipi di danno (in questa tesi sarà studiata la delaminazione) non facilmente individuabili tramite ispezioni visive convenzionali. In questo contesto, nasce la necessità dello Structural Health Monitoring (SHM): un insieme di tecniche che mirano a risolvere questo problema attraverso un monitoraggio continuo o periodico dello stato di salute della struttura, che permette una rilevazione precoce dei potenziali danni, inclusa la delaminazione. L'introduzione di queste tecniche di monitoraggio non solo aumenta la sicurezza e l'affidabilità delle strutture aerospaziali, ma contribuisce anche a ottimizzare le operazioni di manutenzione, riducendo i costi e prolungando la vita utile dei velivoli. In questo senso, il connubio tra materiali compositi avanzati e tecnologie di monitoraggio strutturale rappresenta un passo fondamentale verso l'evoluzione e l'innovazione nell'ambito dell'ingegneria aerospaziale.

Queste tecniche sono ancora in fase di sperimentazione, ma i risultati ottenuti dai

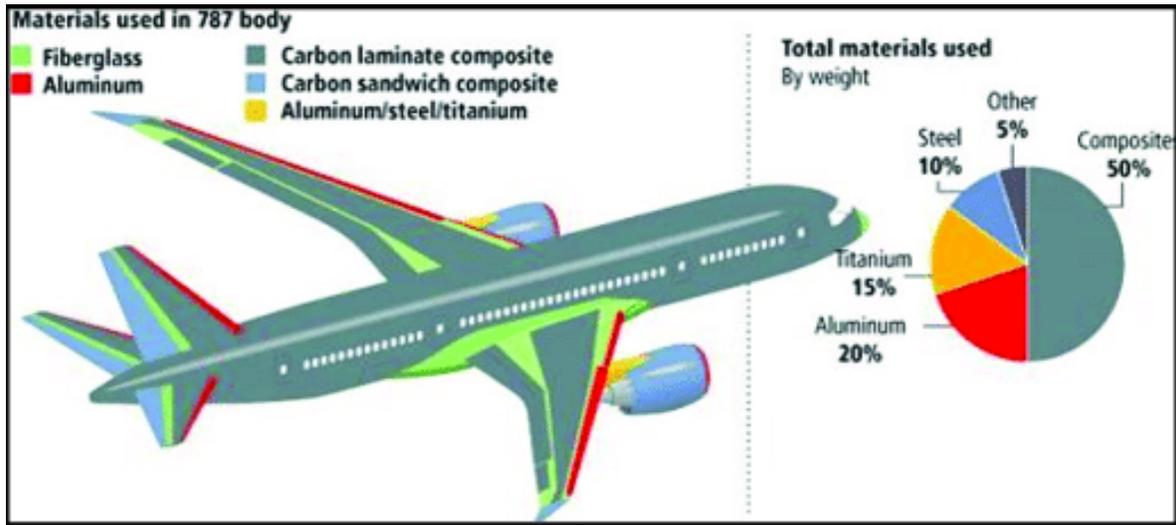


Figura 1.1: Il Boeing 787 è costituito per metà da compositi a base di fibra di carbonio (immagine tratta da [33])

ricercatori sono promettenti, ed è facile pensare che in pochi anni saranno implementati a livello commerciale. Questa tesi indaga sulla possibilità di rilevare una delaminazione nel momento in cui si forma (a seguito di un impatto a bassa velocità) attraverso una rete neurale che analizza i dati di un array di sensori posizionato sulla struttura. Questo approccio rappresenta una forma di Structural Health Monitoring (SHM) passivo, poiché non richiede l'uso di attuatori, ma si basa sulla capacità di rilevare le onde elastiche provocate da eventi esterni (in particolare impatti), e dovrebbe agire continuamente per riconoscere gli eventi pericolosi per l'integrità strutturale.

A questo scopo è stato costruito un modello FEM per verificare le proprietà del materiale usate in seguito, e successivamente un secondo modello per simulare rapidamente un grande numero di impatti. Caratteristica particolare delle reti neurali è infatti la necessità di una grande mole di dati per l'addestramento, e la simulazione FEM di un impatto spesso richiede molto tempo. È quindi necessario (almeno per gli studi preliminari come questo) che il modello utilizzato possa essere simulato in tempi brevi. I dati così ottenuti sono serviti per studiare l'architettura ottimale di una rete neurale,

in particolare è emerso che le reti convoluzionali sono molto efficaci nell'analisi di dati di questi tipo. Sono stati utilizzati due approcci, che si sono tradotti nella definizione di due problemi di diverso tipo per la rete neurale. Il problema di regressione consiste nella previsione di un singolo dato numerico (in questo caso l'area delaminata), mentre il problema di classificazione prevede di suddividere i dati in varie classi (in questo caso 5, in base all'entità del danno subito dalla piastra), e allenare la rete perché assegni ogni caso alla classe corretta.

I risultati ottenuti dimostrano che le reti neurali sono efficaci in entrambi questi problemi e, se è possibile procurarsi il grande numero di dati necessario ad addestrarle, possono essere utilizzate nei sistemi di SHM passivi.

# Capitolo 2

## Materiali Compositi

I compositi sono una categoria di materiali formati da almeno due diversi componenti, questo permette di ottenere proprietà fisiche diverse (e solitamente migliori) da quelle dei materiali di partenza. I componenti possono essere combinati in diversi modi (fibre lunghe, fibre corte, particelle...), che possono essere sfruttati per ottenere le proprietà più adatte a seconda dell'applicazione.

In campo aerospaziale la configurazione più comune è formata da lunghe fibre di un materiale ad alte prestazioni meccaniche (solitamente carbonio o vetro) inserite in una matrice polimerica. Le fibre forniscono la maggior parte della rigidità, mentre la matrice mantiene il materiale coeso e determina le proprietà nelle direzioni trasversali alle fibre.

### 2.1 Proprietà generali

L'indiscussa utilità dei materiali compositi nel settore aerospaziale è legata alle loro eccezionali proprietà di rigidità e resistenza specifiche. In un contesto in cui la riduzione della massa è cruciale per migliorare le prestazioni e l'efficienza dei velivoli, l'impiego di compositi consente di costruire strutture notevolmente più leggere rispetto a quelle realizzate con materiali metallici, mantenendo al contempo le proprietà meccaniche ri-

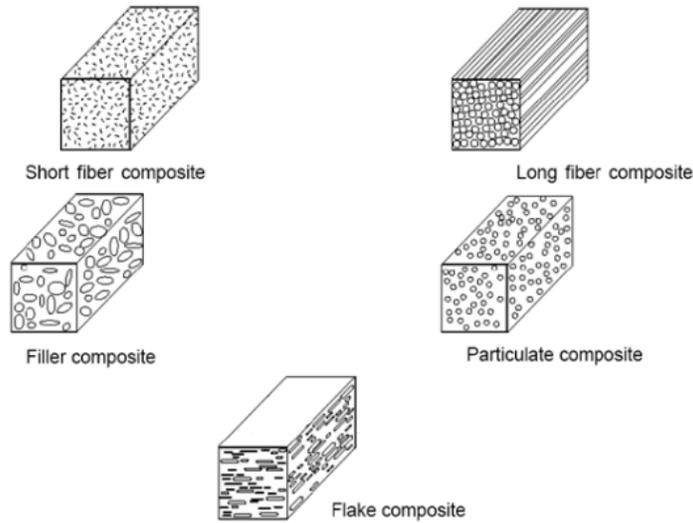


Figura 2.1: Varie configurazioni possibili di matrice e rinforzo in materiali compositi (immagine tratta da [50].)

chieste. Tuttavia, va sottolineato che l'ampio utilizzo dei materiali compositi è limitato da alcuni svantaggi, in particolare il loro elevato costo di produzione e lavorazione. Questi fattori influiscono sulla competitività economica dei compositi rispetto ai materiali tradizionali, come i metalli. Di conseguenza, la loro applicazione è spesso riservata a situazioni in cui i requisiti strutturali sono estremamente stringenti e il beneficio della riduzione di massa supera considerevolmente le spese aggiuntive associate.

Caratteristica peculiare dei materiali compositi a fibre è la loro anisotropia: la presenza delle fibre determina una direzione in cui il materiale è molto più rigido. Per ottenere delle proprietà simili in tutte le direzioni sono stati sviluppati i compositi a fibre intrecciate, tuttavia questa disposizione peggiora le caratteristiche di rigidità del materiale [47]. In seguito saranno trattati solo i compositi a fibre parallele, che sono utilizzati per le simulazioni svolte nei capitoli successivi.

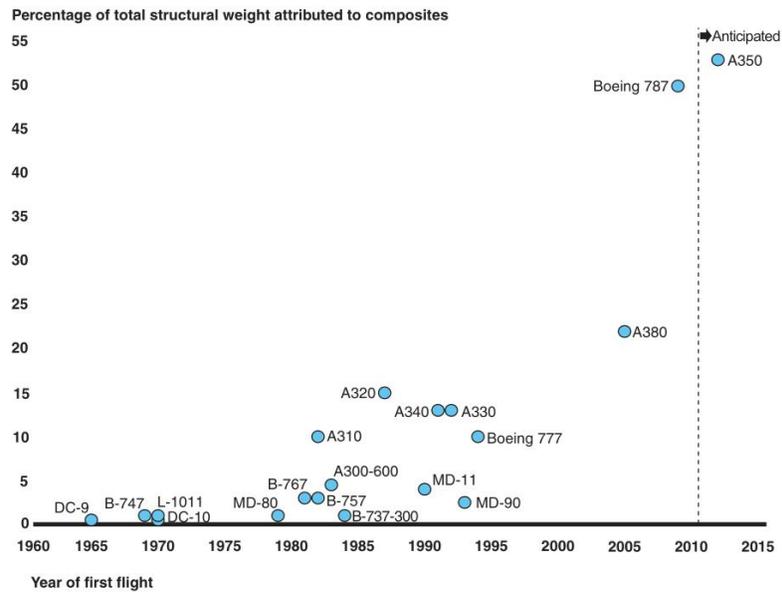


Figura 2.2: Frazione di materiali compositi negli aerei commerciali nel tempo (tratta da [20].)

### 2.1.1 Formulazione meccanica per una lamina

La legge di Hooke generalizzata, relazione tra deformazione ( $\boldsymbol{\varepsilon}$ ) e tensione ( $\boldsymbol{\sigma}$ ) nel materiale si può scrivere come

$$\boldsymbol{\varepsilon} = \mathbf{C}\boldsymbol{\sigma}$$

in alternativa esiste la relazione inversa

$$\boldsymbol{\sigma} = \mathbf{S}\boldsymbol{\varepsilon}$$

dove  $\mathbf{C}$  è la *compliance matrix* e  $\mathbf{S}$  la sua inversa, detta *stiffness matrix*. Queste matrici contengono le proprietà del materiale,  $\boldsymbol{\varepsilon}$  e  $\boldsymbol{\sigma}$  contengono rispettivamente tutti i componenti delle deformazioni e delle tensioni:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{bmatrix} \quad \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{bmatrix}$$

Per i materiali isotropi le proprietà sono le stesse in tutte le direzioni, quindi la *compliance matrix* si può scrivere:

$$\mathbf{C} = \begin{bmatrix} 1/E & -\nu/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & 1/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & -\nu/E & 1/E & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G \end{bmatrix}$$

Dove  $E$  (modulo di Young),  $G$  (modulo di taglio) e  $\nu$  (coefficiente di Poisson) non sono indipendenti, ma sono legate dalla relazione

$$E = 2G(1 + \nu)$$

che porta le variabili effettivamente indipendenti a 2

Per un materiale ortotropo (come una lamina in composito), che presenta cioè 3 piani di simmetria, le variabili indipendenti sono 9, e la matrice  $\mathbf{C}$  prende la forma:

$$\mathbf{C} = \begin{bmatrix} 1/E_1 & -\nu_{21}/E_2 & -\nu_{31}/E_3 & 0 & 0 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & -\nu_{32}/E_3 & 0 & 0 & 0 \\ -\nu_{13}/E_1 & -\nu_{23}/E_2 & 1/E_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{31} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{12} \end{bmatrix}$$

è importante notare che in questo caso i parametri E, G e  $\nu$  non sono legati, quindi le 9 variabili sono tutte indipendenti[18]. Inoltre  $\nu_{ij} \neq \nu_{ji}$  ma vale la relazione

$$\frac{\nu_{ij}}{E_{ii}} = \frac{\nu_{ji}}{E_{jj}}$$

che mantiene la simmetria della matrice

Nella maggior parte dei casi i compositi sono utilizzati in lastre sottili, è quindi utile studiare il caso in cui la sollecitazione fuori dal piano è trascurabile. Le uniche componenti di tensione non nulle sono  $\sigma_{11}$ ,  $\sigma_{22}$  e  $\tau_{12}$ , si può quindi ridurre la legge di Hooke generalizzata a una forma più semplice:

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{bmatrix} = \begin{bmatrix} 1/E_{11} & -\nu_{12}E_{22} & 0 \\ -\nu_{12}E_{22} & 1/E_{22} & 0 \\ 0 & 0 & 1/G_{12} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{bmatrix}$$

Quando si combinano più strati disposti in direzioni diverse si ottiene un composito laminato: le proprietà di questi materiali possono essere alterate modificando la *stacking sequence* per ottenere le caratteristiche desiderate. Per calcolare le proprietà di un laminato è necessario per prima cosa valutare quelle di una lamina orientata in direzione

arbitraria. Le relazioni trovate precedentemente sono valide solo se l'asse di simmetria (quello delle fibre) è uno degli assi di riferimento. In generale si considera che le fibre siano orientate ad un angolo  $\theta$  rispetto all'asse x. Si indica con  $\boldsymbol{\sigma}$  la tensione espressa nelle coordinate globali, e con  $\boldsymbol{\sigma}_l$  quella espressa nelle coordinate del materiale (in cui l'asse x coincide con la direzione delle fibre).

Nel sistema di riferimento del materiale vale (invertendo la relazione trovata in precedenza)

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{bmatrix} = \frac{1}{1 - \nu_{12}\nu_{21}} \begin{bmatrix} E_{11} & \nu_{12}E_{22} & 0 \\ \nu_{12}E_{22} & E_{22} & 0 \\ 0 & 0 & G_{12}(1 - \nu_{12}\nu_{21}) \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{bmatrix}$$

In forma breve:  $\boldsymbol{\sigma}_l = \mathbf{Q}\boldsymbol{\varepsilon}_l$

Da relazioni trigonometriche si può scrivere la relazione tra  $\boldsymbol{\sigma}$  e  $\boldsymbol{\sigma}_l$

$$\boldsymbol{\sigma}_l = \mathbf{T}\boldsymbol{\sigma}$$

dove

$$\mathbf{T} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & 2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -2\sin\theta\cos\theta \\ \sin\theta\cos\theta & -\sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix}$$

Inoltre per le deformazioni vale

$$\boldsymbol{\varepsilon}_l = \mathbf{RTR}^{-1}\boldsymbol{\varepsilon} = (\mathbf{T}^T)^{-1}\boldsymbol{\varepsilon}$$

dove  $\mathbf{R}$  è la matrice di Reuter:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

La relazione tra tensione e deformazione nel sistema di riferimento principale si esprime

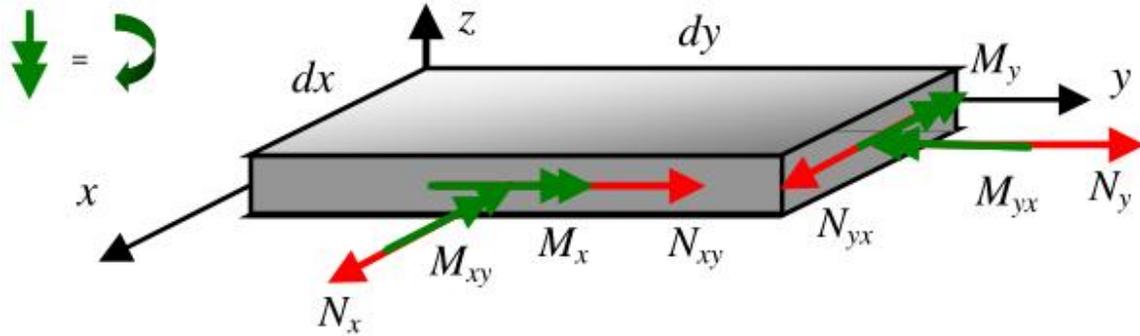


Figura 2.3: Schema delle forze agenti su un laminato (immagine presa da [47])

quindi come

$$\boldsymbol{\sigma} = \mathbf{T}^{-1} \boldsymbol{\sigma}_l = \mathbf{T}^{-1} \mathbf{Q} \boldsymbol{\varepsilon}_l = \mathbf{T}^{-1} \mathbf{Q} (\mathbf{T}^T)^{-1} \boldsymbol{\varepsilon} = \bar{\mathbf{Q}} \boldsymbol{\varepsilon}$$

La matrice  $\bar{\mathbf{Q}}$  è detta matrice di rigidità ridotta, ed esprime la relazione tra deformazione e tensione per una lamina ortotropa con le fibre orientate ad un angolo  $\theta$  rispetto all'asse  $x$  e in condizioni di tensione piana.

### 2.1.2 Laminati

Un laminato è formato da diversi strati ortotropi, le sue proprietà si possono trovare integrando quelle dei singoli strati. Le forze e i momenti agenti su un laminato sono solitamente espressi tramite il vettore  $\mathbf{N}$  delle forze e il vettore  $\mathbf{M}$  dei momenti:

$$\mathbf{N} = \begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix}$$

Questi si possono ottenere dall'integrazione delle tensioni sulla sezione:

$$\mathbf{N} = \int \boldsymbol{\sigma} dz \quad \mathbf{M} = \int \boldsymbol{\sigma} z dz$$

è inoltre utile esprimere le deformazioni come combinazione di deformazione piana ( $\boldsymbol{\varepsilon}_0$ ) e curvatura ( $\mathbf{k}$ ). Considerando nulle le deformazioni fuori dal piano, queste quantità sono legate al vettore delle deformazioni da:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_0 + z\mathbf{k}$$

Ne consegue che

$$\mathbf{N} = \int \boldsymbol{\sigma} dz = \int \bar{\mathbf{Q}}(\boldsymbol{\varepsilon}_0 + z\mathbf{k}) dz \quad \mathbf{M} = \int \boldsymbol{\sigma} z dz = \int \bar{\mathbf{Q}}(z\boldsymbol{\varepsilon}_0 + z^2\mathbf{k}) dz$$

Raccogliendo queste relazioni in forma matriciale si può scrivere

$$\begin{bmatrix} \mathbf{N} \\ \mathbf{M} \end{bmatrix} = \int \begin{bmatrix} \bar{\mathbf{Q}} & \bar{\mathbf{Q}}z \\ \bar{\mathbf{Q}}z & \bar{\mathbf{Q}}z^2 \end{bmatrix} dz \begin{bmatrix} \boldsymbol{\varepsilon}_0 \\ \mathbf{k} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}_0 \\ \mathbf{k} \end{bmatrix}$$

Le proprietà delle lamine (materiale e orientamento) sono contenute in  $\bar{\mathbf{Q}}$ , e sono costanti sulla lamina, che consente di scrivere le espressioni di  $\mathbf{A}$ ,  $\mathbf{B}$  e  $\mathbf{D}$  in forma di sommatoria sulle lamine.

$$\mathbf{A} = \sum_i \bar{\mathbf{Q}}_i t_i$$

$$\mathbf{B} = \sum_i \bar{\mathbf{Q}}_i t_i \bar{z}_i$$

$$\mathbf{D} = \sum_i \bar{\mathbf{Q}}_i t_i (\bar{z}_i^2 + \frac{1}{12} t_i^2)$$

dove  $t_i$  è lo spessore della  $i$ -esima lamina,  $\bar{\mathbf{Q}}_i$  è la sua matrice di rigidità ridotta e  $\bar{z}_i$  è la coordinata  $z$  del suo piano mediano. La relazione tra deformazione e tensione in un

laminato può essere scritta in forma completa come

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{21} & A_{22} & A_{26} & B_{21} & B_{22} & B_{26} \\ A_{61} & A_{62} & A_{66} & B_{61} & B_{62} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{21} & B_{22} & B_{26} & D_{21} & D_{22} & D_{26} \\ B_{61} & B_{62} & B_{66} & D_{61} & D_{62} & D_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{0x} \\ \varepsilon_{0y} \\ \varepsilon_{0xy} \\ k_x \\ k_y \\ k_{xy} \end{bmatrix}$$

In generale la matrice di rigidità complessiva non presenta zeri, questo porta ad accoppiamenti meccanici (solitamente indesiderati) tra le varie sollecitazioni a cui la piastra è sottoposta.

A seconda della *stacking sequence* possono essere cambiate le proprietà: ad esempio un laminato simmetrico (in cui l'orientamento delle lamine è simmetrico rispetto al piano mediano) non presenta accoppiamenti tra momenti e forze ( $\mathbf{B} = 0$ ). Analogamente un laminato bilanciato (in cui ogni lamina con orientamento  $\theta$  è bilanciata da una lamina con orientamento  $-\theta$ ) disaccoppia la tensione normale e il taglio ( $A_{16} = A_{26} = 0$ ).

## 2.2 Modalità di fallimento

Per calcolare la resistenza di una struttura in composito è fondamentale conoscere le sue possibili modalità di fallimento. A causa della loro anisotropia esso avviene in modo diverso rispetto a quanto accade per i materiali più comuni. Nei materiali polimerici il meccanismo di frattura a livello microscopico è diverso da quello tipico dei metalli, per questo motivo la resistenza a trazione e quella a compressione sono spesso diversi [47]. La presenza delle fibre inoltre causa una differenza tra il carico supportabile in direzione longitudinale e trasversale, è quindi necessario che il criterio di fallimento sia valutato nel sistema di riferimento della lamina.

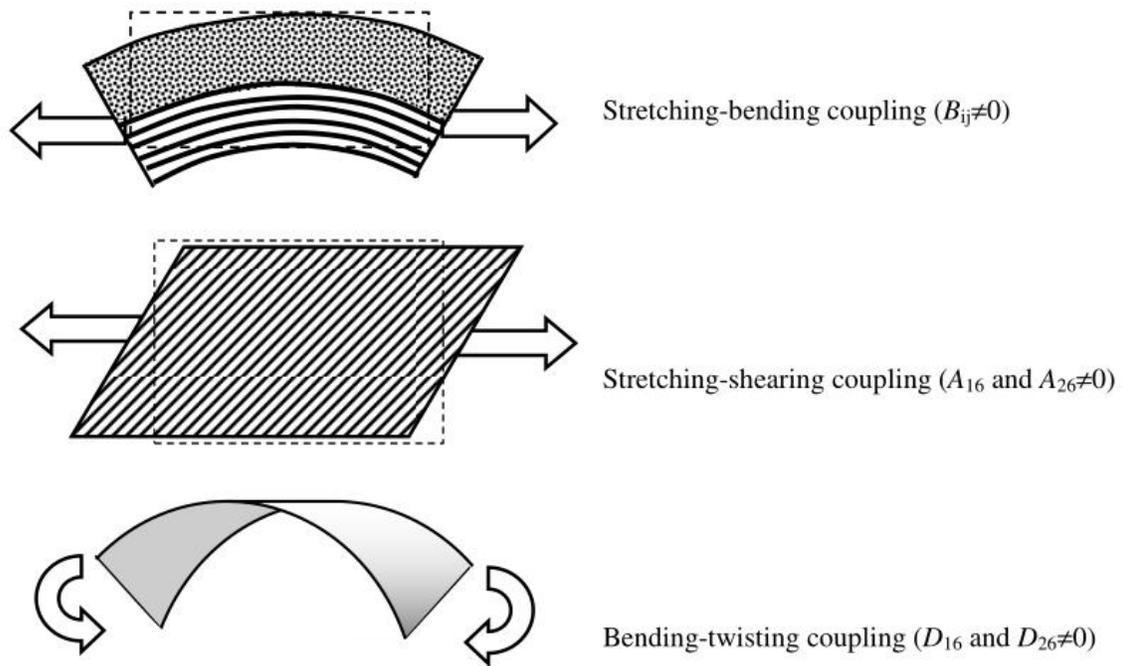


Figura 2.4: Rappresentazione degli accoppiamenti meccanici più comuni nei laminati (immagine presa da [47])

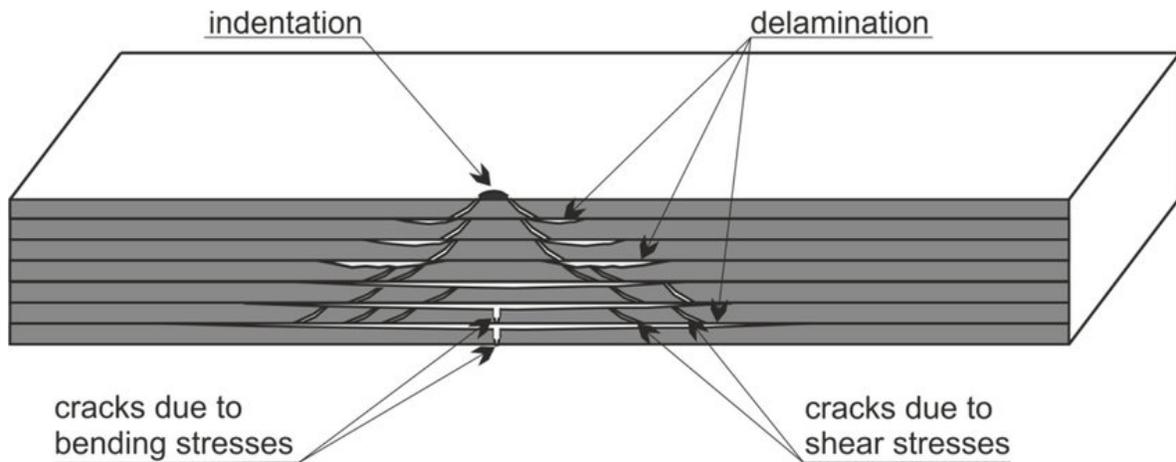


Figura 2.5: Possibili tipi di danno a seguito di un impatto in un composito laminato (immagine presa da [32])

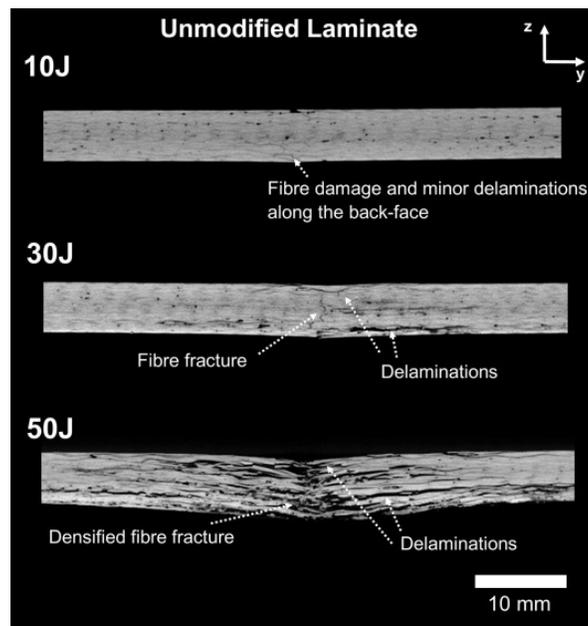


Figura 2.6: Foto di un laminato danneggiato da impatti a diversa energia (immagine tratta da [46])

Le singole lamine sono inoltre sottoposte a sollecitazioni diverse a seconda dell'orientamento, di conseguenza non raggiungono la tensione di fallimento contemporaneamente, ma in un ordine che dipende dall'orientamento e dalla posizione nello spessore.

