

POLITECNICO DI TORINO

---

INGEGNERIA AEROSPAZIALE

Corso di Laurea in Ingegneria aerospaziale

Tesi di Laurea Magistrale

**Scelta di modelli strutturali con tecniche  
"Machine Learning"**



**POLITECNICO  
DI TORINO**

**Relatore**  
prof. Erasmo Carrera

**Correlatore:**  
dott. Marco Petrolo

**Laureando**  
Ilario CAVALLERA  
matricola: 255344

---

ANNO ACCADEMICO 2018 – 2019



*Alla mia famiglia  
A Daniela*

# Sommario

L'obiettivo della tesi è quello di andare a ricercare un metodo per scegliere, tra i molti modelli strutturali presenti ad oggi, quello che soddisfa maggiormente le esigenze di un dato problema. Per riuscire ad ottenere questo tipo di obiettivo si va a sfruttare un'estensione della Carrera Unified Formulation (CUF) ovvero la Best Theory Diagram, (BTD). Quest'ultima teoria viene combinata con la tecnologia del Machine Learning e delle reti neurali per poter dare vita ad un sistema automatico di scelta di modelli strutturali. La prima fase verterà sulla capacità della rete di prevedere un errore dato un modello strutturale in input.

Nello specifico, la tesi, è suddivisa in vari capitoli in cui a poco a poco si vanno ad inserire parametri nuovi che rendano la rete neurale sempre più versatile. In particolare vi è una prima analisi della rete neurale in cui si va a testare come i modelli strutturali di riferimento e le singole frequenze influenzino i risultati. In seguito si vanno ad aumentare le informazioni che è possibile inserire in input e, quindi, si va ad allargare il campo di applicazione della rete neurale.

Proseguendo con gli sviluppi si va a formulare un codice che sia in grado, con una singola analisi, di fornire più risultati contemporaneamente andando così ad abbattere i costi computazionali dovuti alla moltitudine di iterazioni che tali risultati impongono nelle analisi tradizionali.

L'ultimo problema che si affronta è quello "inverso" a quello analizzato sino a questo punto. Infatti, ciò che ci si pone come obiettivo finale è di andare a richiedere un modello strutturale a partire da un errore arbitrario scelto dall'utente. Questo tipo di applicazione è molto interessante in quanto, nella pratica, non tutti i problemi (sia strutturali che non) richiedono lo stesso livello di precisione di analisi. Attraverso questa rete neurale è possibile, in prima approssimazione, capire quali sono i termini che è possibile scartare per avere il livello di precisione richiesto.

# Ringraziamenti

Ringrazio il Professor Erasmo Carrera per la possibilità concessami nel poter lavorare ad un progetto così innovativo.

Ringrazio il Dottor Marco Petrolo per avermi seguito ed aiutato molto in ogni momento durante questo percorso, permettendomi di apprendere tanto e di crescere dal punto di vista professionale.

Ringrazio mia mamma e mio papà, che da sempre sono il mio punto di riferimento e ai quali devo tutto ciò che sono. Li ringrazio per la loro fiducia, per le possibilità che mi hanno dato e per non avermi mai fatto mancare niente. Ringrazio mia sorella Tamara per non avermi fatto smettere quando molti mi avrebbero consigliato di farlo.

Ringrazio la mia fidanzata, Daniela, che in questi due anni mi è sempre stata vicina e grazie alla quale sono migliorato come persona. Grazie infinite per avermi supportato fino all'ultimo giorno e per avermi voluto bene come non aveva mai fatto nessuno.

Infine ringrazio Daniel, perché con lui ho imparato a studiare sotto tutti i punti di vista. Non è stato facile, ma alla fine ce l'abbiamo fatta. Ringrazio Stefano e Federico perché, insieme a Daniel, sono stati dei buoni amici.