Le strutture in composito possono subire buckling come quelle in materiale isotropo, tuttavia questo è più complicato da modellare a causa degli accoppiamenti meccanici presenti nella struttura [1], inoltre quando la lamina è sottoposta a compressione nella direzione del rinforzo, è anche possibile che si verifichi buckling nelle singole fibre [13].

### 2.2.1 Delaminazione

Un'altra modalità di fallimento tipica dei laminati (e quella su cui questa tesi si concentra) è la delaminazione, che si verifica quando gli strati che compongono il materiale si separano uno dall'altro. L'interfaccia tra due lamine è il punto più debole della struttura, dove la resistenza è minima, quindi è facile che sia la posizione dove il danno si innesca. Le cause della delaminazione possono essere molteplici, ma una delle principali sono gli impatti.

In particolare gli impatti a bassa velocità possono avvenire in qualsiasi momento (durante il volo, ma anche in fase di assemblaggio o di trasporto), e possono essere pericolosi nonostante la bassa energia coinvolta. La peculiarità della delaminazione è che il danno potrebbe essere invisibile all'occhio umano, poiché gli strati esterni del materiale non sono danneggiati, è quindi fondamentale sviluppare delle tecniche che permettano di controllare lo stato di salute del laminato. La delaminazione è infatti pericolosa proprio perchè potrebbe non essere rilevata, e un componente danneggiato potrebbe essere installato sul prodotto finito, compromettendo la sicurezza delle operazioni. Le delaminazioni quasi invisibili dovute a impatti a bassa velocità sono dette BVID (barely visible impact damage), e sono oggetto di studio da parte di diversi team di ricerca proprio a causa del rischio che comportano [35] [29].

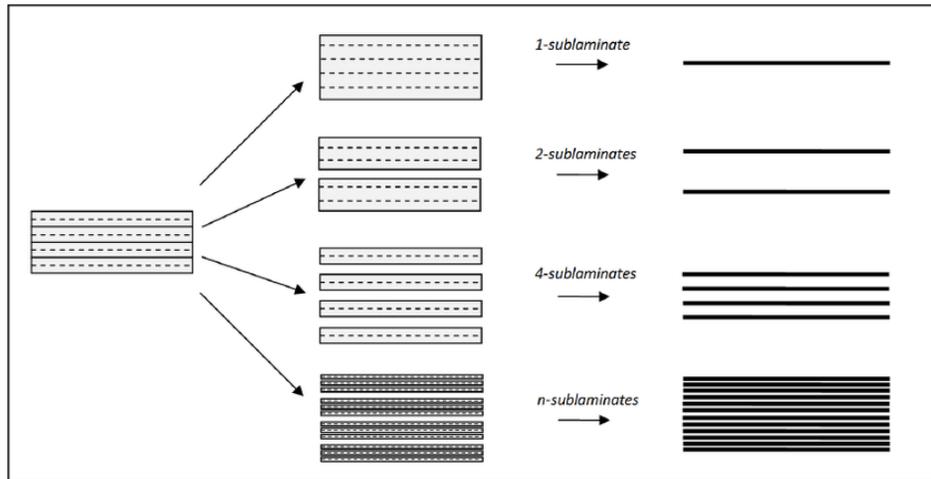


Figura 2.7: Diversi livelli di discretizzazione di un laminato composito per la simulazione FEM (immagine tratta da [30])

## 2.3 Modellazione FEM

I software di simulazione FEM (con particolare riferimento ad Abaqus, usato per questo lavoro), forniscono diverse tecniche per modellare i compositi laminati, a seconda del grado di dettaglio richiesto[26]. I metodi più comuni sono 3:

- *Layered shell*
- *Stacked shell*
- *Ply splitting*

### 2.3.1 Layered shell

l'approccio *layered shell* è quello più semplificato: la superficie viene modellata con elementi di tipo *shell* in modo che ci sia un solo elemento lungo lo spessore della piastra. Il solutore integra quindi le proprietà del materiale e trova le caratteristiche complessive del laminato, si possono ottenere i valori di tensione e deformazione in ogni lamina

utilizzando un numero sufficiente di punti di integrazione. Questo metodo riproduce bene il comportamento del laminato ad un costo computazionale ridotto, tuttavia è poco adatto a simulare il danneggiamento poiché la delaminazione non può avvenire quando è presente un solo elemento lungo lo spessore.

### 2.3.2 Stacked shell

Per ovviare a questo problema è stato sviluppato l'approccio *stacked shell*, in cui ogni lamina è modellata separatamente, di conseguenza lo spessore del laminato è rappresentato da un elemento per ogni strato. Questo aumenta considerevolmente il costo computazionale del modello, poiché il numero di elementi viene moltiplicato per il numero di strati presenti (solitamente nell'ordine delle decine).

Tra una lamina e la successiva è presente una zona coesiva, che può essere modellata in due modi: con elementi coesivi o con superfici coesive [52].

Gli elementi coesivi sono elementi tridimensionali che hanno la possibilità di rompersi: quando viene soddisfatto il criterio di rottura, la rigidità dell'elemento è gradualmente ridotta fino a diventare nulla, a questo punto l'elemento può essere eliminato dalla simulazione.

Le superfici coesive utilizzano un approccio molto diverso per ottenere dei risultati simili. Il comportamento coesivo non è modellato con l'inserimento di elementi appositi, ma tramite l'algoritmo di contatto. Due superfici vengono incollate fino al raggiungimento della soglia di fallimento, a quel punto la rigidità di legame viene ridotta (in modo analogo a quanto succede per gli elementi coesivi) fino a diventare nulla.

L'approccio *stacked shell* permette quindi di simulare la delaminazione della struttura, ma richiede la conoscenza delle proprietà coesive del materiale, che non sono spesso disponibili e richiedono una verifica sperimentale per fornire risultati accurati [14].

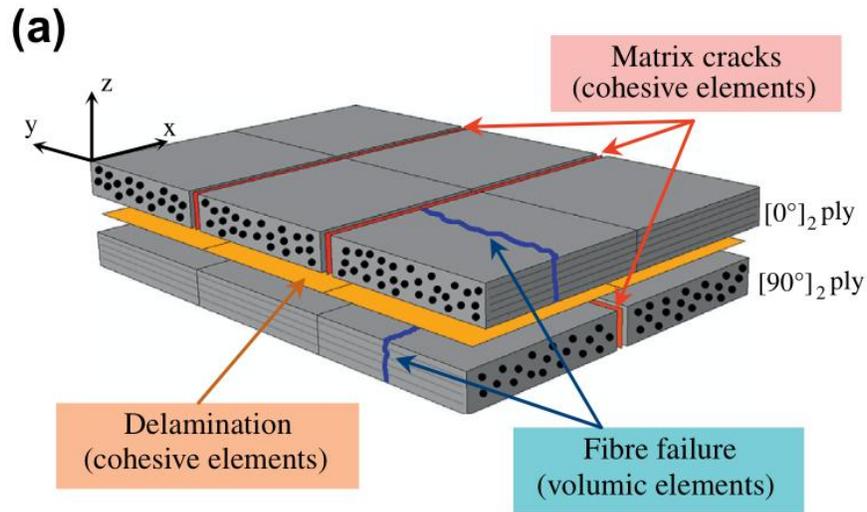


Figura 2.8: Implementazione FEM di un modello *ply splitting* nell'articolo [\[42\]](#)

### 2.3.3 Ply splitting

Per simulare la struttura in modo ancora più dettagliato è possibile sviluppare un terzo tipo di modello: il *ply splitting model*. Questo è molto simile al modello *stacked shell* presentato in precedenza, ma vengono aggiunte delle ulteriori zone coesive all'interno delle lamine, per simulare la rottura della matrice. Questo approccio ha un costo computazionale estremamente elevato, dato il grande numero di elementi o superfici coesive necessari.

# Capitolo 3

## Structural Health Monitoring

Per *structural health monitoring* (SHM) si intende un'insieme di tecniche che hanno l'obiettivo di controllare lo stato di salute di una struttura, rilevando quindi la presenza di danni (cricche, delaminazioni...). Esistono diversi approcci a questo problema, a seconda del campo di applicazione, ma hanno tutti in comune l'utilizzo di sensori posizionati sulla struttura in esame che permettono di rilevare il danno e determinarne posizione ed entità, con l'obiettivo di valutare la resistenza residua del componente [17].

### 3.1 Motivazioni

Nell'industria aeronautica, i materiali compositi hanno assunto un ruolo sempre più rilevante nel corso degli ultimi decenni, grazie alle loro ottime proprietà specifiche, particolarmente adatte a questo settore. Tuttavia, l'impiego diffuso di materiali compositi nell'aeronautica ha portato alla luce una sfida significativa: la gestione del danno. Il problema fondamentale è rappresentato dalla difficoltà nel rilevare le delaminazioni in questi materiali. La natura stratificata dei compositi rende arduo individuare difetti che potrebbero essere nascosti all'interno della struttura, non immediatamente visibili attraverso metodi di ispezione tradizionali.

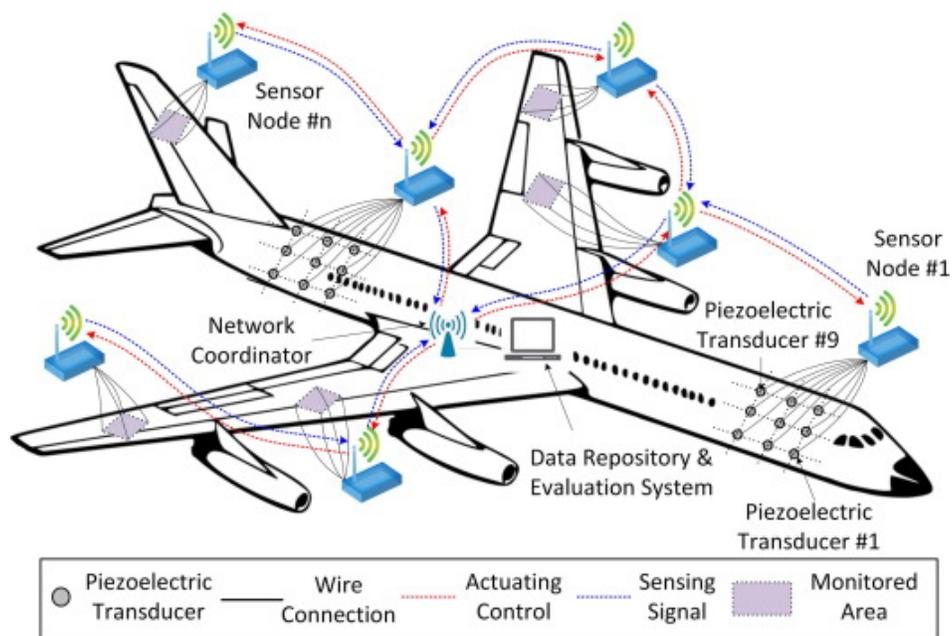


Figura 3.1: Schema di funzionamento di un sistema SHM applicato a un aereo di linea (immagine tratta da [21])

Storicamente la maggior parte dei componenti in metallo era progettata per non subire danno durante l'intera durata della vita operativa ed essere sostituita alla fine (*safe life*). Successivamente è stato sviluppato l'approccio *fail safe*: progettare il componente in modo che il danno non sia catastrofico, ma consenta l'operazione per un breve periodo prima della riparazione. L'avvento dei materiali compositi ha poi portato all'introduzione del concetto di *damage tolerance*: la struttura deve essere in grado di sostenere i carichi di lavoro per lunghi periodi anche in presenza di danno, parallelamente si effettuano controlli dell'integrità strutturale a periodi regolari.

Questo approccio necessita quindi di tecniche di ispezione non distruttive che permettano di valutare la presenza di un danno e la resistenza residua della struttura. Al momento questi controlli sono svolti principalmente tramite ispezione visiva, che però è molto lunga e costosa poiché, per un'ispezione particolarmente approfondita, richiede che il velivolo sia disassemblato per osservare con cura ogni componente.

Un sistema di structural health monitoring potrebbe rendere molto più rapidi questi passaggi, rilevando i danni nel momento in cui si verificano (con sistemi di monitoraggio continuo) o durante un'analisi periodica (che sarebbe automatica e non richiederebbe di disassemblare l'aeromobile). Un controllo molto frequente dell'integrità strutturale permetterebbe inoltre di ridurre la massa della struttura, che al momento deve essere sovradimensionata per soddisfare i criteri di *damage tolerance*. I sistemi di monitoraggio continuo sono spesso paragonati al sistema nervoso animale poiché, in modo analogo, raccolgono informazioni in tempo reale sulla salute di tutti i componenti, e la comunicano al centro di elaborazione dei dati.

I sensori utilizzabili a questo scopo sono di diversi tipi, è importante distinguere tra due diversi approcci: i sensori possono essere montati esternamente o integrati nella struttura. Il vantaggio di un sensore esterno è la facilità con cui può essere sostituito in caso di malfunzionamento, mentre per un sensore integrato è necessario cambiare l'intero pannello. Inoltre l'inserimento di sensori all'interno di un laminato introduce dei punti deboli nella struttura, che quindi deve essere rinforzata, aumentandone la massa. I sensori applicati esternamente, tuttavia, possono cambiare le caratteristiche della superficie, in particolare l'alterazione delle proprietà aerodinamiche di un velivolo

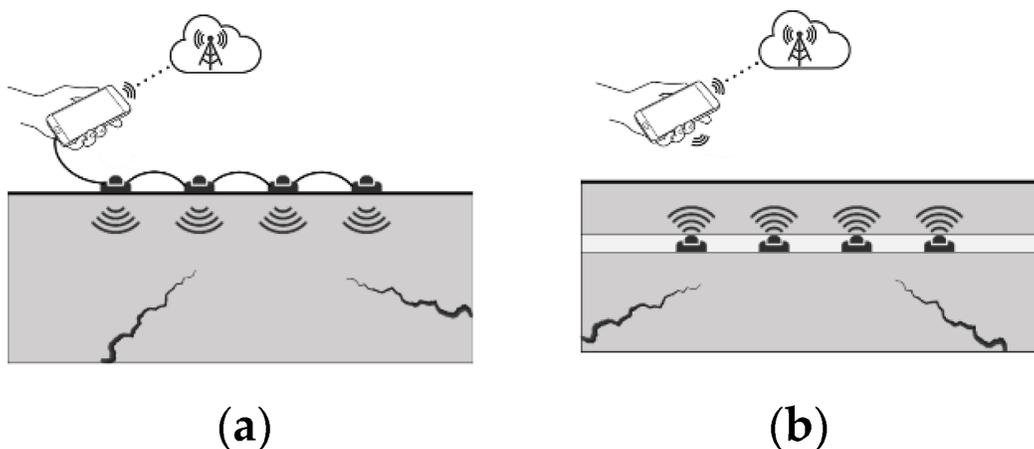


Figura 3.2: Rappresentazione di un sistema con sensori esterni (a) e uno con sensori integrati (b) tratta da [10]

potrebbe essere uno svantaggio troppo importante.

È possibile utilizzare diversi tipi di sensori per lo SHM: i più comuni sono gli accelerometri, che sono economici ed affidabili, ma in alcune applicazioni sono stati usati anche estensimetri o sensori in fibra ottica. La gestione di dati di diverso tipo richiede la scrittura di diversi algoritmi per l'analisi, per questo motivo le reti neurali sono particolarmente utili, dato che possono facilmente adattarsi ad input di natura diversa.

Le tecniche proposte per effettuare SHM sono molteplici, ma si possono dividere in due categorie principali:

- Metodi attivi (Il sistema sollecita la struttura e analizza la risposta)
- Metodi passivi (il sistema sfrutta le sollecitazioni esterne)

## 3.2 Metodi attivi

I sistemi di SHM attivo sono composti da un array di sensori e attuatori (sono molto usati i trasduttori piezoelettrici, che possono svolgere entrambe le funzioni sfruttando

l'effetto piezoelettrico diretto e inverso). Gli attuatori hanno la funzione di sollecitare la struttura generando delle onde elastiche, che vengono registrate dai sensori. Il vantaggio più importante di un approccio attivo è che l'onda è nota e controllabile, quindi è possibile scegliere la frequenza per eccitare un solo modo di vibrare e ottenere un rapporto segnale-rumore molto alto.

L'applicazione principale dei sistemi SHM attivi è nei controlli periodici da effettuare a velivolo fermo: le vibrazioni indotte dagli attuatori sono troppo deboli per essere rilevate durante il volo (a causa dei motori e delle forze aerodinamiche, che coprono il segnale), ma a terra è possibile usarli per rilevare difetti nella struttura.

Uno dei primi metodi sviluppati prevede di utilizzare a turno uno dei trasduttori come attuttore e gli altri come sensori: per ogni coppia di trasduttori viene confrontato il segnale trasmesso con lo stesso segnale misurato su un campione non danneggiato. Se l'onda elastica attraversa il danno, si registra una differenza nel contenuto in frequenza del segnale ricevuto, è quindi possibile stimare la posizione e la dimensione dell'area danneggiata valutando le coppie di sensori influenzate da essa. Utilizzando questo metodo, Liu et al. [16] sono riusciti nel 2012 a trovare una delaminazione con un raggio di confidenza  $< 2.2$  cm su diversi pannelli in composito di area  $35 \times 35$  cm<sup>2</sup> con 4 coppie di sensori.

### 3.3 Metodi passivi

In alternativa è possibile progettare il sistema di monitoring in modo che sia sempre operativo, e rilevi il danno nel momento in cui si forma. Queste tecniche sono dette passive poiché necessitano dei sensori ma non degli attuatori: la sollecitazione è fornita invece dagli eventi esterni (principalmente gli impatti). Le possibili cause del danno sono molteplici, e possono avvenire sia durante il volo (i casi più comuni sono gli urti con uccelli o grandine) che a terra (dove solitamente avvengono a bassa velocità, ad esempio per la caduta di un attrezzo durante l'assemblaggio o la manutenzione). Risulta quindi utile sviluppare un sistema di controllo che registra le sollecitazioni subite dalla struttura e determina se c'è il rischio di un danno (solitamente in forma di delaminazione).

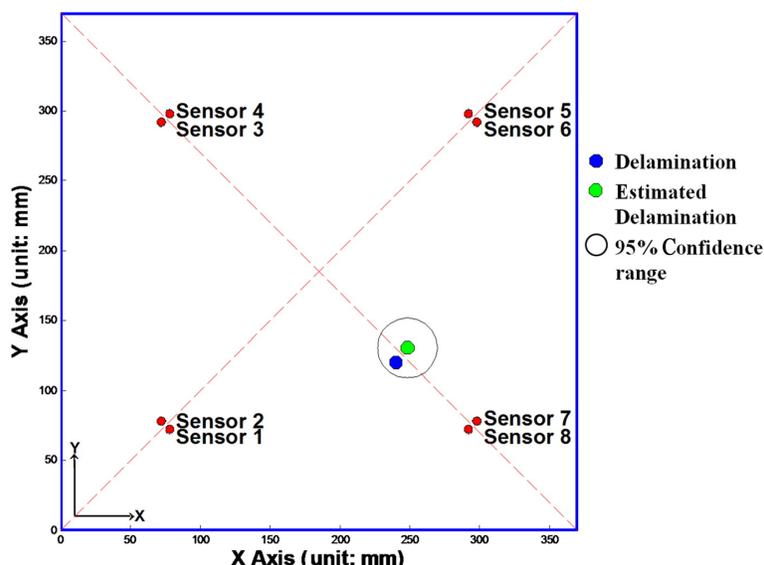


Figura 3.3: Stima della posizione del danno in [16]

L'onda elastica generata da un impatto non è, ovviamente, controllabile, quindi risulta più difficile estrarre le informazioni necessarie a determinare l'entità del danno. Per questo motivo i sistemi passivi sono solitamente meno affidabili e precisi di quelli attivi, tuttavia permettono di rilevare immediatamente l'impatto senza bisogno di condurre un'ispezione.

In un articolo del 2011 ad esempio, è stato proposto un algoritmo per calcolare la posizione di un impatto su una piastra in composito [3]. L'errore medio misurato è stato di 2.4 cm con 5 sensori di deformazione in fibra ottica, tuttavia l'errore massimo è significativamente più elevato (6.2 cm).

Data la difficoltà nello studio analitico di questo problema, negli ultimi anni sono stati spesso usati metodi di machine learning, che sono molto efficaci in situazioni di questo tipo, dove i metodi di calcolo espliciti sono relativamente poco affidabili. Le reti neurali infatti, hanno dimostrato ottime capacità di riconoscere pattern anche dove l'uomo fatica e, vista la relativa facilità con cui si possono costruire ad oggi, sono sicuramente un metodo da considerare per affrontare il problema.

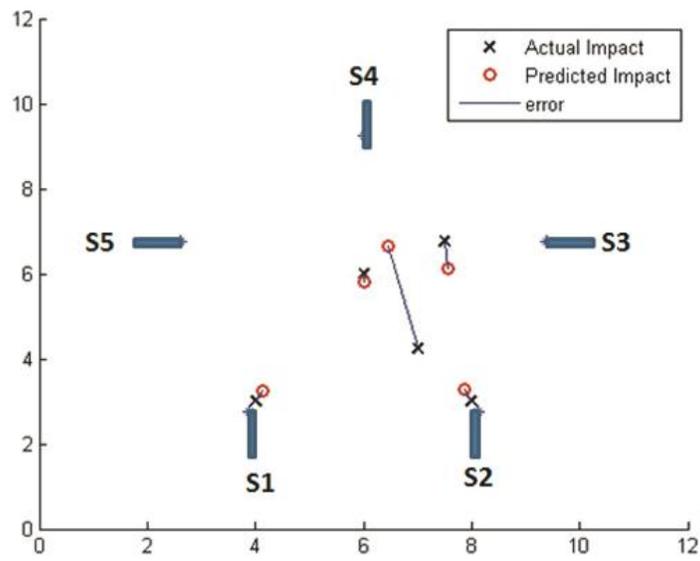


Figura 3.4: Prestazioni dell'algoritmo proposto in [3] nella determinazione della posizione di un impatto.

# Capitolo 4

## Machine learning

Il machine learning, o apprendimento automatico, rappresenta una disciplina fondamentale nell'ambito dell'intelligenza artificiale, in cui i computer acquisiscono la capacità di apprendere modelli e compiere decisioni senza essere esplicitamente programmati per compiti specifici. Questa metodologia si basa sull'analisi di dati e sul riconoscimento di pattern, consentendo alle macchine di migliorare le proprie prestazioni nel tempo attraverso l'esperienza. Le applicazioni del machine learning sono molteplici, spaziano dalla medicina alla guida autonoma delle automobili, e negli ultimi anni sono diventate molto note al pubblico grazie allo sviluppo degli assistenti virtuali come ChatGPT e dei generatori di immagini come Dall-e.

In questa sezione è spiegato il funzionamento di base delle reti neurali ponendo particolare attenzione ai concetti utilizzati nei capitoli successivi della tesi.

### 4.1 Reti neurali

Le reti neurali sono un modello di machine learning che ha avuto particolare successo negli ultimi anni. Nonostante questa popolarità recente, le prime reti neurali risalgono agli albori dell'informatica [8], tuttavia solo da pochi anni gli sviluppi tecnologici hanno

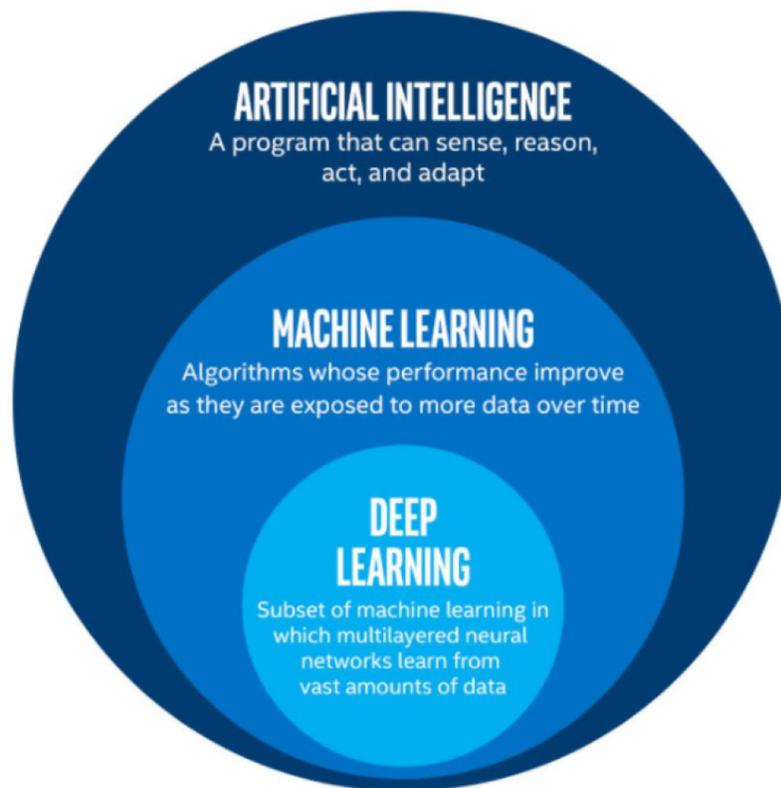


Figura 4.1: Relazione tra i concetti di intelligenza artificiale, machine learning e deep learning (tratto da [51])

permesso di utilizzarle con successo. I principali fattori che hanno portato a questa diffusione recente sono:

- **Hardware:** l'addestramento di una rete neurale richiede una grande potenza di calcolo che non era disponibile fino a pochi anni fa alla maggior parte dei ricercatori.
- **Disponibilità di dati:** le reti neurali sono efficaci solo se è disponibile una quantità di dati sufficiente ad addestrarle, e questi sono spesso il fattore limitante. Tuttavia sono oggi disponibili in internet grandi archivi di dati di ogni tipo, che possono essere sfruttati per molteplici applicazioni.
- **Software:** la disponibilità di pacchetti software dedicati alle reti neurali (come il *deep learning toolbox* di MATLAB e *TensorFlow* per Python) ha permesso a molti ricercatori di sviluppare questo tipo di software anche senza una preparazione informatica particolarmente approfondita.

## 4.2 Funzionamento di base

Una rete neurale è un programma che imita il funzionamento di un cervello biologico, essa riceve un input che viene elaborato da una serie di neuroni artificiali, e produce un output numerico. Risulta utile definire alcuni concetti di base che verranno utilizzati nel proseguimento del capitolo e che sono necessari a comprendere il funzionamento di una rete neurale.

**Neurone:** è l'unità di base di una rete, riceve degli input numerici dall'esterno o da altri neuroni e produce un output (anch'esso numerico) che viene utilizzato dai neuroni successivi o come output complessivo della rete

**Funzione di attivazione:** ad ogni neurone è associata una funzione che determina il valore dell'output ( $R(x)$ ) data la somma degli input ricevuti ( $x$ ). Questa è la componente che introduce non-linearità nel programma, le funzioni di attivazione più comuni comprendono:

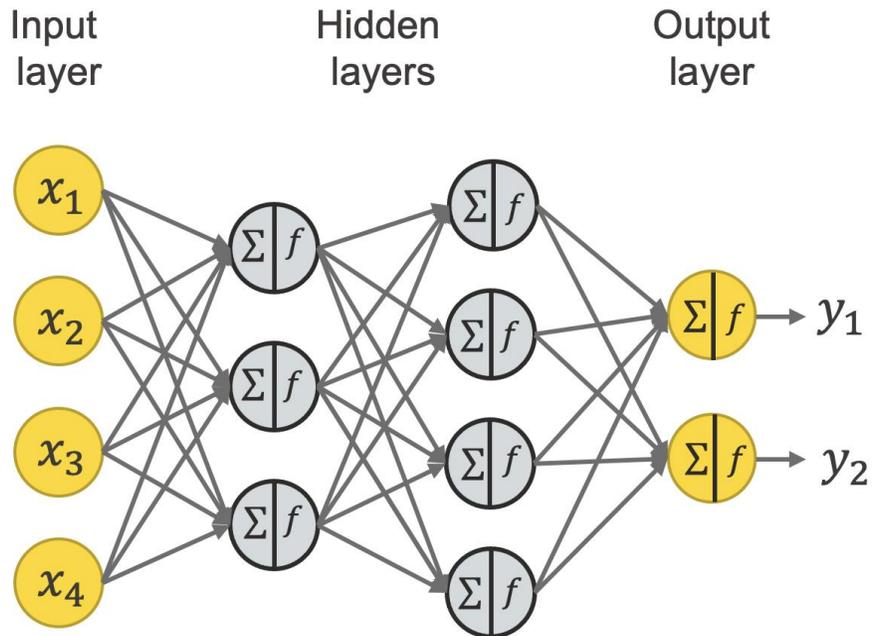


Figura 4.2: Tipico schema di una rete neurale: il primo strato riceve gli input, gli *strati nascosti* elaborano i dati, e l'ultimo strato fornisce l'output. (tratto da [31])

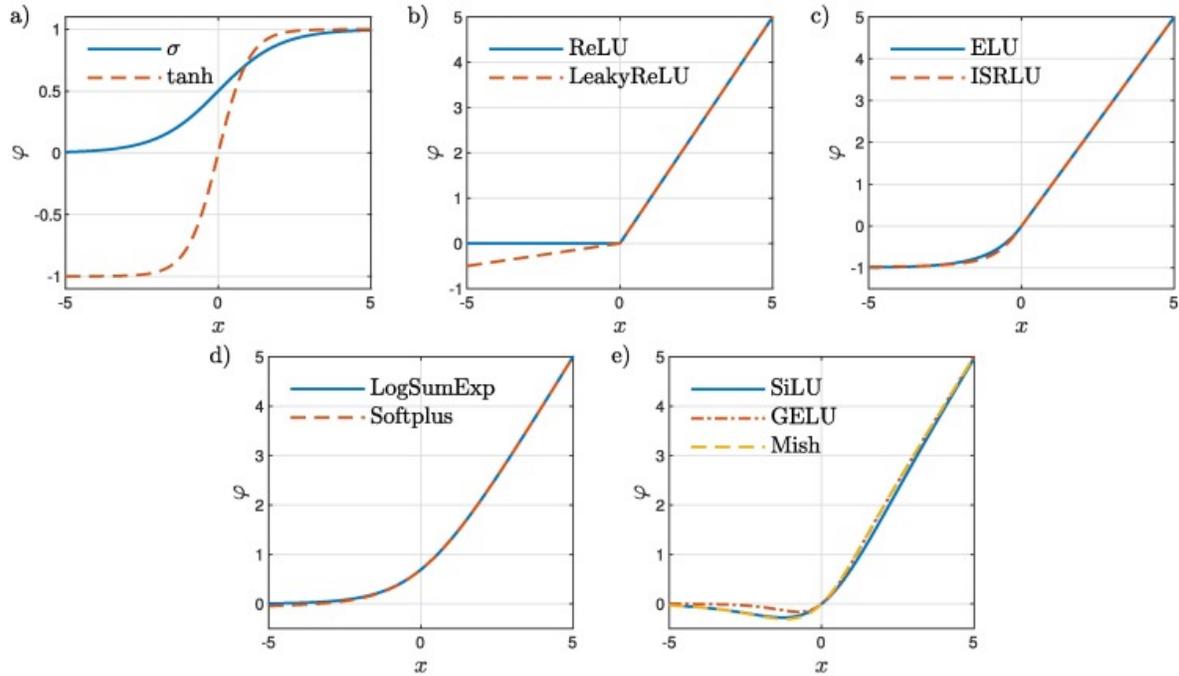


Figura 4.3: Grafici di alcune tra le funzioni di attivazione più comuni (tratti da [62])

- ReLU (Rectified Linear Unit)  $R(x) = \max(0, x)$
- Sigmoid  $R(x) = \frac{1}{1+e^{-x}}$
- Tangente iperbolica  $R(x) = \tanh(x)$
- leaky ReLU  $R(x) = \max(\alpha x, x)$
- ELU (Exponential Linear Unit)  $R(x) = \begin{cases} x & \text{se } x > 0 \\ \alpha \cdot (e^x - 1) & \text{se } x \leq 0 \end{cases}$

**Pesi:** ad ogni neurone è assegnato un peso per ogni input che riceve, questi sono i parametri della rete, che vengono variati durante il processo di training per migliorare le prestazioni.

**Strati:** i neuroni di una rete sono organizzati in strati, ogni strato riceve input da quello precedente e invia gli output a quello successivo. Tipicamente l'architettura della rete è definita dai suoi strati, in base alla loro tipologia e al numero di neuroni presenti.

**Addestramento:** è il processo tramite il quale vengono variati i pesi. La rete predice l'output relativo a una serie di input e calcola l'errore commesso, successivamente i pesi vengono modificati allo scopo di migliorare l'accuratezza.

**Loss function:** è una funzione che rappresenta l'errore commesso dalla rete durante l'addestramento. Solitamente è possibile individuare un parametro che misura l'efficacia della rete ( ad esempio per una classificazione categorica è la percentuale di accuratezza, per una previsione numerica può essere l'errore quadratico medio), ma questo potrebbe non essere trattabile dall'ottimizzatore [2], è quindi necessario definire una funzione, detta *loss function* strettamente legata a questo parametro. L'obiettivo dell'ottimizzatore è quindi minimizzare la *loss function*, i problemi di machine learning sono quindi problemi di ricerca del minimo di una funzione che può avere migliaia o anche milioni di parametri (i pesi). Esistono diverse funzioni utilizzabili come *loss function* a seconda della tipologia di problema.

**Output layer:** l'ultimo strato della rete determina il tipo di risposta che viene fornita. Se la grandezza da prevedere è uno scalare numerico si parla di un problema di regressione, ed è necessario un solo neurone nello strato di output. Se il problema è una classificazione categorica, cioè gli input devono essere assegnati a una di  $n$  classi, lo strato di output deve essere composto di  $n$  neuroni (tanti quante le classi), e il neurone con il valore più alto determina la classe prevista dalla rete.

### 4.3 Reti convoluzionali

Le reti neurali convoluzionali (CNN) sono un particolare tipo di rete neurale specializzato nell'analisi di dati bidimensionali (come le immagini). Caratteristica peculiare di queste reti è l'utilizzo di strati convoluzionali, che applicano filtri spaziali a porzioni locali dell'input, consentendo una maggiore generalizzazione e rilevamento delle caratteristiche visive.

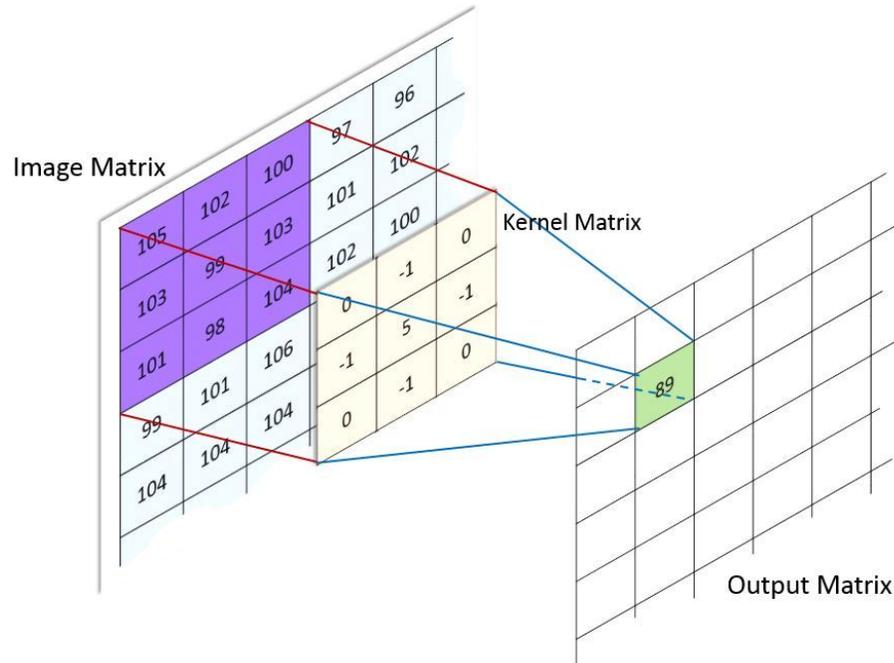


Figura 4.4: Rappresentazione grafica di uno strato convoluzionale: il filtro produce un valore di output per ogni posizione a cui è applicato, spostando il filtro sull'immagine si ottiene la mappa degli output (*feature map*). (immagine tratta da [9])

I parametri di uno strato convoluzionale sono i pesi del filtro (o *kernel*): questo viene applicato ad una parte dell'input e produce un output scalare, come rappresentato nella figura 4.4. La matrice degli output che si ottengono applicando il filtro in tutte le posizioni è detta *feature map*. Questa architettura permette al filtro di adattarsi per riconoscere una particolare caratteristica dell'input, che se presente causa un valore molto alto sulla *feature map*. Dato che nella maggior parte dei casi il programma deve riconoscere diverse caratteristiche, viene usato un numero  $n$  di filtri con pesi diversi, che produce quindi  $n$  diverse *feature maps*.

Dopo uno strato convoluzionale si posiziona solitamente uno strato di pooling, che riduce la dimensione del problema estraendo un solo valore da un gruppo (*pool*) di input. Le tipologie di pooling più comuni sono:

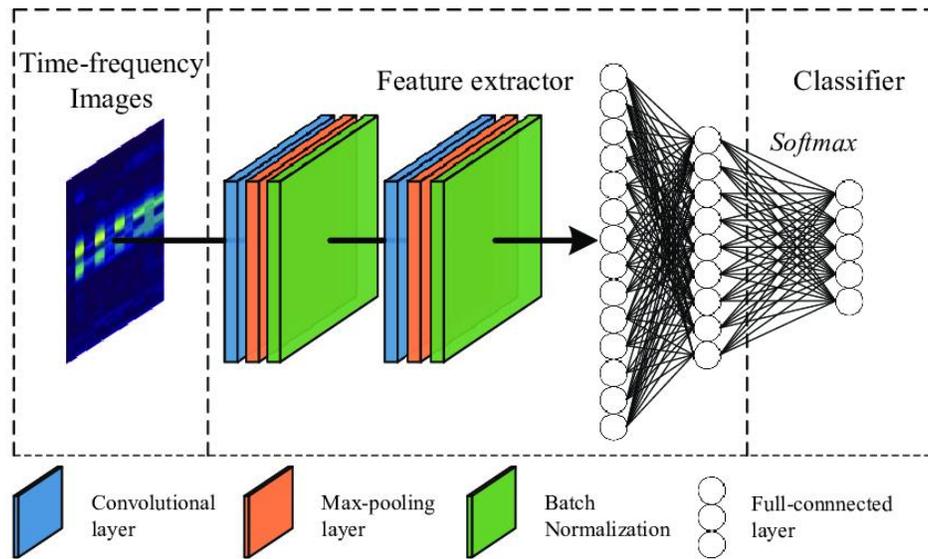


Figura 4.5: Schema tipico di una rete per l'analisi di immagini: i primi strati sono convoluzionali e riconoscono le caratteristiche dell'immagine, la seconda parte è costituita da strati densi e analizza i risultati per classificare l'immagine (immagine tratta da [59])

- Max-pooling (l'output è il valore più alto tra quelli del *pool*)
- Average-pooling (l'output è la media di tutti i valori nel *pool*)

l'architettura tipica di una rete convoluzionale prevede la suddivisione in due parti con scopi diversi: il *feature extractor* e il *classifier* [4.5]. Il *feature extractor* è la parte convoluzionale della rete, e dovrebbe quindi riconoscere le caratteristiche locali dell'input, mentre il *classifier* ha il compito di analizzare le *feature maps* e determinare cosa è rappresentato nell'immagine. Il *classifier* è solitamente costituito da strati densi (strati in cui tutti i neuroni sono collegati a tutti i neuroni degli strati adiacenti) e termina nello strato di output.

## 4.4 Addestramento

Caratteristica fondamentale del machine learning è che il programma migliora le sue prestazioni imparando dai dati a disposizione. In una rete neurale i parametri che possono essere cambiati sono i pesi assegnati ai neuroni o ai filtri convoluzionali, questa operazione è svolta dall'ottimizzatore: un algoritmo che ha lo scopo di minimizzare la *loss function*.

I dati vengono divisi in piccoli gruppi (detti *mini batches*), la rete fornisce delle predizioni per ognuno degli elementi e l'ottimizzatore valuta l'errore commesso. Il compito dell'ottimizzatore è stimare il gradiente della *loss function* (i parametri sono troppi per calcolarlo esplicitamente) e aggiustare di conseguenza i pesi della rete per migliorarne le prestazioni. Ogni aggiornamento dei pesi è detto *iterazione*, quando l'intero dataset di addestramento è stato usato una volta si definisce il passaggio di un'*epoca*. Per ottenere una buona prestazione possono essere necessarie decine o centinaia di epoche (a seconda della dimensione del dataset e del numero di parametri della rete).

### 4.4.1 Gestione dei dati

I dati raccolti sono, nella maggior parte dei casi, una risorsa limitata, e sono usati per due diversi scopi: allenare la rete e verificare le sue capacità predittive. Per fare questo si suddivide il dataset in 3 gruppi: l'insieme di training, l'insieme di validazione e l'insieme di test (quest'ultimo non è sempre necessario).

Il training set è quello che viene effettivamente utilizzato per l'addestramento, i pesi vengono quindi aggiornati basandosi sulla prestazione della rete su questi dati (l'accuratezza di training). L'insieme di validazione è necessario per valutare periodicamente l'efficacia della rete su dati nuovi: dopo un certo numero di iterazioni (a discrezione del programmatore) si calcola l'accuratezza sull'insieme di validazione, questo dato viene comunicato al programmatore, ma non viene usato per aggiornare i pesi. La rete non impara dall'insieme di validazione, ma solo dall'insieme di training, questo consente al programmatore di accorgersi dell'insorgenza dell'overfitting (il caso in cui la rete rico-

nosce molto bene i casi dell'insieme di training ma non riesce a generalizzare quando affronta casi nuovi). L'insieme di test è utilizzato solo dopo l'addestramento per effettuare delle valutazioni aggiuntive, e può essere eliminato in caso il dataset sia molto limitato.

Durante l'addestramento, l'ottimizzatore traccia un grafico che mostra l'accuratezza di training e l'accuratezza di validazione, da questo grafico è possibile valutare l'efficacia dell'addestramento e l'insorgenza dell'overfitting anche prima del completamento del processo. La suddivisione del dataset è effettuata a discrezione del programmatore: la maggior parte dei dati dovrebbe essere assegnata all'insieme di training, che determina le prestazioni che la rete può raggiungere, tuttavia un insieme di validazione troppo piccolo non permette di valutare correttamente l'efficacia dell'addestramento.

#### 4.4.2 Overfitting

L'overfitting è un problema estremamente comune nell'addestramento delle reti neurali. Si tratta del fenomeno per cui la rete si adatta troppo bene ai dati di training, memorizzando anche il rumore e le peculiarità di quello specifico set di dati invece di riconoscere le caratteristiche generali del problema. Questo si manifesta nel grafico di addestramento come una forte differenza tra accuratezza di training e di validazione (figura [4.7](#)): la rete infatti riconosce molto bene i dati di training, su cui è stata allenata, ma non quelli di validazione, che sono nuovi.

Essendo un problema molto comune è anche molto studiato: in generale l'azione più efficace nel contrasto all'overfitting è espandere il dataset, ma questo è raramente possibile. Sono state quindi sviluppate delle strategie per ritardare l'overfitting il più possibile, come la normalizzazione o il dropout.

Un leggero overfitting non è un problema per una rete neurale: l'obiettivo finale è che il programma sia in grado di riconoscere in modo efficace dei casi nuovi, e questa capacità viene misurata dall'accuratezza di validazione, l'accuratezza di training finale non ha nessun effetto sulle prestazioni della rete.

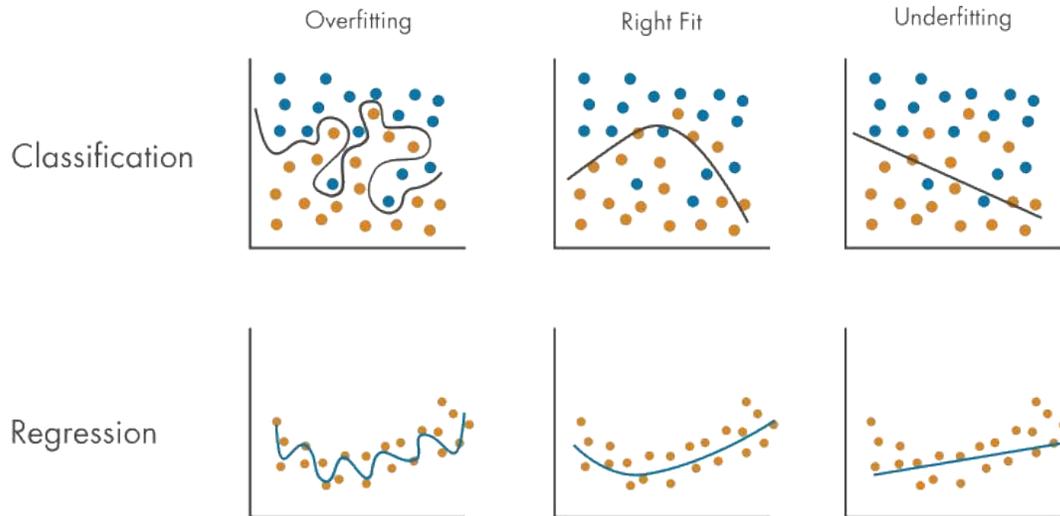


Figura 4.6: Rappresentazione grafica del concetto di overfitting (immagine tratta da [39])

Anche il numero di epoche in cui effettuare l'addestramento è a discrezione del programmatore, in generale si procede fino al raggiungimento di un *plateau*, ossia il punto in cui le prestazioni cessano di migliorare. In casi particolari l'accuratezza di training potrebbe arrivare al 100%, questo significa che l'ottimizzatore ha trovato il minimo della *loss function* e le iterazioni successive non hanno effetto.

## 4.5 Reti neurali in SHM

Nell'ambito dello structural health monitoring le reti neurali sono state usate da vari gruppi di ricerca e per scopi molto diversi tra loro.

Alcuni studi riescono, a partire dai dati dei sensori (reali o simulati) a determinare la posizione dell'impatto. Un articolo di Sarego et al. [54] usa una serie di simulazioni per addestrare la rete a riconoscere la posizione di impatto e il picco di forza, sono utilizzati

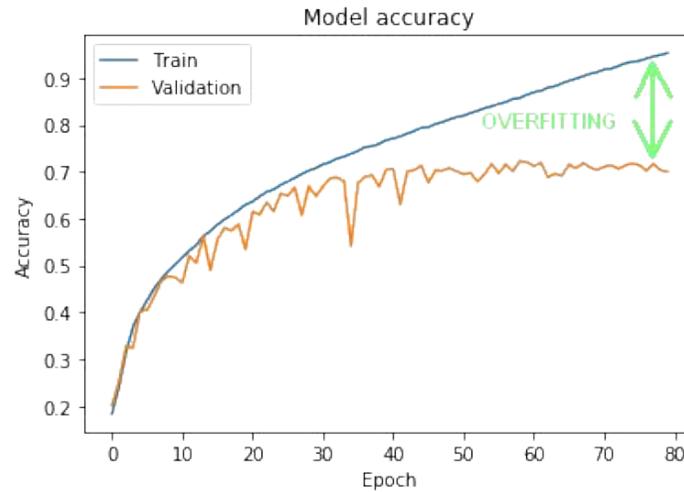


Figura 4.7: Grafico di addestramento tipico di una rete neurale che presenta overfitting (immagine tratta da [24])

inoltre degli algoritmi genetici per migliorare ulteriormente i risultati. Questo metodo ha permesso di ottenere un errore medio di 12.16 mm sulla posizione dell’impatto (su un pannello di  $450 \times 375 \text{ mm}^2$ ), e un errore tra il 10% e il 15% sul picco di forza ricostruito. Lo studio non si occupa però della delaminazione, che non è simulata, ma si limita a studiare un pannello integro.

Altri articoli convertono i dati dei sensori in immagini per sfruttare le reti convoluzionali, utilizzando sia approcci di SHM attivi che passivi. In particolare uno studio di De Oliveira [55] applica dei trasduttori piezoelettrici ad alcune piastre in cui è presente un danno, grazie ad una rete convoluzionale è stato possibile individuare la posizione del danno a partire dalle risposte dei sensori.

Un articolo di Sharif-Khodaei et al. [37] costruisce una rete neurale che non utilizza tutti i dati dei sensori, ma solo alcuni valori rielaborati, come il picco di deformazione o il ritardo nella ricezione del segnale per trovare la posizione di un impatto. Questo metodo è un ibrido tra gli algoritmi tradizionali di SHM e l’utilizzo delle reti neurali, in cui è necessaria un’analisi preliminare dei dati prima di fornirli alla rete, che trova il punto di contatto.

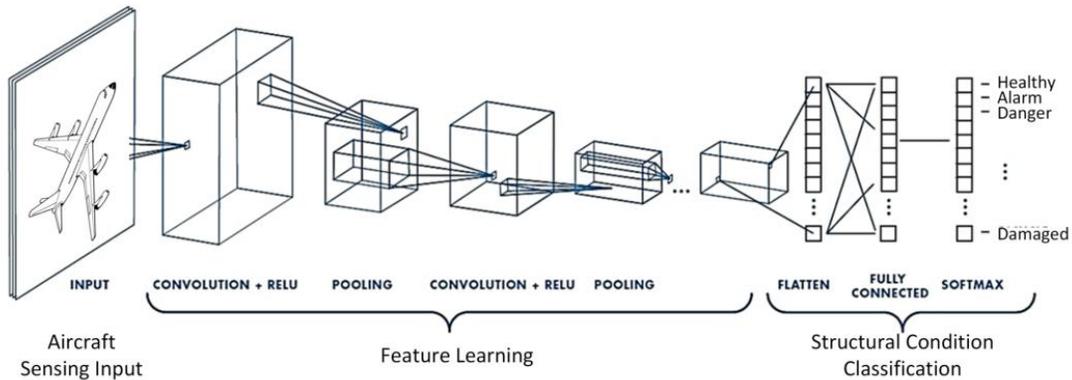


Figura 4.8: Schema di funzionamento di un sistema di monitoraggio strutturale basato su una rete convoluzionale (tratto da [61])

Esistono poi altri modi di utilizzare le reti neurali nello studio degli impatti, è possibile ad esempio prevedere l'entità del danno a partire dalle informazioni sulla struttura e dall'energia dell'urto [12]. Questo non è structural health monitoring, in quanto non utilizza i dati dei sensori posti sulla struttura, ma necessita dei parametri dello scenario (in particolare dell'energia di impatto, che è generalmente ignota), può essere considerato un'alternativa più rapida alla simulazione FEM.

# Capitolo 5

## Modello FEM

Lo scopo della ricerca è allenare una rete neurale che possa prevedere l'area delaminata di una struttura in composito. La delaminazione è uno degli aspetti più critici in una simulazione FEM, dato che i parametri necessari sono spesso ignoti. Per verificare l'affidabilità del modello per quanto riguarda le proprietà del materiale e la delaminazione sono stati considerati gli articoli di Rivallant [42] e Tan [38], dove è studiato il comportamento di una piastra in CFRP (carbon fibre reinforced plastic) a seguito di un impatto, svolgendo simulazioni ed esperimenti.

Accertata la validità dei parametri del materiale è stato costruito un secondo modello più semplice che permette di effettuare un numero di simulazioni sufficiente ad allenare la rete (in particolare è necessario ridurre il numero di lamine).

Tutte le simulazioni sono state svolte tramite Abaqus/explicit, i solutori espliciti sono infatti particolarmente adatti alla simulazione di eventi di breve durata in cui è presente un grande numero di elementi, come i casi di impatto analizzati in questa tesi [52].

## 5.1 Modello di verifica

Per verificare il modello del materiale e della zona coesiva è stata impostata una simulazione tramite Abaqus di una piastra come quella esaminata negli articoli [38] e [42]. Questi utilizzano due approcci diversi per la modellazione FEM: in particolare il primo usa il metodo *stacked shell*, e il secondo il metodo *ply splitting*. Siccome i risultati ottenuti sono molto simili, la validazione è stata effettuata su un modello di tipo *stacked shell*, che è più semplice da implementare e permette di ridurre i tempi di simulazione. La piastra è appoggiata su un supporto con un foro centrale, e viene colpita da un proiettile indeformabile di massa  $2\text{kg}$ .