# Indice

<b>Elenco delle tabelle</b>	<b>8</b>
<b>Elenco delle figure</b>	<b>12</b>
<b>1 Introduzione</b>	<b>14</b>
1.1 Machine Learning . . . . .	14
1.2 Modelli strutturali . . . . .	17
<b>2 Modelli strutturali e reti neurali</b>	<b>21</b>
2.1 Carrera Unified Formulation . . . . .	21
2.2 Best Theory Diagram . . . . .	25
2.3 Reti neurali . . . . .	27
2.3.1 Utilizzo della rete neurale per la determinazione della BTM . . . . .	32
2.3.2 Allenamento della rete neurale . . . . .	32
<b>I Risultati preliminari</b>	<b>37</b>
<b>3 Analisi statica e dinamica di gusci multistrato</b>	<b>39</b>
<b>4 Best Theory Diagrams per caso statico e dinamico</b>	<b>43</b>
4.1 Risultati analisi statica . . . . .	43
4.1.1 Guscio semplicemente appoggiato $0^\circ/90^\circ/0^\circ$ . . . . .	43
4.1.2 Guscio semplicemente appoggiato $0^\circ/90^\circ/0^\circ/90^\circ$ . . . . .	44
4.1.3 Guscio con lati incastrati/liberi $0^\circ/90^\circ/0^\circ/90^\circ$ . . . . .	44
4.1.4 Casi particolari . . . . .	44
4.2 Risultati analisi dinamica . . . . .	45
<b>5 Valutazione dell'errore BTM</b>	<b>63</b>
<b>6 Sandwich</b>	<b>75</b>
<b>II Risultati</b>	<b>87</b>
<b>7 Analisi rete neurale</b>	<b>89</b>
7.1 Variazione del Modello di riferimento . . . . .	89
7.2 Influenza singola frequenza . . . . .	90

<b>8</b>	<b>Influenza dello spessore</b>	<b>93</b>
8.1	Aumento dei dati in input . . . . .	95
<b>9</b>	<b>Rete multi target</b>	<b>101</b>
9.1	Best Theory Diagram e modi di vibrare . . . . .	101
9.2	Analisi dinamica e statica . . . . .	104
<b>10</b>	<b>Problema inverso</b>	<b>113</b>
10.1	Singolo input . . . . .	113
10.1.1	Problema inverso con base FSDT . . . . .	116
10.1.2	Errori arbitrari in input . . . . .	117
10.2	Multi input . . . . .	120
10.2.1	Errore sui modi di vibrare . . . . .	120
10.2.2	Input dinamico e statico . . . . .	121
<b>11</b>	<b>Conclusioni</b>	<b>129</b>

# Elenco delle tabelle

2.1	Esempio tabella utilizzata per MAAA, nello specifico per la ricerca di una curva BTD . . . . .	26
3.1	$0/90/0$ , $\bar{u}_z(z=0) = 100u_z E_T h^3 / (\bar{p}_z a^4)$ . . . . .	40
3.2	$0/90/0/90$ , $\bar{u}_z(z=0) = 100u_z E_T h^3 / (\bar{p}_z a^4)$ . . . . .	40
3.3	$0/90/0/90$ , $\bar{u}_z(z=0) = 100u_z E_T h^3 / (\bar{p}_z a^4)$ . . . . .	40
3.4	$E_L/E_T=25$ , $G_{LT}/E_T=0.5$ , $G_{TT}/E_T=0.2$ , $\nu_{LT}=\nu_{TT}=0.25$ , $a/h=10$ e $m=1$ , $n=1$ . Mesh 16x16. Laminazione $0^\circ/90^\circ/0^\circ$ . . . . .	41
3.5	$E_L/E_T=25$ , $G_{LT}/E_T=0.5$ , $G_{TT}/E_T=0.2$ , $\nu_{LT}=\nu_{TT}=0.25$ , $a/h=10$ e $m=1$ , $n=1$ . Mesh 16x16. Laminazione $0^\circ/90^\circ/0^\circ$ , Vincolato/libero/vincolato/libero . . . . .	41
4.1	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ$ , $R = 5$ e $h = 0.01$ . . . . .	48
4.2	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ$ , $R = 5$ e $h = 0.1$ . . . . .	48
4.3	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ$ , $R = 5$ e $h = 0.2$ . . . . .	48
4.4	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.01$ . . . . .	48
4.5	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.1$ . . . . .	54
4.6	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.2$ . . . . .	54
4.7	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 50$ e $h = 0.1$ . . . . .	54
4.8	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 100$ e $h = 0.1$ . . . . .	54
4.9	Modello BTD per guscio con 2 lati incastrati, laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.01$ . . . . .	56
4.10	Modello BTD per guscio con 2 lati incastrati, laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.1$ . . . . .	56
4.11	Modello BTD per guscio con 2 lati incastrati, laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.2$ . . . . .	56
4.12	Modello BTD per laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.2$ . Caso nel punto 1 . . . . .	56
4.13	Modello BTD per guscio con 2 lati incastrati, laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.1$ . Caso nel punto 1 . . . . .	57
4.14	Modello BTD per guscio con 2 lati incastrati, laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ , $R = 5$ e $h = 0.2$ . Caso nel punto 1 . . . . .	57
4.15	Modello BTD analisi dinamica, laminazione $0^\circ/90^\circ/0^\circ$ , $R = 5$ e $h = 0.01$ . . . . .	58
4.16	Modello BTD analisi dinamica, laminazione $0^\circ/90^\circ/0^\circ$ , $R = 5$ e $h = 0.1$ . . . . .	58
4.17	Modello BTD analisi dinamica, laminazione $0^\circ/90^\circ/0^\circ$ , $R = 5$ e $h = 0.2$ . . . . .	58
5.1	Modello BTD $h = 0.01$ , modo 1 . . . . .	67
5.2	Modello BTD $h = 0.01$ , modo 2 . . . . .	67
5.3	Modello BTD $h = 0.01$ , modo 3 . . . . .	68
5.4	Modello BTD $h = 0.01$ , modo 4 . . . . .	68
5.5	Modello BTD $h = 0.01$ , modo 5 . . . . .	68
5.6	Modello BTD $h = 0.01$ , modo 6 . . . . .	68
5.7	Modello BTD $h = 0.01$ , modo 7 . . . . .	69
5.8	Modello BTD $h = 0.01$ , modo 8 . . . . .	69