Lo scenario è tipico dell'*Airbus industries test method* [6](AITM), sviluppato per valutare le prestazioni di un materiale a seguito di un urto.

### 5.1.1 Geometria

La piastra è lunga 150 mm e larga 100 mm, è composta da 16 lamine pre-impregnate di spessore 0.26 mm, lo spessore totale del campione è quindi 4.16 mm. La *stacking sequence* è  $[0_2, 45_2, 90_2, -45_2]_s$ , si tratta quindi di un layup simmetrico e bilanciato.

Il proiettile è modellato come una semisfera rigida di raggio 6 mm, a cui è associata una massa di 2 kg e un momento di inerzia unitario su tutti gli assi (che non ha influenza sulla simulazione, ma è necessario un valore non nullo per evitare che il processore si interrompa incontrando di una singolarità).

Il supporto è rettangolare delle stesse dimensioni della piastra, ma presenta un foro al centro di dimensioni  $125 \times 75 \text{ mm}^2$ , anche questo è un corpo rigido ed è fissato nella sua posizione da un incastro.

### 5.1.2 Materiale

Il materiale è un composito di fibra di carbonio in matrice epossidica (T700/M21) ampiamente studiato in letteratura [38] [42] [56] [36], di cui quindi è possibile trovare non solo i parametri meccanici (forniti dal produttore) ma anche i parametri per modellare

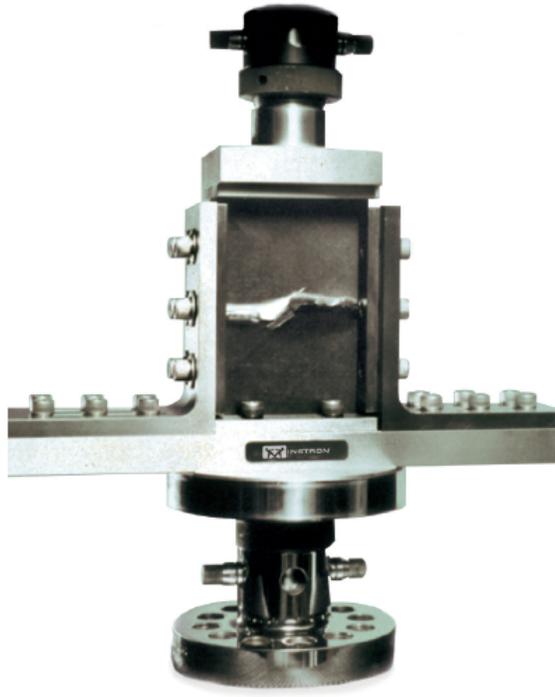


Figura 5.1: Setup sperimentale per un test AITM per misurare la resistenza a compressione di una lastra in composito dopo un impatto (immagine tratta da [25])

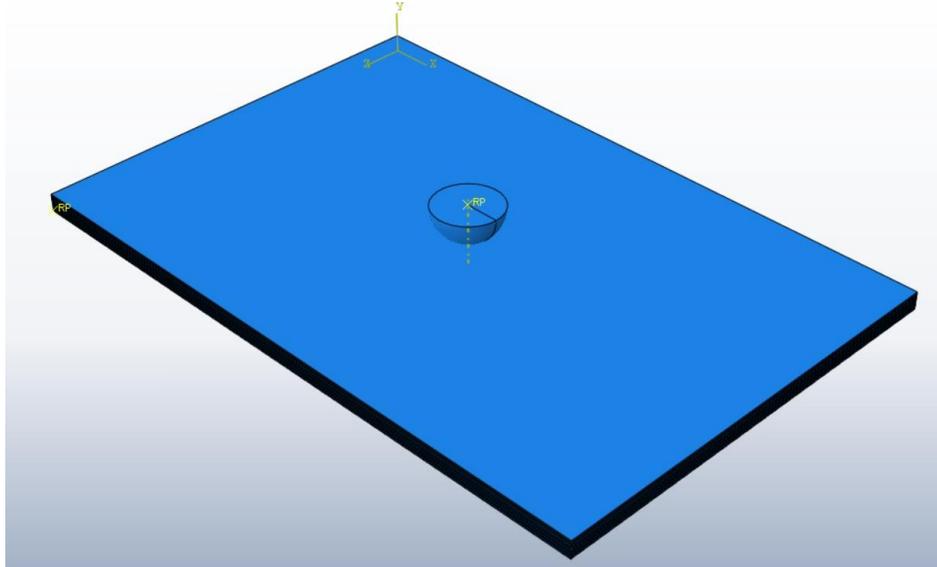


Figura 5.2: Geometria del modello in esame

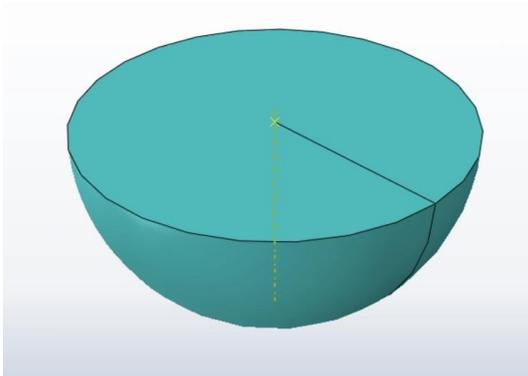


Figura 5.3: Dettaglio della geometria del proiettile

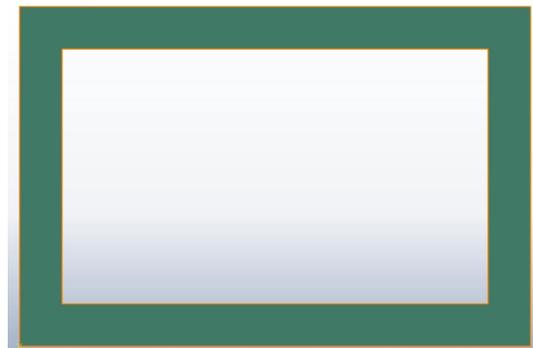


Figura 5.4: Geometria del supporto

il danno, che solitamente devono essere ricavati dal confronto dei dati sperimentali con alcune simulazioni standard, come il *double cantilever beam* [57].

Le proprietà meccaniche utilizzate per la simulazione sono raccolte nella tabella 5.1.2

Proprietà		Valore	Unità
Densità	$\rho$	1600	kg/m <sup>3</sup>
Modulo di Young	$E_1$	130	GPa
	$E_2$	7.7	GPa
	$E_3$	7.7	GPa
Coefficiente di Poisson	$\nu_{12}$	0.3	-
	$\nu_{13}$	0.3	-
	$\nu_{23}$	0.35	-
Modulo di taglio	$G_{12}$	4.8	GPa
	$G_{13}$	4.8	GPa
	$G_{23}$	3.8	GPa

Tabella 5.1: Proprietà meccaniche del materiale T700/M21

### 5.1.3 Interfaccia

Nei laminati compositi, il punto di contatto tra due lamine è solitamente il punto più debole della struttura, e la resistenza dell'interfaccia è sostanzialmente minore di quella del materiale, in questi casi è particolarmente adatto il modello di zona coesiva [4]. Abaqus permette di modellare l'interfaccia tra due lamine in due diversi modi: elementi coesivi o superfici coesive. Dato l'elevato numero di elementi già presenti nel modello, è stato scelto di non aggiungerne altri, utilizzando quindi le superfici coesive per simulare la delaminazione.

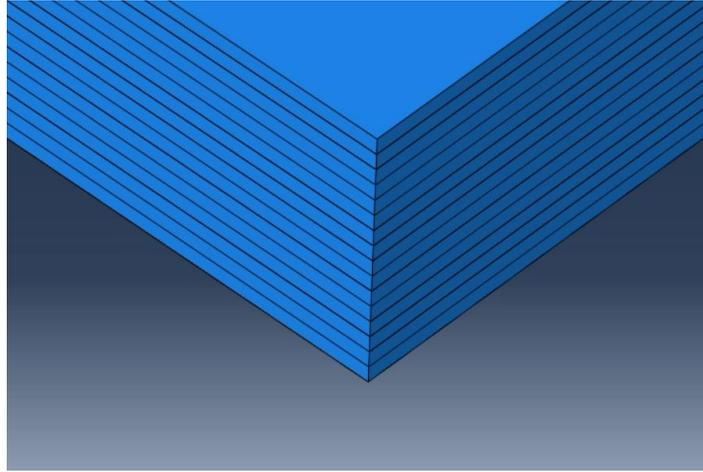


Figura 5.5: Dettaglio del modello FEM che mostra la suddivisione della piastra in 16 lamine

La relazione che lega tensione e spostamento tra due nodi di una superficie coesiva è di tipo bilineare, sono quindi necessari 3 parametri per definire il comportamento dell'interfaccia:

- $K$ : la rigidezza durante il comportamento elastico
- $G$ : l'energia di frattura
- $\tau^0$ : la tensione massima sopportabile

La rigidezza di una superficie coesiva non è un parametro fisico, in quanto una zona di spessore nullo dovrebbe avere nominalmente una rigidezza infinita, per trovare un valore adatto bisogna quindi valutare l'errore a cui una rigidezza finita porta. Turon et al., in un articolo [28], forniscono delle indicazioni sul valore di rigidezza adatto: perché l'influenza della zona coesiva sulla rigidezza complessiva della struttura sia piccola, è necessario che valga  $E_3 \ll Kt$  dove  $E_3$  è il modulo di Young nella direzione considerata, e  $t$  è lo spessore della singola lamina. Da questa relazione si ricava

$$K = \frac{\alpha E_3}{t}$$

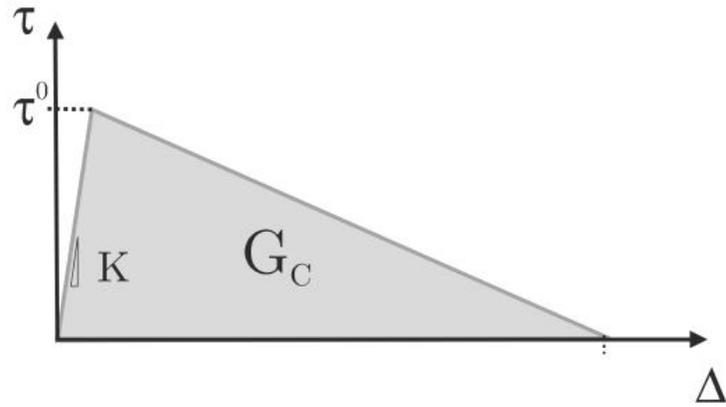


Figura 5.6: Modello coesivo bilineare: relazione tra tensione e deformazione (immagine tratta da [28])

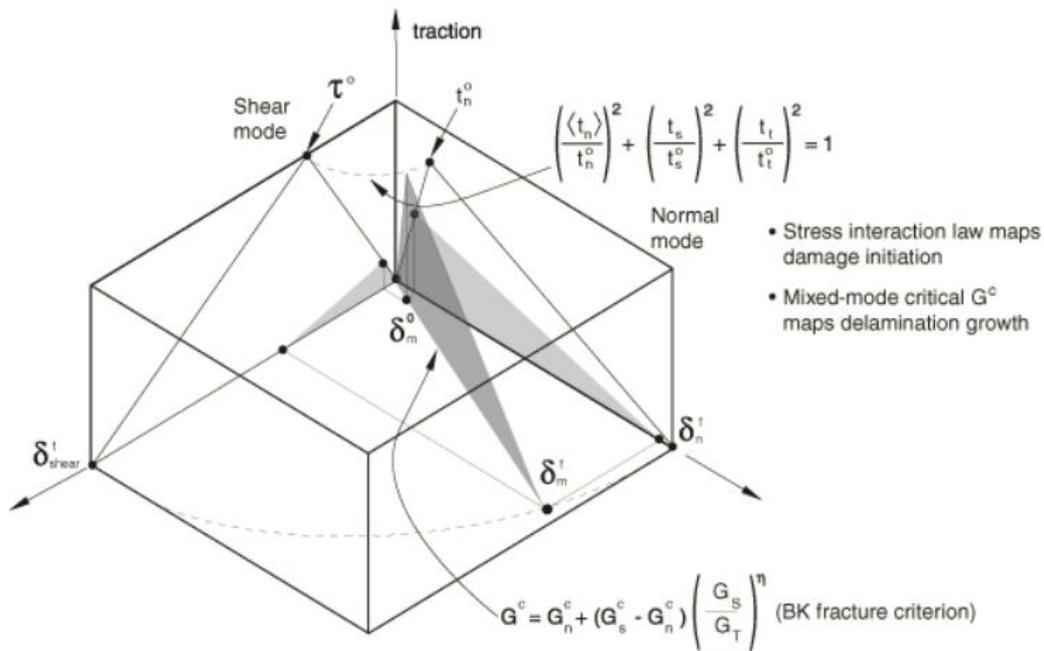


Figura 5.7: Rappresentazione grafica dell'interazione tra i diversi modi di frattura secondo il modello usato in questa tesi (immagine tratta da [52])

dove  $\alpha \gg 1$  è un numero che rappresenta il rapporto tra la rigidezza della zona coesiva e quella della piastra: l'errore relativo indotto dall'interfaccia è quindi  $\frac{1}{K}$ . Ponendo  $K = 50$  si ottiene un errore del 2%, considerato accettabile per la maggior parte dei casi. Il valore di rigidezza usato per il modello di verifica è quindi  $1.5 \times 10^{15}$  N/m<sup>3</sup>. Sebbene aumentare ulteriormente la rigidezza porti a risultati più accurati, questo causa anche un aumento dei tempi di calcolo, poiché l'incremento temporale che Abaqus/explicit usa nelle simulazioni è proporzionale a  $1/\sqrt{K}$ . Risulta quindi utile mantenere il valore di  $K$  limitato, per quanto possibile senza ridurre l'accuratezza del modello.

I valori di  $\tau^0$  e  $G$  sono stati presi dalla letteratura [38][42], inoltre è stato utilizzato il criterio di fallimento *quadratic traction*, che valuta l'interazione tra i diversi modi di frattura secondo la relazione

$$\left(\frac{\langle \tau_1 \rangle}{\tau_1^0}\right)^2 + \left(\frac{\tau_2}{\tau_2^0}\right)^2 + \left(\frac{\tau_3}{\tau_3^0}\right)^2 < 1$$

dove  $\tau_1, \tau_2, \tau_3$  sono le tensioni nelle tre direzioni (una normale e due di taglio), e  $\tau_1^0, \tau_2^0, \tau_3^0$  sono le rispettive massime tensioni alla rottura. Per l'evoluzione della delaminazione è implementato il criterio B-K[34], che stima l'evoluzione del danno in risposta a una sollecitazione *mixed-mode*. L'interazione tra i diversi modi di fallimento è un problema molto complesso, ma il metodo di Benzeggagh-Kenane riesce a descrivere bene il comportamento delle superfici di delaminazione, ed è ampiamente utilizzato in letteratura.

I valori utilizzati per il modello bilineare delle superfici coesive sono raccolti nella tabella 5.1.3.

#### 5.1.4 Criterio di rottura

Negli articoli considerati, è posta molta enfasi sul danno al materiale sotto forma di rottura delle fibre o della matrice, infatti le simulazioni sono svolte utilizzando una subroutine personalizzata programmata dagli autori. Non essendo questo il focus di questa tesi, è stato deciso di utilizzare il criterio di rottura di Hashin [60], già disponibile in

Proprietà		Valore	Unità
Massima tensione	$\tau_1^0$	20	<i>MPa</i>
	$\tau_2^0$	36	<i>MPa</i>
	$\tau_3^0$	36	<i>MPa</i>
Energy release rate	$G_{1C}^0$	500	<i>J/m<sup>2</sup></i>
	$G_{2C}$	1600	<i>J/m<sup>2</sup></i>
	$G_{3C}$	1600	<i>J/m<sup>2</sup></i>
Esponente di B-K	$\eta$	1.45	-

Tabella 5.2: Proprietà di interfaccia del materiale T700/M21

Abaqus, che è ampiamente utilizzato in letteratura per simulare la frattura dei compositi polimerici. Questo criterio richiede la definizione delle tensioni di fallimento nelle 3 direzioni (lungo le fibre, perpendicolare alle fibre e taglio) per trazione e compressione. I parametri utilizzati per implementare il criterio di Hashin sono raccolti nella tabella [5.1.4](#).

Proprietà		Valore	Unità
Massima trazione longitudinale	$X^T$	2080	<i>MPa</i>
Massima compressione longitudinale	$X^C$	1250	<i>MPa</i>
Massima trazione trasversale	$Y^T$	60	<i>MPa</i>
Massima compressione trasversale	$Y^C$	290	<i>MPa</i>
Massima tensione di taglio	$S^L$	110	<i>MPa</i>
Energia di frattura longitudinale in trazione	$G_1^T$	133000	<i>N/m</i>
Energia di frattura longitudinale in compressione	$G_1^C$	10000	<i>N/m</i>
Energia di frattura trasversale in trazione	$G_2^T$	500	<i>N/m</i>
Energia di frattura trasversale in compressione	$G_2^C$	1600	<i>N/m</i>

Tabella 5.3: Proprietà di resistenza del materiale T700/M21

### 5.1.5 Mesh

Seguendo le indicazioni di Tan et al. [38], le lamine sono state suddivise in elementi di 1.5 mm di lato, per un mesh composta di  $100 \times 67$  elementi per lamina: le 16 lamine portano quindi il numero di elementi a 107200, numero che spiega i lunghi tempi di calcolo necessari. Gli elementi utilizzati sono di tipo *continuum shell* a integrazione ridotta (SC8R) dalla libreria di Abaqus/explicit. Questi sono elementi tridimensionali a 6 facce pensati per casi in cui lo spessore è significativamente minore delle altre dimensioni [52]: permettono quindi di simulare una piastra considerando tutti gli effetti dello spessore. Studi precedenti [22][43] hanno inoltre evidenziato come gli elementi *continuum shell* siano più adatti di quelli *conventional shell* alla simulazione degli impatti, dato che, a differenza di questi ultimi, permettono la compressione anche nella direzione dello spessore, e non introducono quindi una rigidità aggiuntiva fittizia.

Oltre ai 107200 elementi della piastra sono simulati anche il supporto e il proiettile, entrambi composti da elementi piani indeformabili (R3D4 e R3D3), che permettono di alleggerire la simulazione. Il numero totale di elementi presenti nel modello è quindi 108187. Il supporto è fissato in posizione da un incastro e non richiede quindi la definizione di proprietà inerziali, al proiettile è invece associata una massa di 2 kg e un momento di inerzia unitario su tutti gli assi. La massa puntiforme è collocata nel centro della semisfera.

### 5.1.6 Steps

Una simulazione in Abaqus è composta di diversi *steps*, che permettono di definire le condizioni di carico e le richieste di output, in questo caso ne sono previsti 3:

- initial
- velocity
- vibration

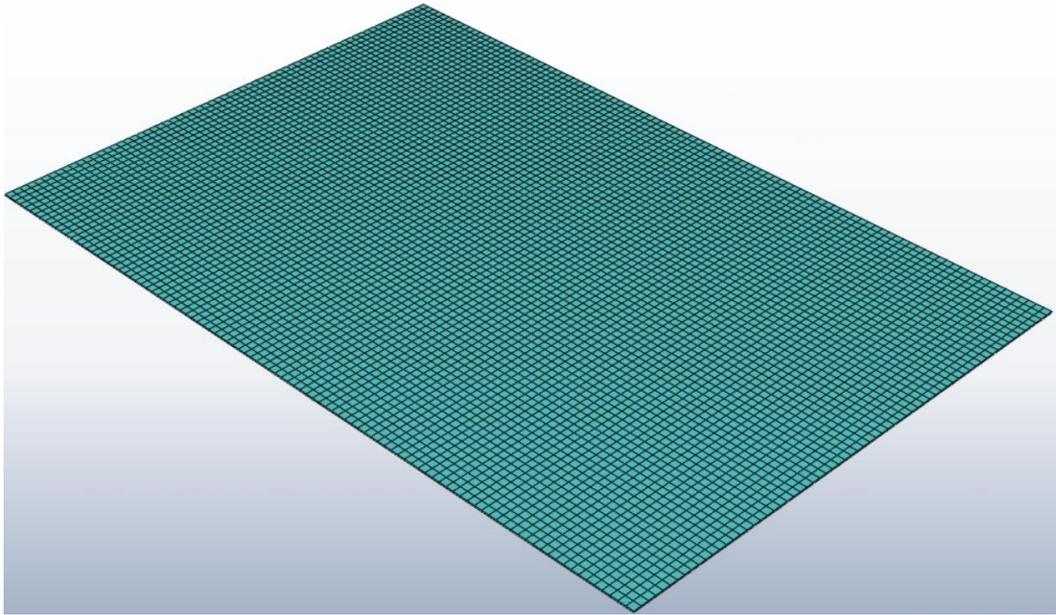


Figura 5.8: Mesh di una lamina per il modello di validazione

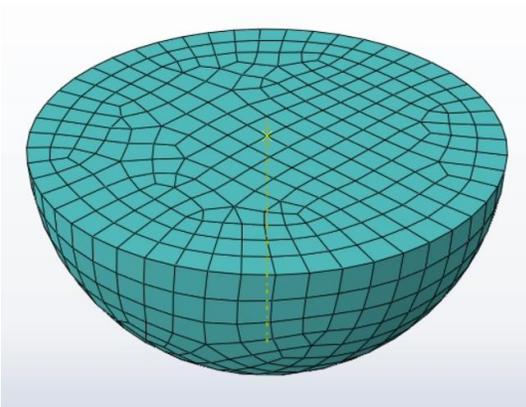


Figura 5.9: Mesh del proiettile

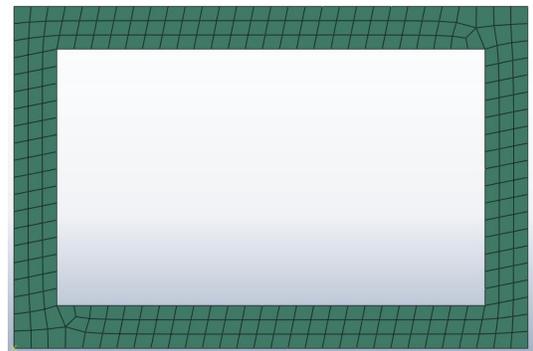


Figura 5.10: Mesh del supporto

Il primo step (*initial*) è obbligatorio per tutte le analisi e serve al programma per inizializzare il modello, è possibile definire solo alcune tipologie di vincoli e ha durata nulla, quindi non sono previsti output. In questo step è imposto solo l'incastro al supporto.

Lo step *velocity* è di tipo `dynamic explicit` e serve ad imporre la velocità desiderata al proiettile: viene quindi aggiunta una condizione al contorno di tipo `velocity` perpendicolare alla piastra applicata al `reference point` del proiettile. La durata è molto breve ( $1 \mu s$ ) e non sono richiesti output, poiché il contatto tra piastra e proiettile avviene solo nello step successivo.

L'ultimo step è anch'esso di tipo `dynamic explicit`, e rimuove il vincolo sulla velocità del proiettile. Durante questo step (di durata 4.5 ms) avviene il contatto e si simula la reazione della piastra. Gli output richiesti al solutore sono i seguenti:

- Campo di spostamento (U)
- Campo di accelerazione (A)
- Criterio di iniziazione della delaminazione (CSQUADSCRT)
- Degradazione della rigidezza di interfaccia (CSDMG)
- Degradazione della rigidezza degli elementi (SDEG)
- Forza di contatto tra piastra e proiettile (CFNM)

Ognuno di questi output è richiesto in un totale di 101 istanti temporali equispaziati nei 4.5 ms di durata dello step, quindi con una frequenza di 22.2 kHz.

### 5.1.7 Risultati

I risultati della simulazione sono stati confrontati con quelli degli articoli di riferimento. I dati a disposizione sono l'andamento della forza di contatto e l'area delaminata, sono quindi stati richiesti questi output al solutore per poter giudicare la correttezza del modello.

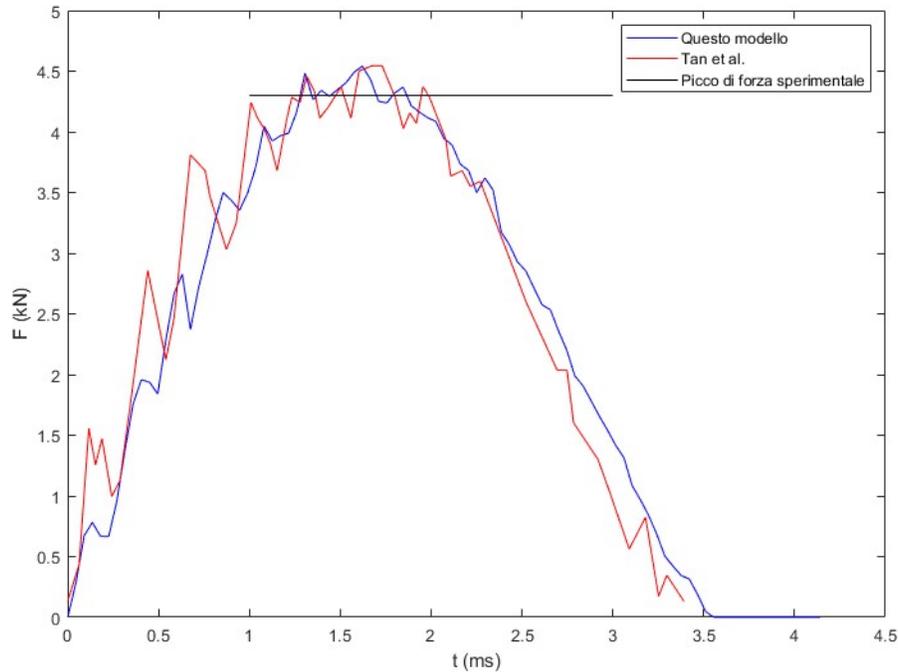


Figura 5.11: Confronto con i risultati del modello di Tan [38] per l'impatto da 6.5 J

La forza di contatto calcolata è molto simile a quella trovata numericamente da [38], e ai valori sperimentali trovati da [42]. Si nota che la durata del contatto è di circa 4ms, e che è poco influenzata dall'energia dell'impatto.

Le figure mostrano il confronto tra la forza di contatto calcolata nei due modelli per i due casi presi in esame: 6.5 J (5.11) e 29.5 J (5.12).

La superficie delaminata è diversa per ogni interfaccia, tuttavia la dimensione del danno trovato sperimentalmente (e quindi il valore che il modello deve predire) è l'involuppo di tutte le aree danneggiate. Attraverso un codice MATLAB è stato possibile trovare l'insieme dei nodi in cui avviene almeno una delaminazione tra le 15 interfacce presenti nel modello. I grafici risultanti mostrano la forma della zona delaminata e permettono di calcolarne l'area. Il valore dell'area danneggiata calcolato da questo modello è  $464 \text{ mm}^2$  per l'impatto da 6.5 J e  $2683 \text{ mm}^2$  per l'impatto da 29.5 J. Questi

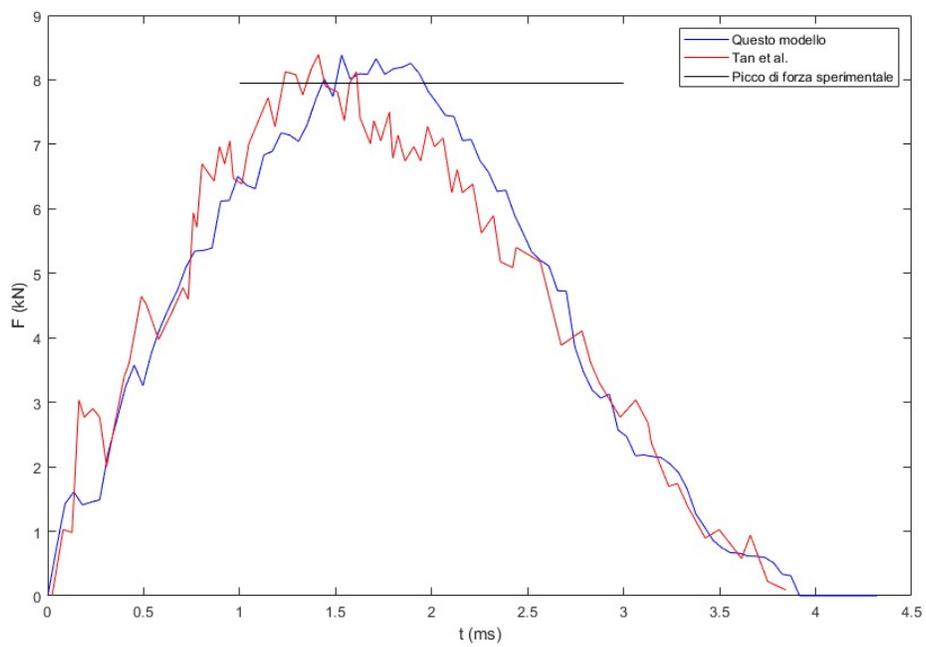


Figura 5.12: Confronto con i risultati del modello di Tan per l'impatto da 29.5 J

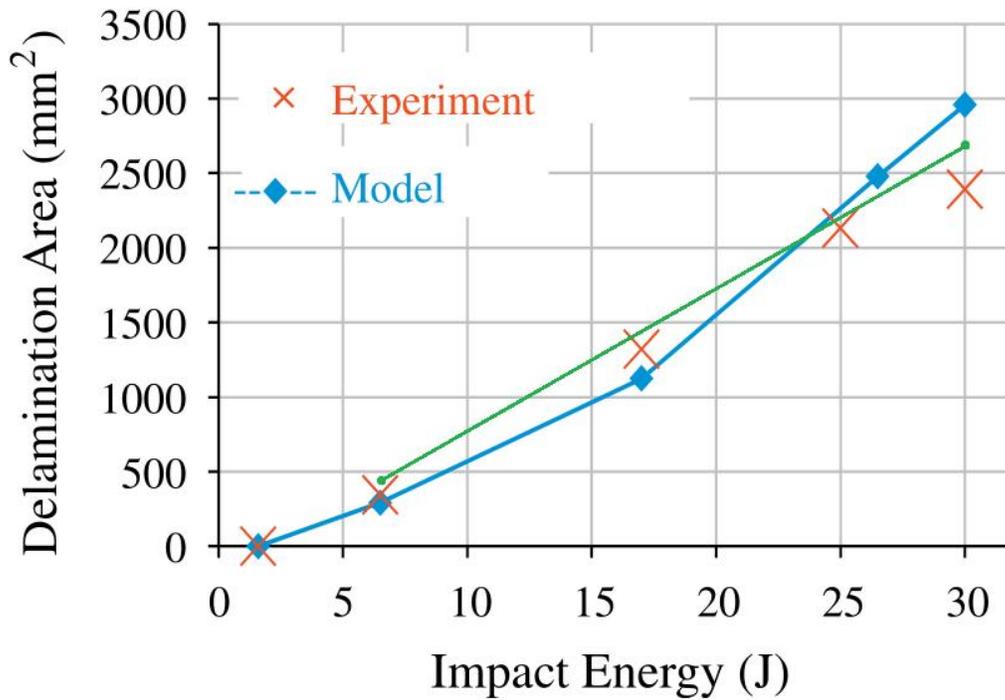


Figura 5.13: Area delaminata calcolata da [42] confrontata con i dati sperimentali, in verde sono aggiunti i risultati del modello di questa tesi

valori sono confrontati con quelli dell'articolo [42] nella figura 5.13, da cui si può vedere che sono molto vicini ai dati sperimentali.

Anche la forma dell'area delaminata è simile a quella trovata nell'articolo [42], come mostrato nella figura 5.1.7.

Nonostante le proprietà del materiale, la geometria e la dimensione degli elementi siano identiche a quelle dell'articolo di riferimento, sono presenti alcune differenze tra i risultati. Queste sono dovute alle poche differenze residue tra i modelli: il criterio di frattura del materiale e la discretizzazione nello spessore. Questo modello infatti definisce un elemento per ogni lamina, mentre il modello di Tan accoppia le lamine orientate nella stessa direzione all'interno dello stesso elemento.

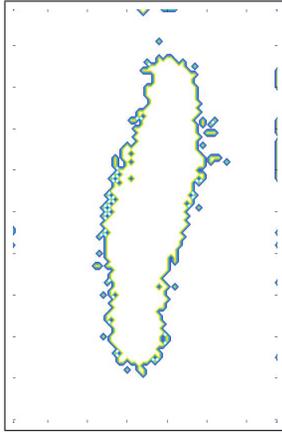


Figura 5.14: Inviluppo delle superfici delaminate nel modello proposto

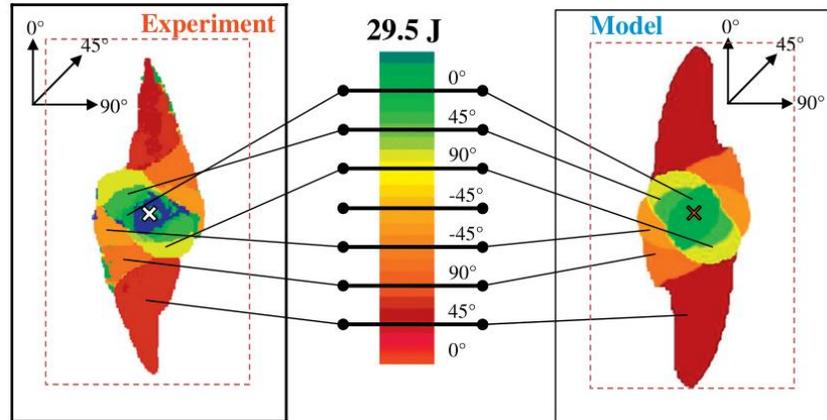


Figura 5.15: Dati sperimentali (a sinistra) e risultati del modello di Rivallant (a destra) [42]

## 5.2 Modello di lavoro

Dopo aver verificato che le proprietà del materiale riproducono bene i risultati sperimentali, è stato necessario costruire un secondo modello più rapido da simulare. Una singola run del modello precedente richiede infatti circa 25 ore con i computer virtuali dell'Università di Padova, che permettono di far lavorare 4 cores in parallelo. Questo valore è decisamente incompatibile con la necessità di svolgere un grande numero di simulazioni per allenare una rete neurale (è stato scelto di effettuarne 1000), quindi è stata cambiata la composizione della piastra per accelerare il processo.

I parametri che più influenzano il tempo di calcolo sono il numero di elementi e la rigidità dell'interfaccia, quindi le modifiche si sono concentrate sulla riduzione di questi due valori. Per ridurre il numero di elementi è stata costruito un laminato con pochi strati (4 invece di 16). La rigidità della zona coesiva è invece calcolata come

$$K = \frac{\alpha E_3}{t}$$

dove  $\alpha$  è il coefficiente moltiplicativo, e non può essere modificato a meno di aumentare l'errore del modello,  $E_3$  è il modulo di Young, che è determinato dal materiale, e  $t$  è lo spessore della lamina, ed è l'unico parametro modificabile, lo spessore delle 4 lamine che compongono la piastra è stato quindi alzato a 1 mm. Questi accorgimenti hanno permesso di ridurre il tempo necessario per una simulazione a 15 minuti.

Il modello risultante è poco realistico, poiché lo spessore delle lamine composite è solitamente minore (indicativamente 0.1-0.4 mm), tuttavia lo scopo di questa tesi non è costruire un sistema di SHM funzionante (dato che la piastra è solo simulata), ma verificare le potenzialità di questo approccio al problema.

Inoltre diversi articoli che studiano problemi simili [27][49][38] non utilizzano un elemento per ogni lamina nello spessore, ma uniscono quelle orientate nella stessa direzione in un solo strato. Nonostante non ci siano giustificazioni teoriche per questa semplificazione, i modelli riproducono bene i risultati sperimentali. Si può quindi considerare ogni strato di 1 mm come unione di diverse lamine più sottili e realistiche.

Data la riduzione nella massa del proiettile, anche la durata dell'impatto è diminuita: il tempo di simulazione è stato quindi accorciato da 4.5 ms a 4ms.

### 5.2.1 Sensori

Lo scopo della ricerca è simulare un sistema di SHM passivo per verificare la capacità di una rete neurale di rilevare la delaminazione. L'input fornito a questa rete è costituito dai dati registrati dai sensori applicati alla struttura (spostamenti o accelerazioni), mentre l'output è l'area delaminata. Per fare ciò, è stato richiesto ad Abaqus il campo di spostamento ed accelerazione non su tutti i nodi, ma su un sottoinsieme di 12 nodi che rappresentano i sensori.

Il posizionamento dei sensori sulla struttura è un problema molto complesso e studiato [5][53][45], tuttavia esula dagli scopi di questa tesi, è quindi stato scelto di posizionare i sensori in un pattern regolare sulla piastra, come mostrato nella figura 5.16

È importante anche assicurarsi che i sensori siano in grado di registrare onde a frequenze molto alte come quelle provocate dagli impatti: in questo caso la frequenza di

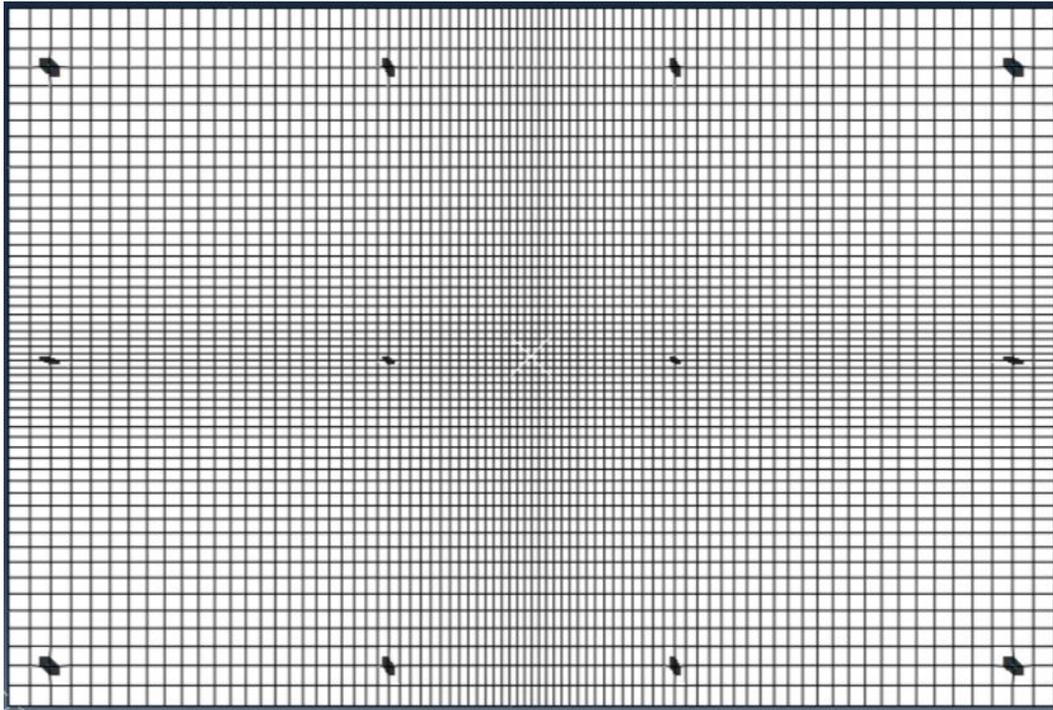


Figura 5.16: Modello di lavoro con evidenziata la posizione dei nodi scelti come sensori

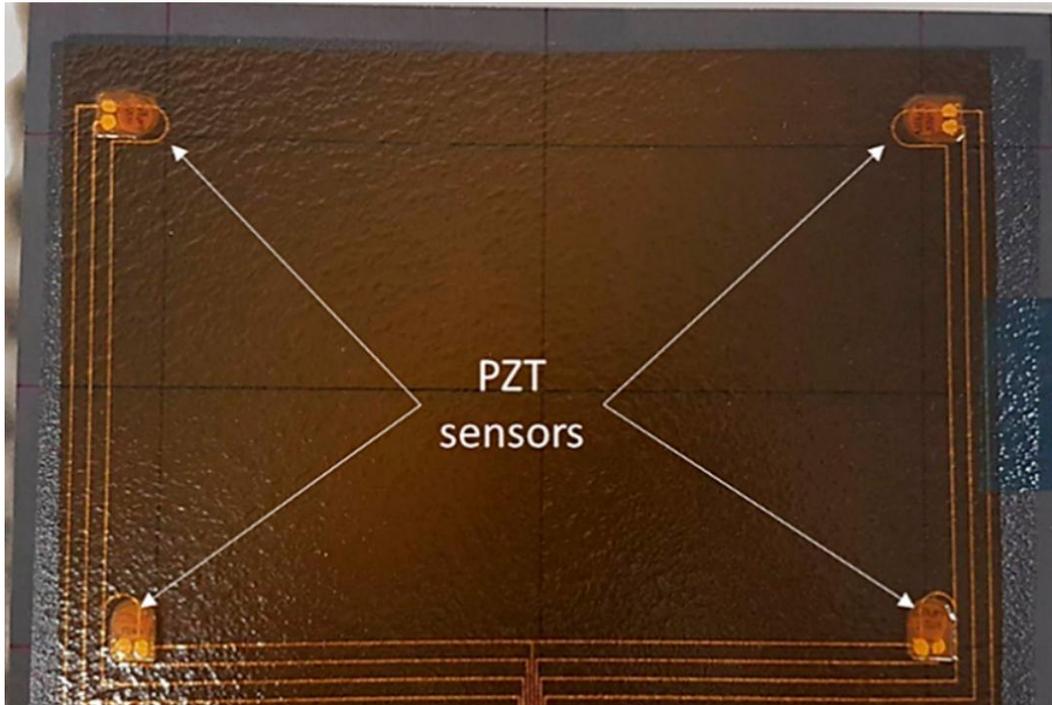


Figura 5.17: Foto di una pellicola con sensori sviluppata per applicazioni di SHM da [40]

campionamento è di 50 kHz, che sarebbe troppo alta per la maggior parte degli accelerometri disponibili in commercio, tuttavia esistono sensori sviluppati per questo tipo di applicazioni (solitamente accelerometri piezoelettrici) che permettono di utilizzare frequenze di campionamento anche molto più alte [40].

Per applicazioni di SHM i sensori più comunemente usati sono gli accelerometri, a causa della loro affidabilità, compattezza e basso costo, quindi sarebbe legittimo esportare solo il campo di accelerazione, tuttavia esistono studi che utilizzano gli spostamenti come input per sistemi di SHM [54], ed è stato scelto di investigare l'efficacia di entrambi i dati per determinare se esista un vantaggio dato dall'utilizzo dello spostamento.

### 5.2.2 Variabili considerate nelle simulazioni

Tramite MATLAB è stato possibile programmare il computer virtuale per eseguire 1000 simulazioni variando alcuni dati, in particolare sono stati scelti la massa e la velocità del proiettile. La posizione dell'impatto è sempre al centro della piastra, e non può essere cambiata a causa della mesh non uniforme, che permette di simulare bene solo gli urti che avvengono al centro. Inoltre altre ricerche hanno già fornito risultati positivi sulla capacità delle reti neurali di individuare la posizione di impatto.

La scelta di variare contemporaneamente massa e velocità, e non solo uno di questi è giustificata dal fatto che l'energia del proiettile non è sufficiente a definire completamente l'impatto, ma entrambi i parametri hanno influenza sulla dinamica del fenomeno. Il tempo di contatto tra lamina e proiettile ad esempio, dipende dalla massa ma non dall'energia, questo si può spiegare considerando (in prima approssimazione) il sistema come un oscillatore armonico: la rigidità è fornita dalla piastra, mentre la massa è dovuta al proiettile. Il tempo di impatto è metà del periodo dell'oscillatore, non dipende quindi dalla velocità iniziale ma solo da  $k$  e  $m$  secondo la formula  $T = 2\pi\sqrt{\frac{m}{k}}$ , i risultati delle simulazioni confermano questa teoria, infatti il tempo di contatto dipende solo dalla massa del proiettile.

Il codice scritto assegna a ogni simulazione un valore di massa tra 0.2 e 0.8  $kg$  e uno di velocità tra 0.2 e 4.2  $m/s$ , scelti aleatoriamente con probabilità uniforme tramite la funzione `rng` di MATLAB per esplorare tutte le possibili combinazioni. Lo spazio di tutte le combinazioni esplorate è rappresentato nella figura [5.18](#).

### 5.2.3 Geometria

Lunghezza e larghezza della piastra sono invariate ( $150 \times 100 \text{ mm}^2$ ), lo spessore invece è stato leggermente ridotto a 4 mm. Il supporto e il proiettile sono sempre corpi rigidi e le loro dimensioni sono uguali alle precedenti.

La *stacking sequence* è  $[0\ 90\ 90\ 0]$ . Si tratta di un layup simmetrico e di tipo *cross-ply* (cioè contiene solo lamine orientate a  $0^\circ$  o  $90^\circ$ ), di conseguenza è anche bilanciato.

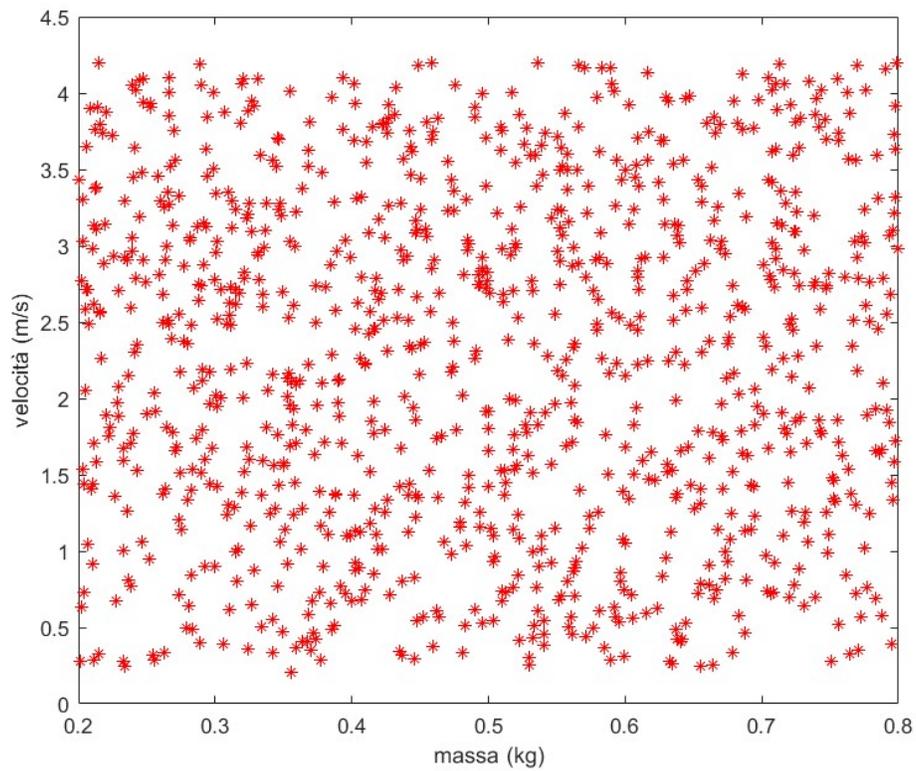


Figura 5.18: Rappresentazione grafica di tutte le combinazioni di velocità e massa utilizzate. Ad ogni punto sul grafico corrisponde una delle 1000 simulazioni effettuate

### 5.2.4 Materiale

Il materiale è lo stesso del modello di validazione: di conseguenza le proprietà meccaniche sono uguali. Analogamente le proprietà dell'interfaccia non sono cambiate, tuttavia è stata aumentato lo spessore delle lamine, cosa che consente di ridurre la rigidità della superficie coesiva a  $4 \times 10^{14}$  N/m<sup>3</sup>. Inizialmente è stato utilizzato il criterio di Hashin per il danno al materiale, esattamente come nel modello precedente, tuttavia anche nel caso più gravoso possibile (0.8 kg e 4.2 m/s) la forza di impatto non è sufficiente a causare la frattura, ma avviene solo delaminazione. Questo è coerente con un danno di tipo BVID, che infatti non è visibile dall'esterno, ed è proprio la tipologia di eventi che questo progetto vorrebbe studiare. Non avendo alcuna influenza sulla simulazione, il criterio di rottura di Hashin è stato rimosso dal modello per ridurre ulteriormente il costo computazionale.

### 5.2.5 Mesh

Poiché gli impatti avvengono sempre nello stesso punto, i requisiti sulla mesh sono meno stringenti ai bordi del modello. Sempre allo scopo di limitare i tempi di calcolo è stata quindi costruita una mesh non uniforme di  $55 \times 80$  elementi per lamina con un bias che rende più piccoli quelli centrali. Il numero totale di elementi di tipo SC8R è quindi 17600.

Come in precedenza, il supporto e il proiettile sono composti di elementi rigidi R3D4 e R3D3, il supporto è bloccato da un incastro e il proiettile è associato a una massa puntiforme. La massa del proiettile è uno dei parametri che variano tra le simulazioni (tra 0.2 e 0.8 kg).

### 5.2.6 Steps

La suddivisione in steps è molto simile al caso precedente: il primo (*initial*) serve a inizializzare il modello e impone l'incastro al supporto, il secondo (*velocity*) dura  $1 \mu s$  e impone la velocità (tra 0.2 e 4.2 m/s) al proiettile e il terzo (*vibration*) simula l'impatto.

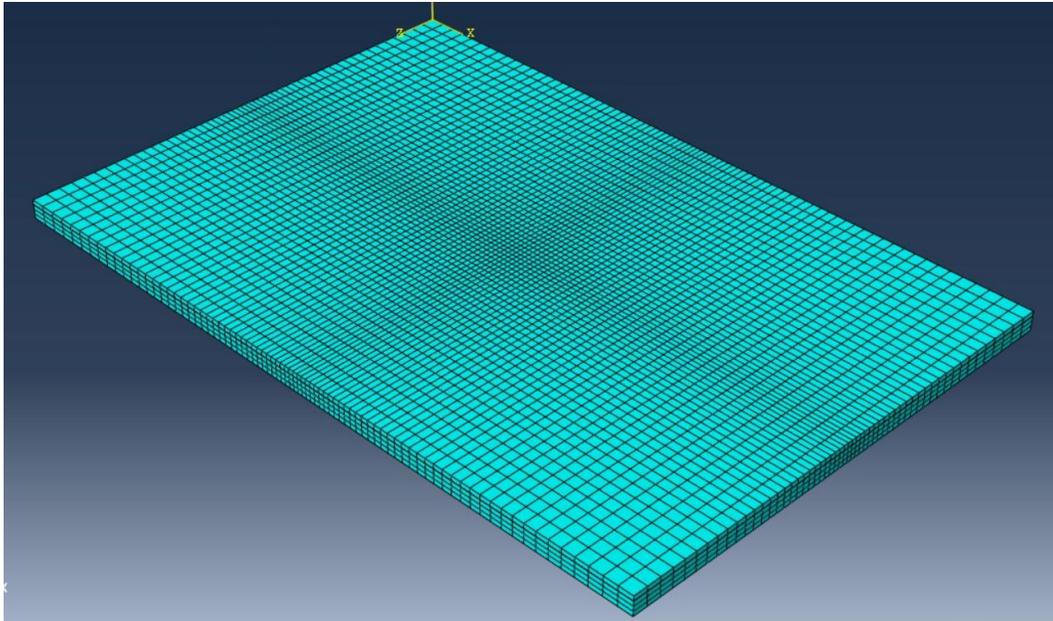


Figura 5.19: Mesh del modello di lavoro: è visibile il bias negli elementi che rende la mesh più fine vicino al punto di impatto

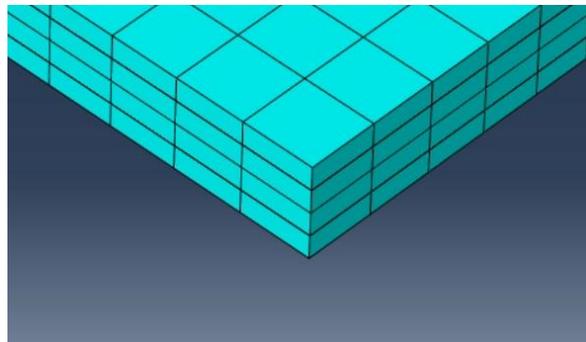


Figura 5.20: Il modello di lavoro prevede 4 elementi lungo lo spessore della piastra

Data la minore massa del proiettile e la conseguente durata più breve dell'impatto, è stato leggermente accorciato il terzo step (da 4 a 4.5 ms).

Gli output richiesti ad Abaqus (sempre durante lo step *vibration*) sono:

- Spostamenti e accelerazioni (U, A) in un sottoinsieme di 12 nodi, scelti per simulare i sensori del sistema di SHM, campionati a frequenza di 50 *kHz*.
- Criterio di iniziazione del danno interlaminare (CSQUADSCRT) su tutte le superfici coesive, richiesto solo a fine analisi.

La tabella [5.2.6](#) riassume le differenze principali tra i due modelli.