5.9	Modello BTD $h = 0.01$ , modo 9 . . . . .	69
5.10	Modello BTD $h = 0.01$ , modo 10 . . . . .	69
5.11	Modello BTD $h = 0.1$ , modo 1 . . . . .	70
5.12	Modello BTD $h = 0.1$ , modo 2 . . . . .	70
5.13	Modello BTD $h = 0.1$ , modo 3 . . . . .	70
5.14	Modello BTD $h = 0.1$ , modo 4 . . . . .	70
5.15	Modello BTD $h = 0.1$ , modo 5 . . . . .	71
5.16	Modello BTD $h = 0.1$ , modo 6 . . . . .	71
5.17	Modello BTD $h = 0.1$ , modo 7 . . . . .	71
5.18	Modello BTD $h = 0.1$ , modo 8 . . . . .	71
5.19	Modello BTD $h = 0.1$ , modo 9 . . . . .	72
5.20	Modello BTD $h = 0.1$ , modo 10 . . . . .	72
5.21	Modello BTD $h = 0.2$ , modo 1 . . . . .	72
5.22	Modello BTD $h = 0.2$ , modo 2 . . . . .	72
5.23	Modello BTD $h = 0.2$ , modo 3 . . . . .	73
5.24	Modello BTD $h = 0.2$ , modo 4 . . . . .	73
5.25	Modello BTD $h = 0.2$ , modo 5 . . . . .	73
5.26	Modello BTD $h = 0.2$ , modo 6 . . . . .	73
5.27	Modello BTD $h = 0.2$ , modo 7 . . . . .	74
5.28	Modello BTD $h = 0.2$ , modo 8 . . . . .	74
5.29	Modello BTD $h = 0.2$ , modo 9 . . . . .	74
5.30	Modello BTD $h = 0.2$ , modo 10 . . . . .	74
6.1	Frequenze adimensionali per una piastra sandwich non simmetrica ( $0^\circ/90^\circ/\text{core}/0^\circ/90^\circ$ )	76
6.2	Frequenze adimensionali per un guscio sandwich non simmetrico ( $0^\circ/90^\circ/\text{core}/0^\circ/90^\circ$ )	79
7.1	Primi 10 modi di vibrare per diversi gusci venti $R/a = 0.2$ e normalizzazione del tipo: $\omega \cdot 10\sqrt{\rho h^2/E_T}$ . . . . .	90
9.1	Modello BTD $h = 0.15$ , analisi dinamica, multi target . . . . .	110
9.2	Modello BTD $h = 0.15$ , analisi statica, multi target . . . . .	111
10.1	Tabella target . . . . .	115
10.2	Risultati per guscio $h/a = 0.01$ , usando come input l'errore sul primo modo di vibrare	116
10.3	Risultati per guscio $h/a = 0.01$ , usando come input la media degli errori dei primi dieci modi di vibrare . . . . .	116
10.4	Risultati per guscio $h/a = 0.4$ , usando come input l'errore sul primo modo di vibrare	117
10.5	Risultati per guscio $h/a = 0.4$ , usando come input la media degli errori dei primi dieci modi di vibrare . . . . .	117
10.6	Risultati per guscio $h/a = 0.4$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati . . . . .	118
10.7	Risultati per guscio $h/a = 0.2$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati . . . . .	118
10.8	Risultati per guscio $h/a = 0.2$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati. Problema risolto con rete multipla . . . . .	119
10.9	Risultati per guscio $h/a = 0.4$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati. Problema risolto con rete multipla . . . . .	119
10.10	Risultati per guscio $h/a = 0.01$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati. Problema risolto con rete multipla . . . . .	120
10.11	Tabella target con base FSDT . . . . .	120
10.12	Risultati per guscio $h/a = 0.01$ , usando come input la media degli errori dei primi dieci modi di vibrare con base FSDT. Problema risolto con rete multipla . . . . .	121
10.13	Risultati per guscio $h/a = 0.1$ , usando come input la media degli errori dei primi dieci modi di vibrare con base FSDT. Problema risolto con rete multipla . . . . .	121