Proprietà	Modello di verifica	Modello di lavoro
Numero di lamine	16	4
Spessore delle lamine	0.26 mm	1 mm
Numero di elementi	107200	17600
Tipo di mesh	regolare	con bias
Massa del proiettile	2 kg	0.2-0.8 kg
Velocità del proiettile	2.56 o 5.43 m/s	0.2-4.2 m/s
Durata della simulazione	4.5 ms	4 ms
Tempo di calcolo	25 h	15 min

Tabella 5.4: differenze principali tra i modelli

### 5.2.7 Risultati

I dati grezzi prodotti da Abaqus sono i vettori di spostamento e accelerazione calcolati ai nodi scelti come sensori: per ogni simulazione si possono quindi estrarre tutte le componenti di spostamento e accelerazione in 12 diversi nodi (come fatto nelle figure [5.21](#) e [5.22](#)). Questo è stato fatto con l'ausilio della libreria di funzioni Abaqus2Matlab [\[19\]](#), che permette di importare facilmente la maggior parte dei dati delle simulazioni Abaqus in MATLAB.

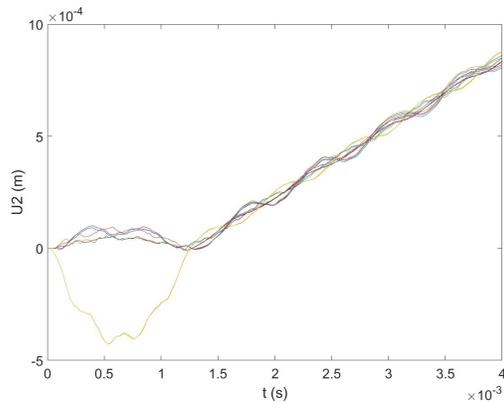


Figura 5.21: Spostamenti registrati dai sensori nella simulazione 42

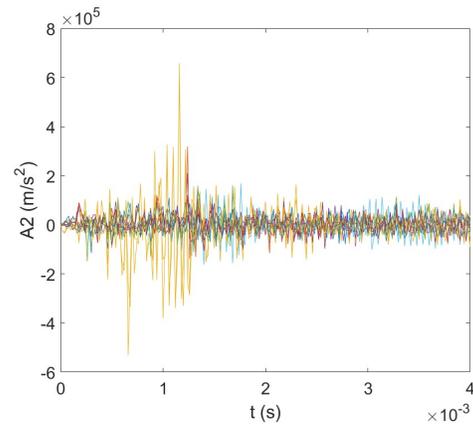


Figura 5.22: Accelerazioni registrate ai nodi nella simulazione 42

L'altro dato prodotto è il danno all'interfaccia, che però è una tipologia di output non supportata da Abaqus2Matlab. Per leggere questi risultati è stato necessario automatizzare la scrittura dei file .rpt (report di Abaqus), e scrivere un codice in grado di leggerli. Un altro codice MATLAB calcola l'involuppo delle superfici delaminate (come fatto in precedenza per il modello di verifica). Siccome la mesh di questo modello non è uniforme, è necessario modificare il codice perchè assegni un peso diverso ad ogni nodo a seconda della dimensione degli elementi adiacenti.

Scritti i codici per calcolare il valore corretto della superficie delaminata, è quindi possibile valutarla per tutte le 1000 simulazioni eseguite. Le simulazioni sono state divise in 5 classi a seconda dell'entità del danno in ordine crescente:

- Classe 1: area delaminata nulla
- Classe 2: area delaminata  $< 10 \text{ mm}^2$
- Classe 3: area delaminata  $< 50 \text{ mm}^2$
- Classe 4: area delaminata  $< 150 \text{ mm}^2$
- Classe 5: area delaminata  $> 150 \text{ mm}^2$

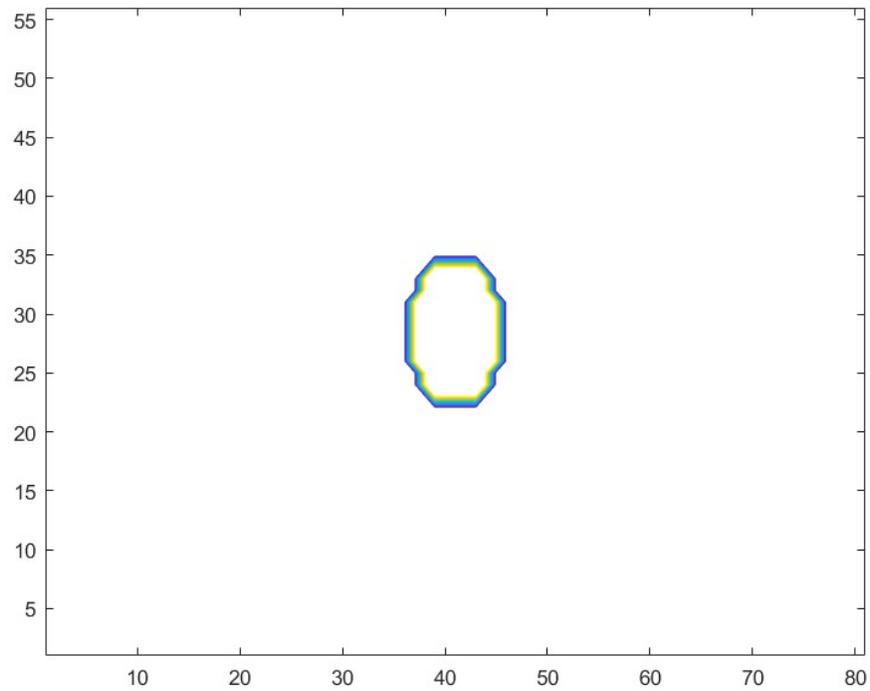


Figura 5.23: Danno all'interfaccia nella simulazione 100. Gli assi rappresentano il numero di nodi e non le dimensioni della piastra: siccome la mesh non è uniforme questo porta a sovrastimare la dimensione dell'area danneggiata (che si trova al centro, dove gli elementi sono più piccoli).

Le simulazioni sono state inserite in un grafico analogo a quello della figura [5.18](#) ma dividendo i punti tra le 5 classi appena definite. Come prevedibile, gli urti a energia più alta provocano delaminazioni più grandi, ma è interessante notare quanto è stretta la relazione tra area delaminata ed energia di impatto.

Altri studi hanno evidenziato una correlazione lineare tra energia di impatto e area danneggiata [58](#)[15](#), ed è stato provato sperimentalmente che l'entità del danno dipende solo dall'energia cinetica, e non da massa e velocità separatamente [44](#), tesi che è confermata numericamente da questo grafico. Nell'ambito dello structural health monitoring quindi, determinare l'energia di impatto o l'area delaminata sono risultati quasi equivalenti. Ad esempio uno studio di Tabian et al. [61](#) svolge un lavoro molto simile a questo in cui, basandosi su dati sperimentali, viene allenata una rete neurale per riconoscere l'energia di impatto tramite un sistema di SHM passivo. Questo studio utilizza un numero molto ridotto di dati per il training, ma riesce comunque ad ottenere degli ottimi livelli di accuratezza.

Nonostante l'energia sia un ottimo predittore del danno, questa è in generale ignota, e due impatti possono provocare risposte molto differenti nella piastra anche a parità di energia, se ottenuti con diverse combinazioni di massa e velocità. Nella figura [5.25](#) sono raffigurate le forze di contatto per due impatti alla stessa energia: provocano lo stesso danno ma i dati registrati dai sensori sono radicalmente diversi. Per allenare una rete neurale a riconoscere tutti i tipi di impatto che la piastra può subire è quindi necessario simulare impatti con diverse combinazioni di massa e velocità, e non è sufficiente variare uno solo di questi parametri.

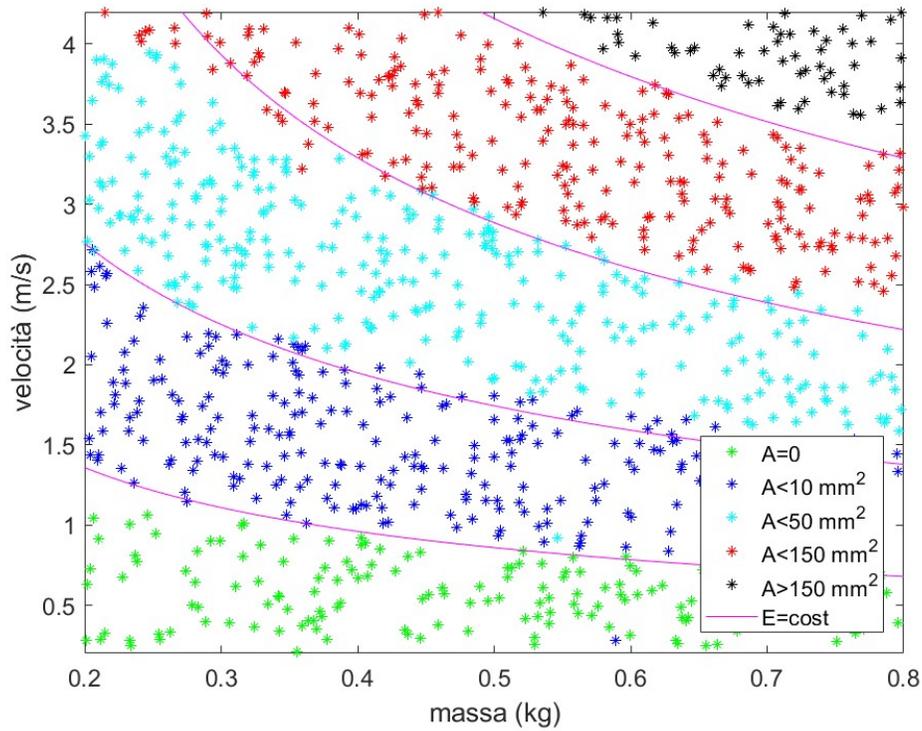


Figura 5.24: Mappa delle 1000 simulazioni svolte divise per classi in base al danno provocato. Le linee in viola sono le curve isolivello dell'energia cinetica, e dividono quasi perfettamente le classi di danno.

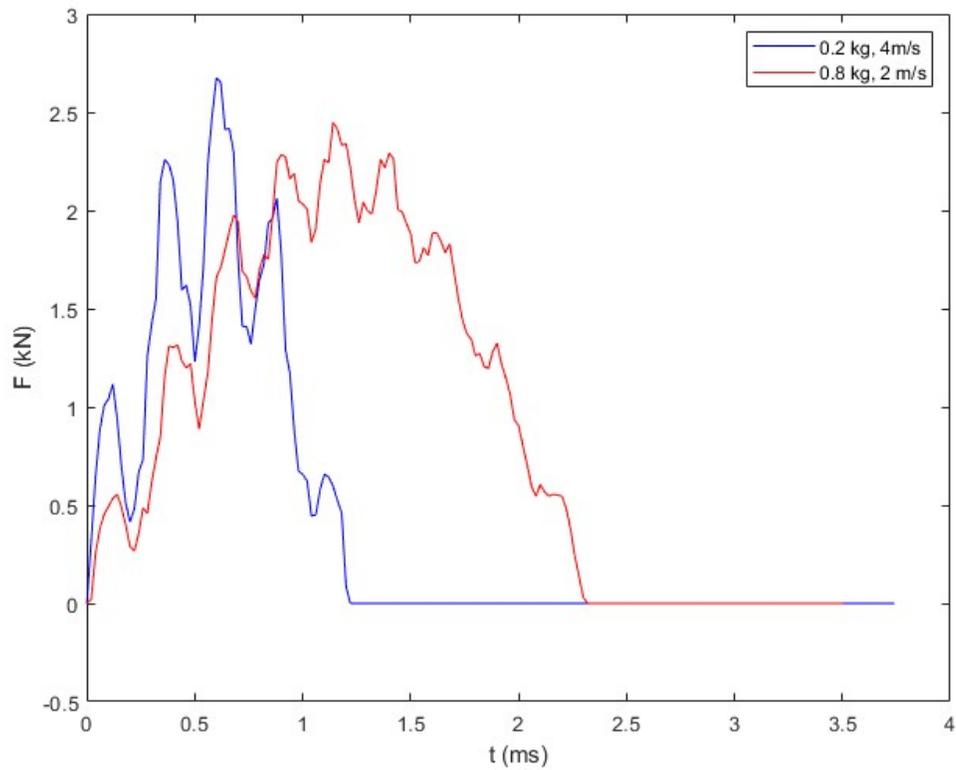


Figura 5.25: Grafico della forza di contatto tra lastra e proiettile per due simulazioni a pari energia: la superficie delaminata è la stessa, ma la risposta registrata dagli accelerometri è molto diversa

# Capitolo 6

## Risultati e discussione

Con i dati delle simulazioni sono state allenate diverse reti neurali, differenti per architettura, tipologia di input e tipologia di output. I dati sono stati divisi in modo casuale in 3 sottoinsiemi:

- training set (750 elementi)
- validation set (150 elementi)
- test set (100 elementi)

Le reti neurali sono state costruite in MATLAB grazie al *deep learning toolbox*, che contiene molte funzioni utili al training di reti neurali.

L'input consiste di 12 serie di dati (una per ogni sensore simulato), organizzate in una matrice in cui ogni colonna corrisponde a un sensore, e ogni riga ad un istante temporale. Essendo l'input bidimensionale, il primo strato della rete è di tipo `imageInputLayer`, cioè uno strato pensato per gestire immagini, non è tuttavia necessario che l'input sia formato effettivamente da immagini, poiché anche queste sono (per una rete neurale) semplicemente delle matrici in cui ogni elemento contiene la luminosità di un pixel.

Il problema si può affrontare in 2 modi: come problema di regressione o di classificazione categorica. In un problema di regressione la rete fornisce in output un singolo numero (in questo caso corrispondente all'area delaminata), mentre in un problema di

classificazione categorica la rete assegna ogni input a una classe (in questo caso le classi definite nel capitolo precedente in base all'entità del danno).

Per la classificazione categorica la variabile di interesse è l'accuratezza, ossia la percentuale di dati assegnati alla classe giusta. Scegliendo tra 5 categorie una rete non allenata assegna la classe in modo casuale e riconosce correttamente il 20% dei dati, quindi la prestazione della rete è da misurare confrontando l'accuratezza raggiunta con questo valore. Per quanto riguarda la regressione, la valutazione delle prestazioni è meno intuitiva e si basa sull'errore rms. La grandezza da calcolare è l'area di delaminazione in  $\text{mm}^2$ , il suo valore medio tra tutte le 1000 simulazioni è  $40.2 \text{ mm}^2$ , e la sua deviazione standard  $51.9 \text{ mm}^2$ .

I grafici di addestramento riportati in seguito disegnano due curve con significato molto diverso. La curva di training rappresenta l'accuratezza (o errore rms per i problemi di regressione) valutata sui casi che l'ottimizzatore utilizza per addestrare la rete. La curva di validazione rappresenta invece l'accuratezza valutata su un dataset separato (l'insieme di validazione), ed è quindi il parametro più significativo. Le reti neurali infatti, spesso soffrono di overfitting (la rete impara a riconoscere bene solo i casi su cui è stata allenata, e non riesce a generalizzare), che si manifesta come differenza tra l'accuratezza di training e quella di validazione.

Per i primi test svolti è stata usata la funzione ReLU per l'attivazione di tutti i neuroni, e l'input è costituito dall'accelerazione sui 3 assi, questi parametri sono in seguito cambiati per esplorare altre possibilità.

## 6.1 Architettura della rete

Per prima cosa è stato affrontato il problema di definizione dell'architettura: data la forma bidimensionale dell'input, una rete convoluzionale può essere la scelta giusta, sono stati quindi preparati 3 tipi di rete:

- Densa (solo *classifier*)
- Convoluzionale pura (solo *feature extractor*)

- Convolutionale completa (*classifier* e *feature extractor*)

e sono stati confrontati i risultati.

Nella costruzione di una rete neurale le possibili scelte di parametri sono praticamente infinite: è possibile cambiare il numero di strati, il numero di neuroni, la tipologia di strati inseriti (convoluzionali, densi, ricorsivi...) e la presenza di tecniche particolari di potenziamento della rete (*batch normalization*, *dropout*...). Risulta quindi impossibile esplorare tutte le combinazioni esistenti, ed è necessario ricorrere all'esperienza e alle pratiche comuni per individuare alcune architetture che più probabilmente forniranno delle buone prestazioni. Gli unici parametri che non si possono cambiare sono lo strato di input (che deve essere formato da tanti neuroni quanti sono i dati di input per ogni simulazione) e quello di output (che deve essere di un solo neurone per i problemi di regressione e di 5 neuroni per la classificazione categorica).

Esistono inoltre diversi algoritmi di ottimizzazione implementati da MATLAB, tuttavia la scelta dell'algoritmo influisce sulla velocità dell'addestramento più che sui risultati finali. Per tutte le architetture testate la durata del training non supera i 5 minuti, quindi non è stato necessario provare diversi ottimizzatori, e per tutti i casi descritti in seguito è stato utilizzato *adam*.

### 6.1.1 Reti dense

La prima tipologia studiata è la rete neurale concettualmente più semplice, in cui tutti i neuroni di uno strato sono collegati a tutti i neuroni degli strati adiacenti. Sono state costruite 3 reti di questo tipo, con numero variabile di neuroni e strati:

- Densa 1: tre strati tra input e output, rispettivamente di 32, 16 e 8 neuroni.
- Densa 2: quattro strati tra input e output, rispettivamente di 128, 64, 64 e 32 neuroni.
- Densa 3: cinque strati tra input e output, 3 di 128 neuroni e 2 di 64.

Queste reti sono molto veloci da allenare, perché contengono un numero relativamente basso di parametri, tuttavia forniscono risultati completamente fuori scala. Per le prime due, in 100 epoche, l'errore rms supera i 4000 mm<sup>2</sup>, mentre il terzo arriva a 600 mm<sup>2</sup> dopo 200 epoche. Tutti i modelli presentano un forte overfitting dalle prime iterazioni, in particolare il primo, che contiene meno parametri.

Anche nel problema della classificazione categorica le reti dense sono poco efficaci: la rete 3 (che dà le prestazioni migliori), presenta un forte overfitting, e raggiunge un'accuratezza di training del 100% dopo poche epoche, mentre quella di validazione resta sotto al 70%. Nonostante l'aumento dei parametri abbia portato a risultati migliori, l'errore medio è ancora molto lontano da un valore accettabile, è quindi necessario passare ad un'architettura differente.

### 6.1.2 Reti convoluzionali pure

Le reti convoluzionali sono caratterizzate dall'utilizzo di filtri che rilevano la presenza di alcune caratteristiche dell'input. Ogni strato convoluzionale è composto da un certo numero di filtri, che si adattano per riconoscere caratteristiche diverse.

L'architettura classica delle reti di questo tipo prevede che ad ogni strato convoluzionale segua uno strato di pooling, è inoltre comune utilizzare la *batch normalization*, che riduce il problema dell'*internal covariate shift* [11] (nelle reti composte da molti strati, i pesi dei primi strati hanno un'influenza troppo forte sulla rete complessiva).

In una rete convoluzionale è fondamentale la disposizione spaziale dei dati di input, è quindi necessario descrivere la struttura della matrice fornita alla rete. Le righe rappresentano gli istanti temporali, e le colonne le diverse serie di dati: ogni sensore raccoglie le accelerazioni sui 3 assi, i dati dello stesso sensore sono quindi raggruppati a terne (per gli assi x, y e z). La tabella 6.1.2 riassume la struttura dell'input.

Le reti convoluzionali costruite (rappresentate nelle tabelle 6.1.2 e 6.1.2) sono composte di tre strati (formati da: filtro convoluzionale, *batch normalization* e *max-pooling*). Essendo i dati organizzati in colonne raggruppate a terne, i filtri riflettono questa

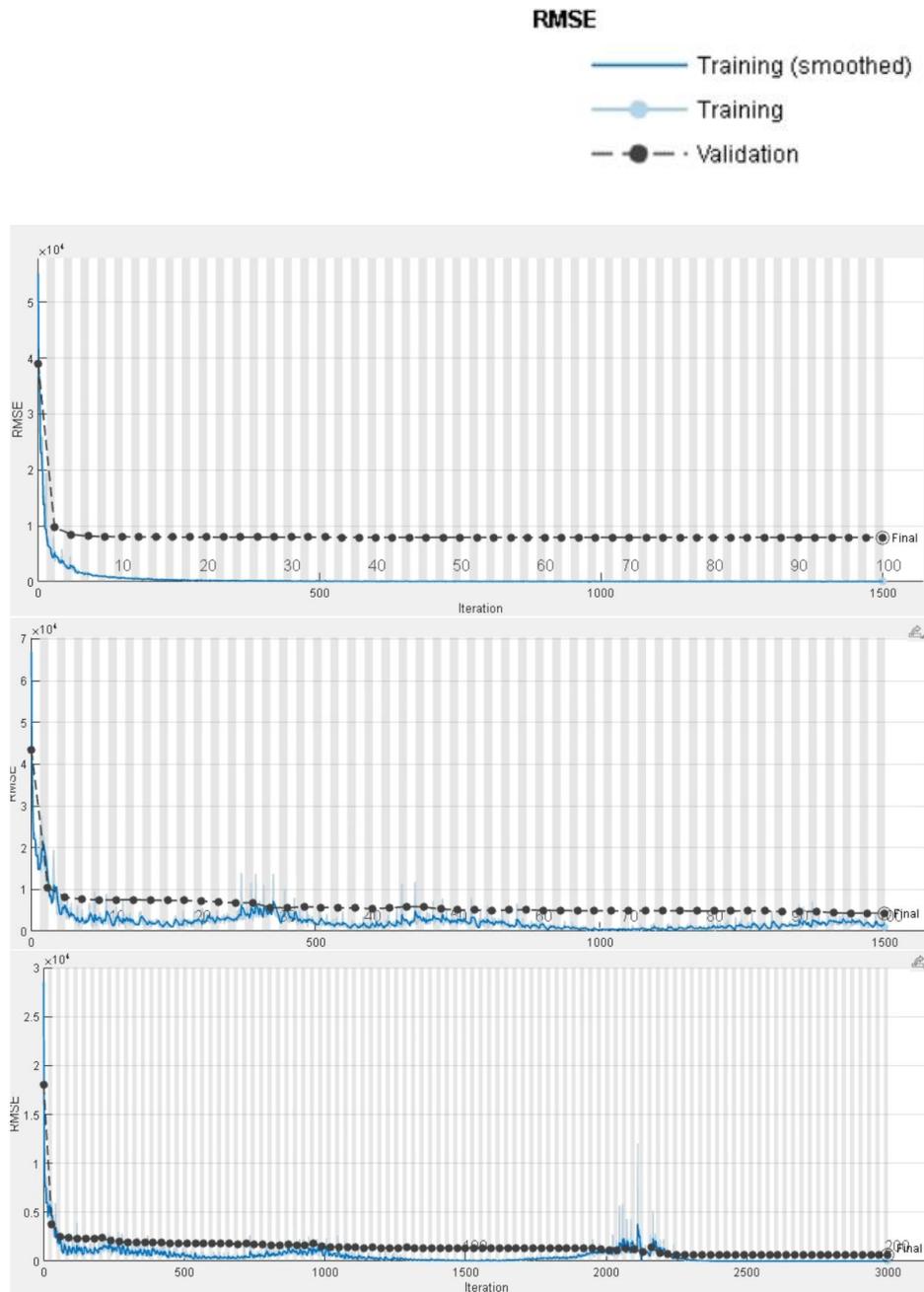


Figura 6.1: Errore rms nel tempo delle reti *Densa 1*, *Densa 2* e *Densa 3*, rispettivamente

sensore 1			sensore 2			...		
$A1(t_1)$	$A2(t_1)$	$A3(t_1)$	$A1(t_1)$	$A2(t_1)$	$A3(t_1)$	...	...	...
$A1(t_2)$	$A2(t_2)$	$A3(t_2)$	$A1(t_2)$	$A2(t_2)$	$A3(t_2)$	...	...	...
$A1(t_3)$	$A2(t_3)$	$A3(t_3)$	$A1(t_3)$	$A2(t_3)$	$A3(t_3)$	...	...	...
...	...	...	...	...	...	...	...	...

Tabella 6.1: Struttura geometrica dell'input fornito alle reti neurali

caratteristica, e sono prevalentemente verticali. Sono state predisposte due diverse architetture con filtri di dimensione diversa.

Conv 1	
strato	filtro
convolution 1	$6 \times 1$
MaxPool 1	$4 \times 1$
convolution 2	$4 \times 3$
MaxPool 2	$4 \times 3$
convolution 3	$4 \times 1$
MaxPool 3	$4 \times 1$

Tabella 6.2: Struttura della rete conv 1

Conv 2	
strato	filtro
convolution 1	$6 \times 3$
MaxPool 1	$4 \times 3$
convolution 2	$4 \times 3$
MaxPool 2	$4 \times 3$
convolution 3	$4 \times 1$
MaxPool 3	$4 \times 1$

Tabella 6.3: Struttura della rete conv 2

Anche reti composte solo da un *feature extractor*, in cui il *classifier* è formato dal solo neurone di output, possono essere efficaci, infatti i risultati sono migliori di oltre un ordine di grandezza rispetto a quelli delle reti dense. La rete Conv 1 produce un errore rms di  $10.5 \text{ mm}^2$  e la Conv 2 di  $11.9 \text{ mm}^2$  (il grafico di allenamento è mostrato

nella figura [6.2](#)) La rete Conv 1 mantiene separati gli input relativi a colonne diverse, mentre la Conv 2 utilizza filtri che coprono l'output di diversi sensori. La differenza è molto piccola, ma il primo approccio fornisce risultati migliori, si può quindi concludere che il compito di unire le informazioni dei diversi sensori dovrebbe essere lasciato al *classifier*. Applicando la rete Conv 1 al problema di classificazione categorica si ottiene un'accuratezza dell' 89%, che viene raggiunto dopo circa 5 epoche, il training successivo non migliora le prestazioni.

### 6.1.3 Rete convoluzionale completa

Come ultima alternativa si analizzano le reti convoluzionali complete: composte da un *feature extractor* di strati convoluzionali e da un *classifier* di strati densi. Le strutture convoluzionali descritte nella sezione precedente si sono già rivelate molto efficaci, le architetture successive utilizzano quindi una prima parte uguale a Conv 1, e una seconda parte variabile.

I miglioramenti nelle prestazioni sono marginali con tutte le architetture provate, ma in generale si verifica che aggiungere alcuni strati densi prima dello strato di output può essere utile a ritardare l'insorgenza dell'overfitting. Il risultato più importante di questa prima parte dell'analisi, tuttavia, è l'efficacia delle reti convoluzionali nella soluzione di questo tipo di problemi. Altri studi [\[61\]](#) [\[55\]](#) che applicano le reti convoluzionali allo SHM, convertono prima i dati in immagini, e successivamente le usano come input, è qui dimostrato che questo passaggio non è necessario, ma la rete può essere allenata in modo efficace anche con i dati di accelerazione grezzi. Nella tabella [6.1.3](#) è descritta la rete con cui sono stati ottenuti i risultati migliori.

Dopo ogni strato *convolution* è svolta un'operazione di *batch normalization*, e la funzione di attivazione è ReLU per ogni neurone.

I risultati ottenuti sono: errore rms di  $9.2 \text{ mm}^2$  per la regressione e accuratezza del 94% per la classificazione. In entrambi i casi la convergenza è rapida: dopo 20 epoche la rete è già arrivata alla massima prestazione, e un addestramento ulteriore può solo aumentare l'overfitting. I grafici di addestramento sono riportati alla figura [6.3](#)

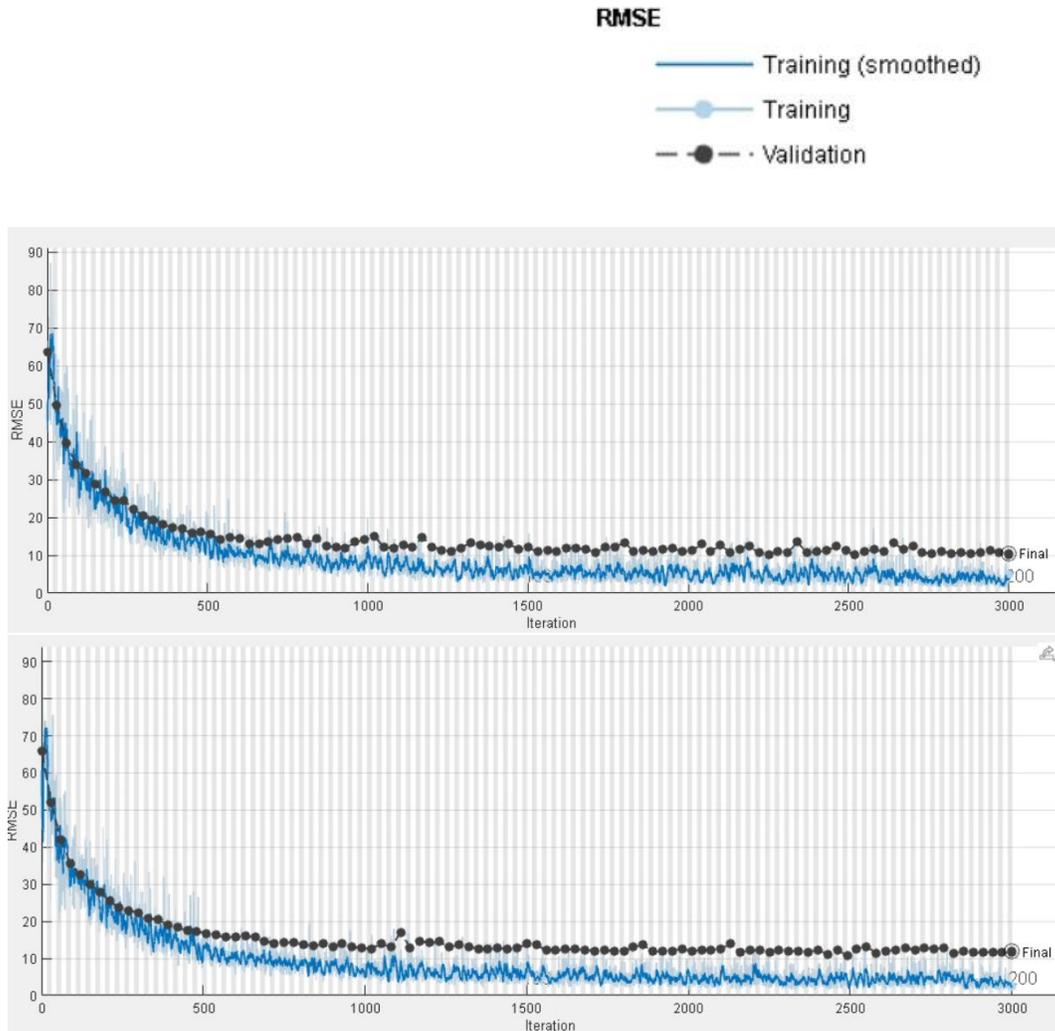


Figura 6.2: evoluzione dell'errore rispettivamente delle reti Conv 1 e Conv 2 in 200 epoche

Strato	Dimensione	Filtro
Image input	$201 \times 36 \times 1$	-
Convolution 1	$201 \times 36 \times 16$	$6 \times 1$
MaxPool 1	$50 \times 36 \times 16$	$4 \times 1$
Convolution 2	$50 \times 36 \times 32$	$4 \times 3$
MaxPool 2	$12 \times 12 \times 32$	$4 \times 3$
Convolution 3	$12 \times 12 \times 32$	$4 \times 1$
MaxPool 3	$3 \times 12 \times 32$	$4 \times 1$
Dense 1	128	-
Dense 2	128	-
Dense 3	64	-
Dense 4	32	-
Output	1 o 5	-

Tabella 6.4: Struttura della rete completa

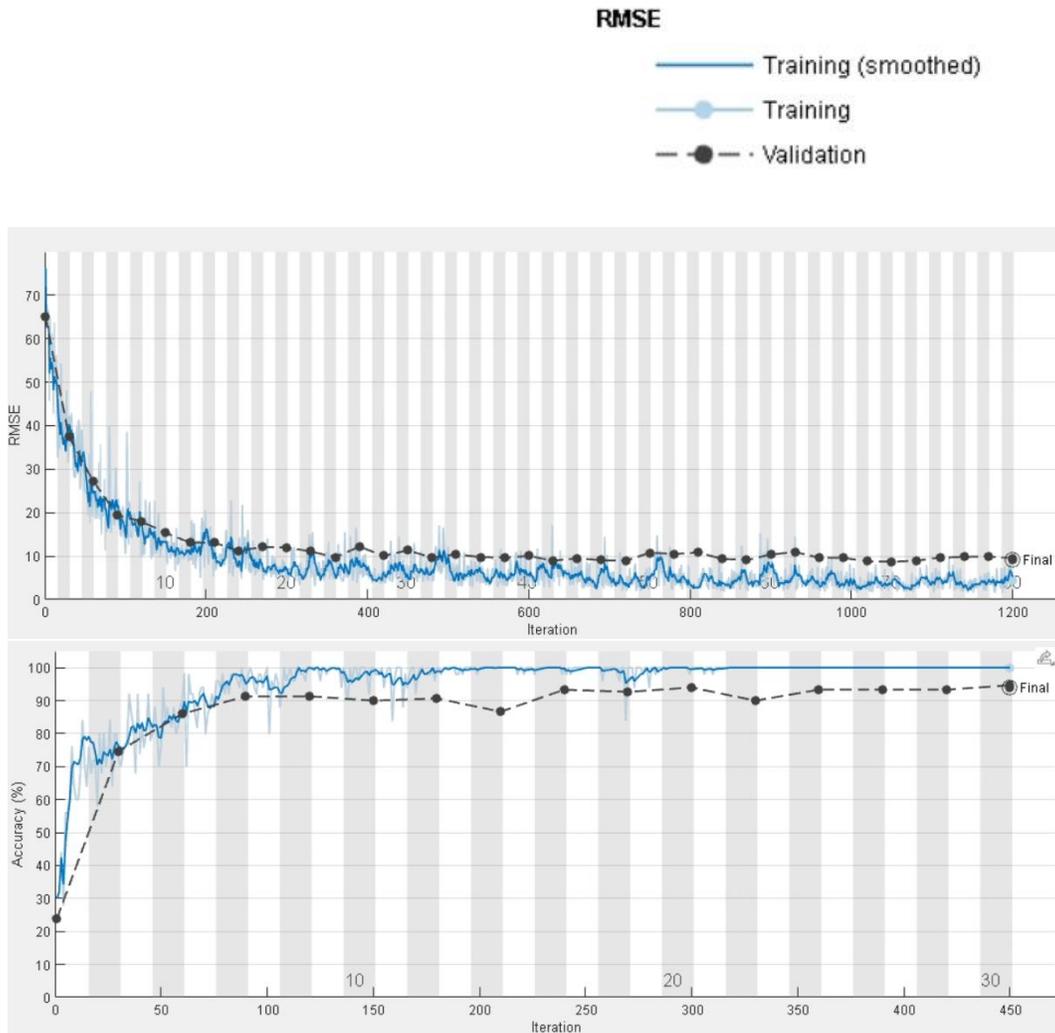


Figura 6.3: addestramento della rete completa nel caso di regressione (sopra) e categorico (sotto).

Una volta prodotta una rete categorica con prestazioni soddisfacenti è possibile scrivere la *confusion matrix*, che mostra quali classi sono riconosciute bene e dove invece si sono commessi gli errori. In una *confusion matrix* l'ordinata rappresenta le classi a cui appartengono i dati, e l'ascissa rappresenta le classi predette dalla rete, nelle celle è scritto il numero di casi per ogni combinazione di classe vera e classe predetta. I casi che la rete ha riconosciuto correttamente si trovano sulla diagonale (classe vera e classe predetta sono la stessa), e per come sono definite le classi di questo modello, la distanza dalla diagonale è una misura della gravità dell'errore. Le 5 categorie sono infatti scelte in ordine crescente per entità del danno all'interfaccia, quindi se la rete predice una categoria adiacente a quella vera, significa che l'errore è piccolo. I dati usati per calcolare la *confusion matrix* sono quelli dell'insieme di test: essendo dati diversi da quelli usati per la validazione, è normale una differenza nell'accuratezza di qualche punto percentuale. Bisogna infatti ricordare che il dataset utilizzato è relativamente piccolo (150 elementi per la validazione e 100 per il test), quindi alcune fluttuazioni aleatorie sono inevitabili quando si considera un diverso insieme di dati.

La *confusion matrix* del modello appena descritto è rappresentata nella figura [6.4](#), da cui si nota che i pochi errori commessi sono tutti adiacenti alla diagonale, quindi di piccola entità.

## 6.2 Impostazioni avanzate

Trovata un'architettura che fornisce risultati soddisfacenti, è possibile esplorare varie impostazioni per migliorarli ulteriormente. La funzione di attivazione utilizzata fino a questo punto è ReLU, ma ne esistono molte altre, e in questo paragrafo vengono esplorate per trovare quella più adatta. Inoltre, per limitare l'overfitting, si può utilizzare la tecnica del *dropout* [\[48\]](#), molto utile nei casi in cui il dataset è limitato.

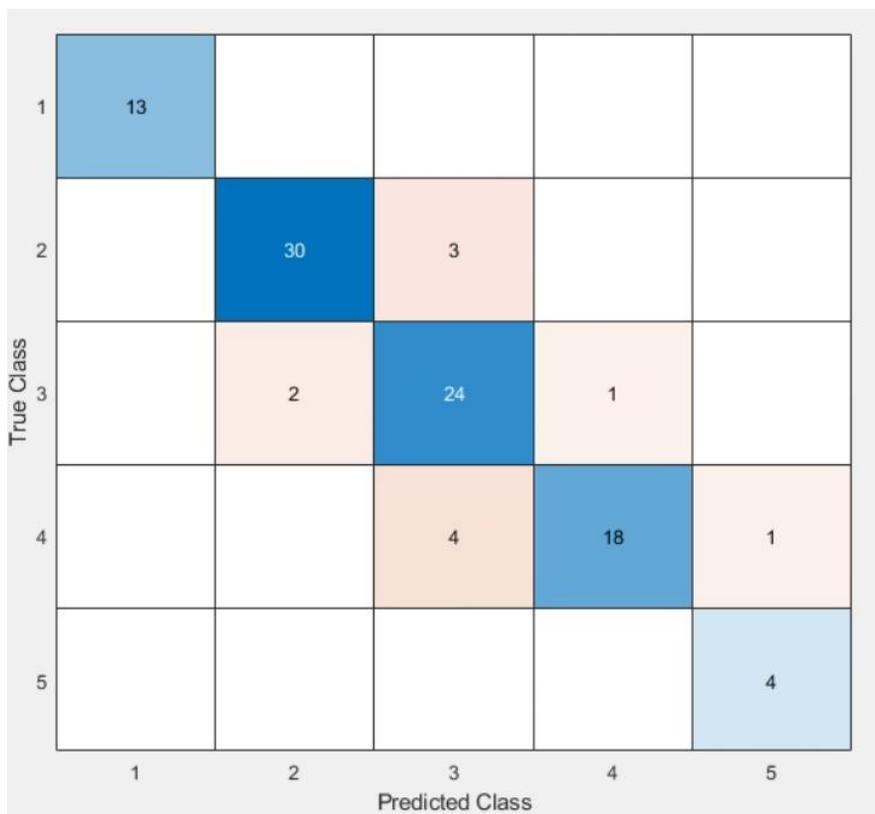


Figura 6.4: Confusion matrix del modello completo prodotto nella sezione 6.1. L'accuratezza nell'insieme di test è dell'89%

Funzione	Errore rms ( $mm^2$ )	Accuratezza
ReLU	9.2	94%
leaky ReLU	10.3	94%
sigmoid	60.3	56%
tanh	56.1	92%
ELU	8.61	91%
tanh+ReLU	7.96	93%
sigmoid+ELU	10.1	85%
tanh+leakyReLU	8.22	95%

Tabella 6.5: Risultati ottenuti con diverse funzioni di attivazione

### 6.2.1 Funzioni di attivazione

Sono stati svolti diversi test utilizzando la rete descritta nella sezione precedente (6.1.3), sia per la regressione che per la classificazione. I risultati ottenuti con le diverse funzioni di attivazione sono riportati nella tabella 6.5. Si possono fare alcune considerazioni su questi risultati:

- Escludendo le reti di sole *sigmoid* e *tanh*, i risultati sono molto simili tra loro. Bisogna considerare che l'addestramento di una rete neurale presenta una forte componente stocastica data dai pesi iniziali e dall'algoritmo di ottimizzazione. Ripetendo l'addestramento è quindi normale trovare risultati leggermente diversi anche se la rete è identica. Di conseguenza le differenze di pochi punti percentuali che si riscontrano in questi casi non sono significative, e si può concludere che la funzione di attivazione ha un'influenza limitata sulle prestazioni della rete.
- Le funzioni con output limitato (*sigmoid* e *tanh*) performano male se utilizzate in tutta la rete, particolarmente nel problema di regressione, l'errore rms è infatti molto vicino alla deviazione standard del dato da predire. La spiegazione può essere data proprio dal fatto che l'output di un neurone con queste funzioni di attivazione non può superare il valore di 1, mentre il valore da predire è alme-

no dell'ordine delle decine. Infatti utilizzando funzioni limitate solo nel *feature extractor* e funzioni diverse nel *classifier* si ottengono risultati molto simili ai precedenti.

- Una migliore prestazione nel problema di regressione non sempre si traduce in una migliore prestazione anche nel problema di classificazione, questo può essere dovuto alle variazioni stocastiche spiegate sopra o alla differenza tra i due problemi, che può favorire una particolare funzione rispetto alle altre.

Nonostante le differenze siano molto limitate, le reti ibride sembrano performare leggermente meglio, per le analisi successive si usa quindi la funzione di attivazione tanh per il *feature extractor*, e leaky ReLU per il *classifier* (corrispondente all'ultimo caso nella tabella [6.5](#)).

### 6.2.2 Dropout

Il dropout è una tecnica utilizzata nelle reti neurali per ritardare l'overfitting. Consiste nel disattivare alcuni neuroni, scelti casualmente con una probabilità definita dal programmatore, durante il training. Questo forza la rete a sviluppare dei percorsi di riconoscimento alternativi, migliorando generalmente le prestazioni della rete completa. Nelle reti convoluzionali il dropout si applica solitamente al *classifier*, per gli strati convoluzionali del *feature extractor* sono state sviluppate alcune tecniche avanzate che producono effetti simili [\[41\]](#), tuttavia non verranno esplorate in questa tesi.

Alla rete definita nella sezione precedente viene quindi aggiunto il dropout nei 4 strati densi del *classifier*. I risultati ottenuti confermano che l'aggiunta del dropout riesce effettivamente a ritardare l'overfitting, come mostrato nella figura [6.6](#), è quindi necessario aumentare il numero di epoche per migliorare i risultati. Nel caso in esame l'insieme di validazione è formato da 150 elementi: quando l'accuratezza supera il 90% significa che gli errori commessi sono poche unità, è quindi molto difficile migliorare ulteriormente i risultati. L'effetto del dropout non è quindi evidente nell'accuratezza del modello (che può aumentare solo di pochi punti percentuali), ma soprattutto dalle

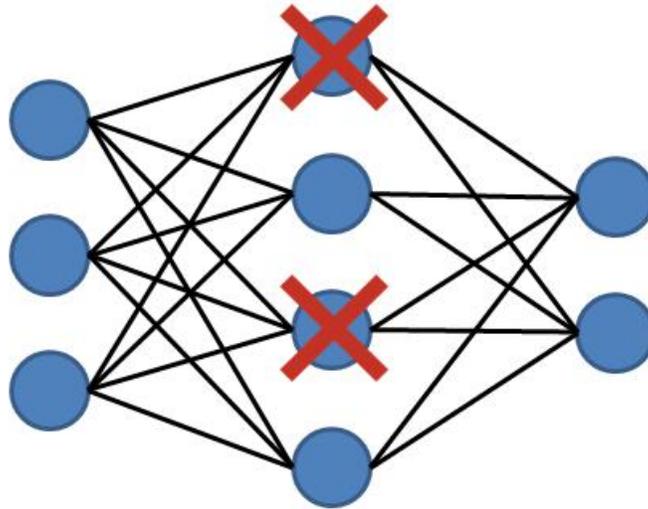


Figura 6.5: Schema di funzionamento della tecnica di dropout (tratto da [23])

curve di training. Per valutare meglio l'efficacia di queste modifiche sarebbe necessario un dataset più ampio, che è però difficile da produrre. I risultati migliori sono stati ottenuti aggiungendo una probabilità di dropout del 30% a tutti gli strati, arrivando a un'accuratezza del 96% e un errore rms di  $8.02 \text{ mm}^2$ .

## 6.3 Input

Nelle analisi precedenti è stato sempre usato lo stesso input, l'ultima parte di questa analisi si occupa di studiare l'effetto dell'utilizzo di dati di tipo diverso.

### 6.3.1 Sensori monoassiali

La rete costruita utilizza come input tutte le componenti di accelerazione registrate dai sensori, tuttavia gli accelerometri utilizzati per lo structural health monitoring solitamente hanno un solo asse sensibile, è quindi utile analizzare l'effetto sulle prestazioni

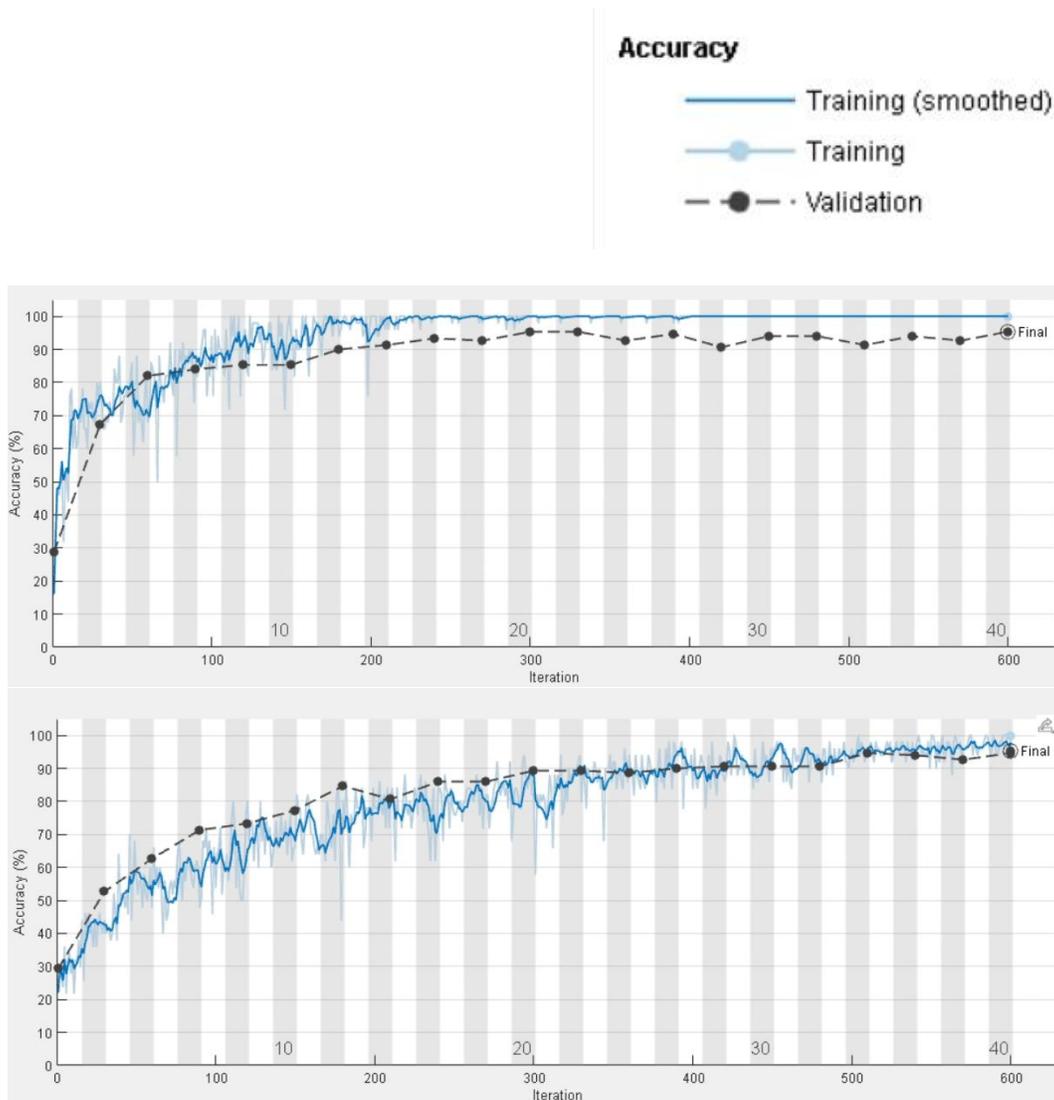


Figura 6.6: Effetto del dropout sulla curva di addestramento: senza dropout (immagine sopra) l'accuratezza di training raggiunge il 100% in poche epoche, aggiungendo una probabilità del 30% di dropout (immagine sotto) si ritarda notevolmente l'overfitting. In questo caso l'accuratezza di validazione raggiunta è la stessa, ma nel caso con dropout è possibile procedere ulteriormente con l'addestramento

dell'utilizzo di un solo asse. Il movimento della piastra avviene prevalentemente nella direzione normale (l'asse  $y$  nel modello Abaqus), che è quindi la direzione da considerare.

L'architettura della rete deve essere modificata per utilizzare dati in un formato diverso: in particolare lo strato di input deve essere di dimensione  $201 \times 12$  (rispetto al precedente  $201 \times 36$ ). L'input è analogo a quello presentato nella tabella [6.1.2](#), ma per ogni sensore è presente una sola colonna di dati (quella relativa all'asse  $y$ ). Inoltre viene rimosso il secondo strato convoluzionale, che ha lo scopo di combinare i dati appartenenti allo stesso sensore in un unico neurone. La rete complessiva è quindi composta di 2 strati convoluzionali uguali al primo e terzo del modello precedente (come rappresentato nella tabella [6.1.3](#)) con funzione di attivazione  $\tanh$ , seguiti da 4 strati densi con funzione di attivazione leaky ReLU e una probabilità di dropout del 30% (il *classifier* non è stato modificato). I risultati mostrano che la differenza nelle prestazioni rispetto al caso precedente è trascurabile: siccome la direzione dell'impatto è sempre perpendicolare alla lastra, i movimenti nelle altre direzioni sono piccoli rispetto a quelli in direzione  $y$ . Quando la rete neurale deve prevedere l'esito dell'impatto quindi, è probabile che si basi soprattutto sull'asse  $y$  anche quando vengono forniti i dati per tutti gli assi.

l'errore rms ottenuto da questa rete è di  $8.14 \text{ mm}^2$ , e l'accuratezza per la classificazione categorica è del 94%. Questi risultati sono praticamente identici a quelli precedenti (considerando le inevitabili fluttuazioni statistiche), ma sono ottenuti con un input più realistico.

### 6.3.2 Spostamenti

Altri studi simili a questo [\[54\]](#) utilizzano gli spostamenti come dati per l'addestramento della rete, questa sezione si occupa di indagare sui possibili vantaggi di questo approccio (se esistono), dato che le accelerazioni sono più facili da ottenere. La rete utilizzata è identica a quella dei casi precedenti, sono stati calcolati i risultati sia per sensori monoassiali che triassiali, i valori numerici sono riassunti nella tabella [6.6](#).

Input	Errore rms ( $mm^2$ )	Accuratezza
Accelerazioni triassiali	8.02	96%
Accelerazioni monoassiali	8.14	94%
Spostamenti triassiali	7.54	93%
Spostamenti monoassiali	5.86	95%

Tabella 6.6: Risultati ottenuti variando l'input della rete

Non ci sono differenze significative per quanto riguarda l'accuratezza della rete per la classificazione, le differenze tra i modelli sono da attribuire alle fluttuazioni statistiche dovute ad un numero limitato di elementi nell'insieme di validazione, e non alla diversa efficacia delle reti, che è difficile da valutare con un numero di errori così basso. Per quanto riguarda il problema di regressione invece si nota una differenza importante data dall'utilizzo degli spostamenti. Sarebbe quindi vantaggioso utilizzare gli spostamenti come dato, anche se sono più difficili da misurare. Come spesso accade questo è quindi un problema di compromesso tra la facilità di acquisire i dati e la precisione dell'analisi.

È inoltre confermato che la differenza tra input triassiale e monoassiale è nulla, tanto che in alcuni casi i risultati migliorano utilizzando meno dati.

Per comprendere il significato di questi dati sono utili le figure [6.7](#) e [6.8](#). Si nota che gli errori più grandi sono commessi per i casi con delaminazione più marcata, mentre i casi di delaminazione piccola o nulla sono predetti bene, quindi l'errore relativo resta limitato. Il problema di una rete di questo tipo è che non distingue tra casi di delaminazione nulla e casi di delaminazione molto piccola, distinzione che è invece molto importante in un sistema di SHM passivo. La classificazione categorica risolve questo problema creando una classe separata per gli impatti che non provocano danno, e si vede dalle *confusion matrices* (figure [6.9](#) e [6.10](#)) che le reti neurali convoluzionali riescono a distinguere molto bene i due casi.

## 6.4 Limitazioni e sviluppi futuri

Questo lavoro si basa esclusivamente su dati simulati: nonostante le simulazioni FEM siano utilizzate in numerosi ambiti in sostituzione dei test sperimentali (in quanto molto meno costose), nessuna simulazione può essere perfetta, resta quindi da determinare se una rete addestrata su dati FEM può essere utilizzata in un caso reale e fornire risultati accurati.

Un altro problema (forse il più grave) di questo approccio è che, nonostante siano comunque più economiche di un test sperimentale, le simulazioni di modelli *layered shell* richiedono tempi di calcolo molto lunghi. Per questa tesi è stato utilizzato un modello estremamente semplificato, e sono stati necessari 10 giorni di calcolo ininterrotto per ottenere una quantità sufficiente di dati, questo tempo può crescere molto se si affrontano simulazioni di parti più complesse. Per utilizzare in modo efficace questo approccio è quindi necessaria un'enorme potenza di calcolo o delle tecniche alternative per produrre i dati (metodi di *data augmentation*, molto comuni nell'ambito delle reti neurali, o nuovi algoritmi più efficienti per simulare le interfacce tra lamine).

Per il proseguimento del lavoro sarebbe utile provare ad applicare lo stesso metodo a geometrie più complesse o a piastre con una disposizione diversa delle lamine. Inoltre si potrebbe allenare una rete neurale con dati provenienti da diversi modelli, in modo che riesca a riconoscere gli aspetti comuni tra i casi di delaminazione su qualsiasi geometria. Se questo fosse possibile, risolverebbe il problema del costo computazionale in quanto servirebbe una rete per molte applicazioni invece di una rete per ogni singolo componente da monitorare.

Un'alternativa più realistica è studiare l'effetto dell'aggiunta di rumore ai dati: una rete neurale applicata ad un caso reale dovrebbe essere in grado di ignorare il rumore dovuto ai sensori, ad altre sollecitazioni o alle imperfezioni nel materiale, che non sono simulate in un modello FEM. In generale le reti neurali si sono rivelate molto efficaci in questo tipo di problemi, e anzi l'aggiunta di rumore ai dati è una tecnica che può essere usata per migliorarne le prestazioni [7]. Si può quindi essere ragionevolmente sicuri che questo approccio può portare dei risultati utili.

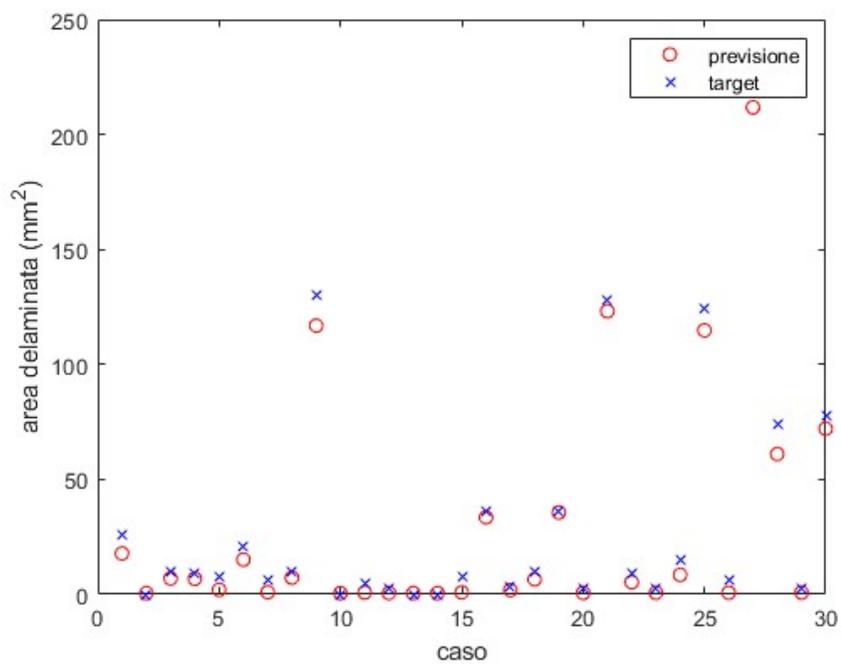


Figura 6.7: Area danneggiata predetta confrontata con quella effettiva per 30 casi da una rete che utilizza gli spostamenti come input

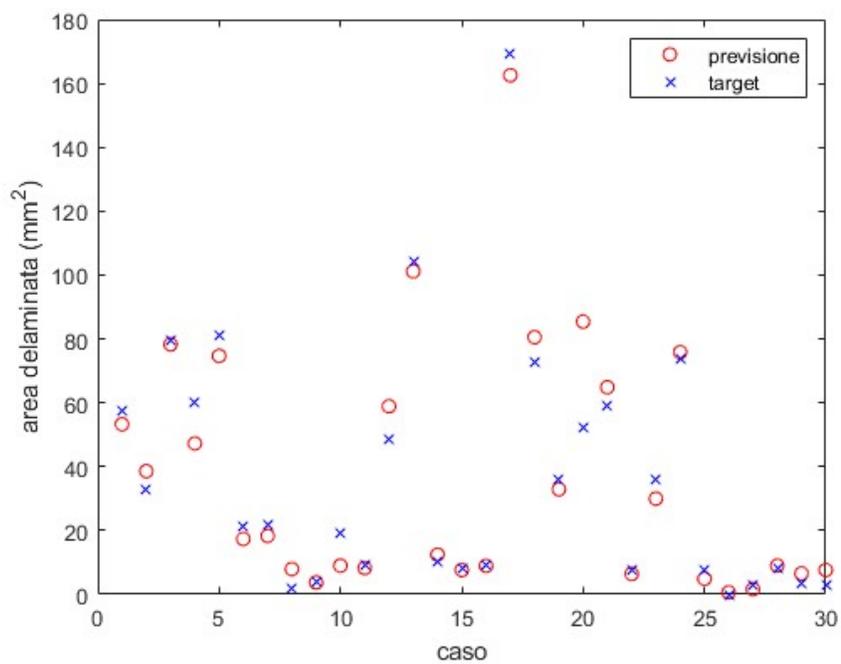


Figura 6.8: Area danneggiata predetta confrontata con quella effettiva per 30 casi da una rete che utilizza le accelerazioni come input

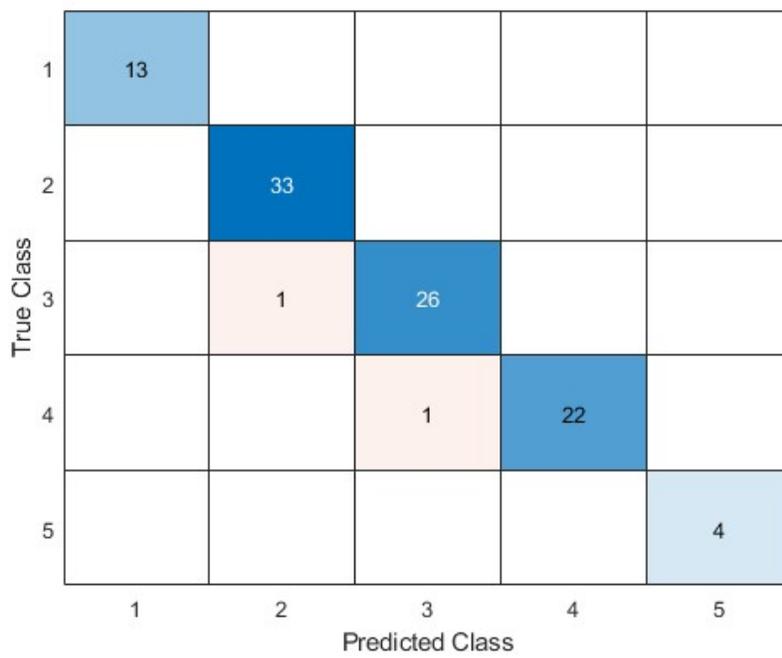


Figura 6.9: *confusion matrix* di una rete che utilizza gli spostamenti in input

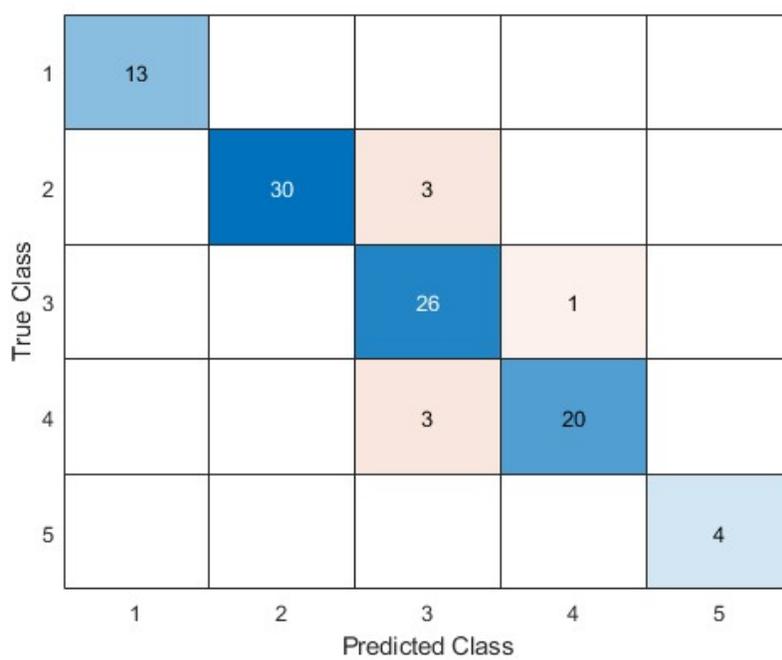


Figura 6.10: *confusion matrix* di una rete che utilizza le accelerazioni in input

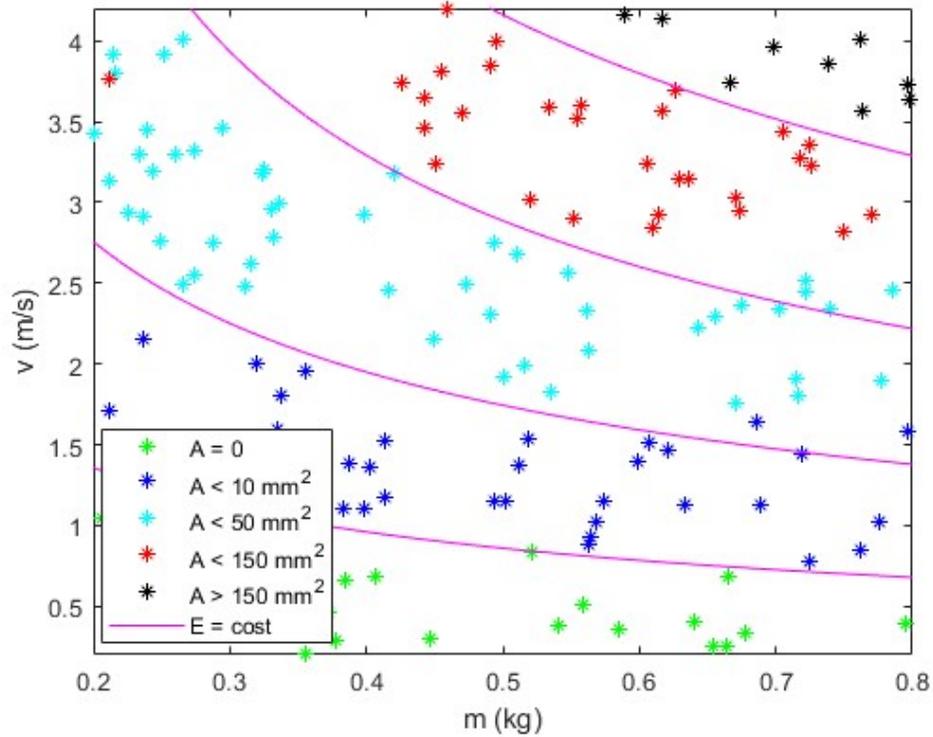


Figura 6.11: Simulazioni dell'insieme di validazione colorate in base al riconoscimento della rete neurale (basata sull'accelerazione). Si nota la suddivisione in fasce simile a quella della figura [5.24](#), si può concludere che gli errori sono commessi soprattutto vicino al confine tra le classi.

# Capitolo 7

## Conclusioni

L'obiettivo di questa tesi è di verificare le potenzialità delle reti neurali in applicazioni di structural health monitoring, in particolare nella predizione della delaminazione a seguito di impatti a bassa velocità su una lastra in materiale composito. Prima di tutto è stato costruito un modello FEM di una lastra in CFRP, è stato verificato che le proprietà del materiale utilizzate permettono di simulare in modo accurato l'effettivo comportamento sperimentale. Particolare attenzione è stata posta al modello della delaminazione, che è il fulcro di questa ricerca, e che viene simulata con le superfici coesive di Abaqus, applicando un modello di danneggiamento bilineare con opportuni criteri di iniziazione ed evoluzione del danno.

Successivamente è stato preparato un secondo modello FEM di una lastra, tramite un codice MATLAB sono state svolte 1000 simulazioni di impatti con velocità e massa del proiettile variabile per ottenere una distribuzione diversificata di casi da studiare.

Queste simulazioni hanno permesso di verificare che l'area delaminata dipende solamente dall'energia di impatto, mentre massa e velocità del proiettile non hanno effetto separatamente. I dati ricavati dalle 1000 simulazioni sono stati utilizzati (opportuna-mente divisi in insiemi di training, validazione e test) per allenare una serie di reti neurali. Sono state esplorate diverse architetture e si è concluso che le reti convoluzionali portano ad ottimi risultati analizzando degli input bidimensionali in modo simile a quanto si fa nell'ambito della *computer vision*. Si è verificato che cambiare le funzioni

di attivazione ha un effetto molto limitato sulle prestazioni della rete, mentre è utile inserire il dropout, che (soprattutto in casi come questo, dove il dataset è di dimensione limitata) ritarda l'insorgere dell'overfitting.

Sono stati studiati due diversi tipi di problema. Il problema di regressione consiste nell'addestrare la rete neurale perché preveda il valore numerico dell'area delaminata, la misura dell'efficienza di una rete di questo tipo è quindi il valore dell'errore commesso nelle predizioni, e in particolare l'errore quadratico medio. Il problema di classificazione categorica invece consiste nel dividere i dati in  $N$  classi (in questo caso 5) in base alla dimensione dell'area delaminata, e allenare la rete perché riesca ad assegnare correttamente i casi a una di queste classi.

Le reti neurali costruite si sono rivelate efficaci in entrambi i problemi. L'errore rms commesso per la regressione è stato tra 5 e 10  $\text{mm}^2$ , a fronte di un dataset in cui il valore medio è 40.2  $\text{mm}^2$  e la deviazione standard 51.9  $\text{mm}^2$ . Si può inoltre notare che la maggior parte dell'errore è dovuta ai casi in cui l'area delaminata è più grande, mentre la rete riconosce meglio i casi di danno piccolo ( $<50 \text{mm}^2$ ). Nel problema di classificazione è stata raggiunta un'accuratezza tra il 93% e il 96%, inoltre si può capire, osservando le *confusion matrices*, che gli errori commessi sono tutti di lieve entità (il caso non è assegnato alla classe corretta ma a quella adiacente).

Per l'applicazione in un sistema di SHM passivo la tipologia di rete più adatta è quella che risolve il problema di classificazione. Il risultato più utile è infatti determinare se si è verificato o meno un danno, cosa che queste reti riescono a fare molto precisamente. Le reti regressive al contrario, difficilmente distinguono tra casi in cui la risposta è zero e casi in cui è un numero molto piccolo, ma per lo stato di salute della struttura questa è una differenza fondamentale.

Come ultimo argomento è stata indagata la tipologia di input (e quindi di sensori) più adatta a questo tipo di sistemi. I dati indicano chiaramente che i sensori triassiali non sono necessari, poiché la rete utilizza quasi esclusivamente i dati nella direzione normale alla lastra, che è quella maggiormente sollecitata, per effettuare le previsioni. I risultati per reti allenate con dati triassiali o monoassiali sono infatti praticamente identici. Si evidenzia inoltre che esiste un vantaggio nell'usare dati di spostamento

piuttosto che di accelerazione, anche se gli accelerometri sono più usati nei sistemi di SHM perché ottenere dei dati di spostamento affidabile è molto più complicato.

# Bibliografia

- [1] Crivelli Visconti I. Caprino G. Langella A. *Materiali Compositi*. Hoepli, 2009.
- [2] Goodfellow I. Bengio Y. Courville A. *Deep Learning*. MIT Press, 2016.
- [3] Hiche C. Coelho C. K. Chattopadhyay A. *A Strain Amplitude-Based Algorithm for Impact Localization on Composite Laminates*. Journal of Intelligent Material Systems and Structures 22(17) (2061-2067), 2011.
- [4] Needleman A. *A Continuum model for void nucleation by inclusion debonding*. Journal of applied mechanics 54 (525-531), 1987.
- [5] Ručevskis S. Rogala T. Katunin A. *Optimal Sensor Placement for Modal-Based Health Monitoring of a Composite Structure*. Sensors 22(10) (3867), 2022.
- [6] Method AIT. *Determination of compression strength after impact*. AITM 1-0010, 2010.
- [7] Noh H. You T. Mun J. Han B. *Regularizing Deep Neural Networks by Noise: Its Interpretation and Optimization*. Advances in Neural Information Processing Systems 30, 2017.
- [8] Widrow B. *An adaptive "adaline" neuron using chemical "memistors"*. Stanford electronics laboratories, 1960.
- [9] Brilliant.org. *Brilliant math and science wiki (online)*. <https://brilliant.org/wiki/convolutional-neural-network/>, 2020.

- [10] Ferreira P. M. Machado M. A. Carvalho M. S. Vidal C. *Embedded Sensors for Structural Health Monitoring: Methodologies and Applications Review*. Sensors 22(21) (8320), 2022.
- [11] Ioffe S. Szegedy C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning 37 (448-456), 2015.
- [12] Meseix L. Soldevila Rivas A. Relandeau A. Bouvet C. *A New Method to Predict Damage to Composite Structures Using Convolutional Neural Networks*. Materials 2023, 16 (7213), 2023.
- [13] Andrianov I. KalamKarov A. L. Weichert D. *Buckling of fibers in fiber-reinforced composites*. Composites part B: engineering 43(4) (2058-2062), 2012.
- [14] Brocks W. Steglich D. *Comprehensive structural integrity - Mechanical characterization of materials*. Elsevier Science, 2007.
- [15] Hong S. Liu D. *On the relationship between impact energy and delamination area*. Experimental mechanics (115-120), 1989.
- [16] Liu Y. Fard M. Y. Chattopadhyay A. Doyle D. *Damage assessment of CFRP composites using a time-frequency approach*. Journal of intelligent material systems and structures 23(4) (397-413), 2012.
- [17] Sause M. G. R. Jasiūnienė E. *Structural health monitoring damage detection systems for aerospace*. Springer, 2009.
- [18] Megson T. H. G. *Aircraft structures for engineering students (fourth edition)*. Elsevier, 2017.
- [19] Papazafeiropoulos G. *Abaqus2Matlab*. MATLAB central file exchange, 2023.
- [20] U. S. government accountability office. *Status of FAA's Actions to Oversee the Safety of Composite Airplanes*. Report to Congressional Requesters, 2011.

- [21] Fu H. Sharif-Khodaei Z. Aliabadi M. H. *Wi-PaSS: Wireless Passive Sensing System for On-Board Structural Health Monitoring of Composite Airframes*. Conference: Composites Research Showcase 2018, 2018.
- [22] Thiene M. Ghajari M. Galvanetto U. Aliabadi M. H. *Effects of the transfer function evaluation on the impact force reconstruction with application to composite panels*. Composite structures 114 (1-9), 2014.
- [23] Dabbura I. *Coding Neural Network — Dropout*. <https://towardsdatascience.com/coding-neural-network-dropout-3095632d25ce>, 2018.
- [24] Javanbakht N. Neshastegaran A. Izadi I. *Overfitting*. <https://it.mathworks.com/discovery/overfitting.html>.
- [25] Instron. *AITM Airbus Test Method for determination of Compression Strength After Impact*. <https://www.instron.com/en/testing-solutions/other-standards/aitm-airbus-test-method>.
- [26] Bogenfeld R. Wille T. Kreikemeier J. *Review and benchmark study on the analysis of low-velocity impact on composite laminates*. Engineering Failure Analysis 86 (72–99), 2018.
- [27] Bouvet C. Castanié B. Bizeul M. Barrau J. *Low velocity impact modelling in laminate composite panels with discrete interface elements*. International Journal of Solids and Structures 46 (2809-2821), 2009.
- [28] Turon A. Dàvila C. G. Camanho P. P. Costa J. *An engineering solution for mesh size effects in the simulation of delamination using cohesive zone models*. Engineering Fracture Mechanics 74 (1665–1682), 2007.
- [29] Wei L. Chen J. *An integrated modeling of barely visible impact damage imaging of CFRP laminates using pre-modulated waves and experimental validation*. Composite structures 304 (116372), 2023.

- [30] Labeas G. Fotopoulos K. *Interlaminar Stresses Calculation Using a Stacked-Shell Finite Element Modeling Approach*. International journal of applied mechanics 7(5), 2015.
- [31] Melcher K. *A Friendly Introduction to [Deep] Neural Networks*. <https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks>, 2021.
- [32] Wronkowicz-Katunin A. Katunin A. Dragan K. *Reconstruction of Barely Visible Impact Damage in Composite Structures Based on Non-Destructive Evaluation Results*. Sensors 19(21) (4629), 2017.
- [33] Aly N. M. *A review on utilization of textile composites in transportation towards sustainability*. IOP Conference Series Materials Science and Engineering 254(4) (042002), 2017.
- [34] Benzeggagh M. L. Kenane M. *Failure analysis of CFRP laminates subjected to compression after impact: FE simulation using discrete interface elements*. Composites: science and technology 56 (439-449), 1996.
- [35] Katunin A. Wronkowicz-Katunin A. Danek W. Wylezół M. *Modeling of a realistic barely visible impact damage in composite structures based on NDT techniques and numerical simulations*. Composite structures 267 (113889), 2021.
- [36] Morton A. Reed P. Scott A. Sinclair I. Spearing M. *Quantification of Carbon Fibre/Epoxy Resin Composite Failure Processes using Synchrotron Radiation Computed Tomography*. ECCM15 - 15th european conference on composite materials, 2012.
- [37] Sharif-Khodaei Z. Aliabadi M. H. Ghajari M. *Determination of impact location on composite stiffened panels*. Smart materials and structures, 2012.

- [38] Tan W. Falzon B. G. Chiu L. N. S. Price M. *Predicting low velocity impact damage and Compression-After-Impact (CAI) behaviour of composite laminates*. Composites part A 71 (212-226), 2015.
- [39] Mathworks.com. *A Novel Joint Adversarial Domain Adaptation Method for Rotary Machine Fault Diagnosis under Different Working Conditions*. Sensors 22(22):9007, 2022.
- [40] Bekas D. G. Sharif-Khodaei Z. Ferri Aliabadi M.H. *An Innovative Diagnostic Film for Structural Health Monitoring of Metallic and Composite Structures*. Sensors 18(7) (2084), 2018.
- [41] Park S. Kwak N. *Analysis on the Dropout Effect in Convolutional Neural Networks*. Asian conference on computer vision, 2016.
- [42] Rivallant S. Bouvet C. Hongkarnjanakul N. *Failure analysis of CFRP laminates subjected to compression after impact: FE simulation using discrete interface elements*. Composites part A 55 (83-93), 2013.
- [43] Khalili S. M. R. Soroush M. Davar A. Rahmani O. *Finite element modeling of low-velocity impact on laminated composite plates and cylindrical shells*. Composite structures 93 (1363-1375), 2011.
- [44] Robinson P. Davies G. A. O. *On the relationship between impact energy and delamination area*. International journal of impact engineering 12(2) (189-207), 1992.
- [45] Sun H. Büyüköztürk O. *Optimal sensor placement in structural health monitoring using discrete optimization*. Smart materials and structures 24 (125034), 2015.
- [46] Ravindran . R. Ladani R. B. Kinloch A. J. Wang C. Mouritz A. P. *Improving the delamination resistance and impact damage tolerance of carbon fibre-epoxy composites using multi-scale fibre toughenings*. Composites part A: applied science and manufacturing 150 (106624), 2021.

- [47] Olsson R. *Composite mechanics and laminate theory*. Department of aeronautics, Imperial College, 2016.
- [48] Srivastava N. Hinton G. Krizhevsky A. Sutskever I. Salakhutdinov R. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Journal of Machine Learning Research 56 (1929-1958), 2014.
- [49] Hongkarnjanakul N. Bouvet C. Rivallant S. *Validation of low velocity impact modelling on different stacking sequences of CFRP laminates and influence of fibre failure*. Composite structures 106 (549-559), 2013.
- [50] Sikarwar S. *Nanocomposite Material For Packaging of Electronic Goods*. International Journal of Scientific and Innovative Research 1(2) (93-108), 2013.
- [51] Singh S. *Cousins of artificial intelligence*. <https://towardsdatascience.com/cousins-of-artificial-intelligence-dda4edc27b55>, 2018.
- [52] Dassault Systemes. *ABAQUS documentation*.
- [53] Hassani S. Dackermann U. *A Systematic Review of Optimization Algorithms for Structural Health Monitoring and Optimal Sensor Placement*. Sensors 23(6) (3293), 2023.
- [54] Sarego G. Cappellini L. Zaccariotto M. Galvanetto U. *Impact force reconstruction in composite panels*. 2nd International Conference on Structural Integrity, 2017.
- [55] De Oliveira M. D. Monteiro A. V. Filho J. V. *A New Structural Health Monitoring Strategy Based on PZT Sensors and Convolutional Neural Network*. Sensors 18(9) (2955), 2018.
- [56] Phadnis V. A. Roy A. Silberschmidt V. V. *Finite element analysis of drilling in carbon fiber reinforced polymer composites*. Journal of Physics: Conference Series 382 (012014), 2012.

- [57] Ungsuwarungsri T. Knauss W.G. *The role of damage-softened material behavior in the fracture of composites and adhesives*. International journal of fracture 35 (221-241), 1987.
- [58] Giannopoulos I. K. Theotokoglou E. E. Zhang X. *Impact damage and CAI strength of a woven CFRP material with fire retardant properties*. Composites part B: engineering 91, 2016.
- [59] Zhao X. Shao F. Zhang Y. *A Novel Joint Adversarial Domain Adaptation Method for Rotary Machine Fault Diagnosis under Different Working Conditions*. Sensors 22(22):9007, 2022.
- [60] Hashin Z. *Failure Criteria for Unidirectional Fiber Composites*. Journal of Applied Mechanics 47, (329–334), 1980.
- [61] Tabian I. Fu H. Sharif-Khodaei Z. *A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures*. Sensors 19 (4933), 2019.
- [62] Ławryńczuk M. *Input convex neural networks in nonlinear predictive control: A multi-model approach*. Neurocomputing 513 (273-293), 2022.