10.14	Risultati per guscio $h/a = 0.2$ , usando come input la media degli errori dei primi dieci modi di vibrare con base FSDT. Problema risolto con rete multipla . . . . .	122
10.15	Risultati per guscio $h/a = 0.4$ , usando come input la media degli errori dei primi dieci modi di vibrare con base FSDT. Problema risolto con rete multipla . . . . .	122
10.16	Risultati per guscio $h/a = 0.01$ , usando come input l'errore percentuale sul primo modo di vibrare con base FSDT. Problema risolto con rete multipla . . . . .	123
10.17	Risultati per guscio $h/a = 0.1$ , usando come input l'errore percentuale sul primo modo di vibrare con base FSDT. Problema risolto con rete multipla . . . . .	123
10.18	Risultati per guscio $h/a = 0.2$ , usando come input l'errore percentuale sul primo modo di vibrare con base FSDT. Problema risolto con rete multipla . . . . .	124
10.19	Risultati per guscio $h/a = 0.4$ , usando come input l'errore percentuale sul primo modo di vibrare con base FSDT. Problema risolto con rete multipla . . . . .	124
10.20	Risultati per guscio $h/a = 0.1$ , usando come input errori arbitrari. Confronto con gli input usati per il training ed i relativi target. . . . .	124
10.21	Risultati per guscio $h/a = 0.2$ , usando come input errori arbitrari. Confronto con gli input usati per il training ed i relativi target. . . . .	124
10.22	Risultati per guscio $h/a = 0.25$ , usando come input errori arbitrari. Confronto con gli input usati per il training ed i relativi target. . . . .	124
10.23	Risultati per guscio $h/a = 0.4$ , usando come input errori arbitrari. Confronto con gli input usati per il training ed i relativi target. . . . .	125
10.24	Risultati per guscio $h/a = 0.01$ , usando come input errori arbitrari, base dei modelli FSDT. Confronto con gli input usati per il training ed i relativi target. . . . .	125
10.25	Risultati per guscio $h/a = 0.1$ , usando come input errori arbitrari, base dei modelli FSDT. Confronto con gli input usati per il training ed i relativi target. . . . .	125
10.26	Risultati per guscio $h/a = 0.2$ , usando come input errori arbitrari, base dei modelli FSDT. Confronto con gli input usati per il training ed i relativi target. . . . .	125
10.27	Risultati per guscio $h/a = 0.4$ , usando come input errori arbitrari, base dei modelli FSDT. Confronto con gli input usati per il training ed i relativi target. . . . .	125
10.28	Risultati per guscio $h/a = 0.2$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla . . . . .	126
10.29	Risultati per guscio $h/a = 0.01$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla . . . . .	126
10.30	Risultati per guscio $h/a = 0.4$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla . . . . .	126
10.31	Risultati per guscio $h/a = 0.2$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla migliorata . . . . .	126
10.32	Risultati per guscio $h/a = 0.01$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla migliorata . . . . .	126
10.33	Risultati per guscio $h/a = 0.4$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla migliorata . . . . .	127
10.34	Risultati per guscio $h/a = 0.01$ , usando come input gli errori su analisi statica e dinamica . . . . .	127
10.35	Risultati per guscio $h/a = 0.1$ , usando come input gli errori su analisi statica e dinamica . . . . .	127
10.36	Risultati per guscio $h/a = 0.15$ , usando come input gli errori su analisi statica e dinamica . . . . .	127
10.37	Risultati per guscio $h/a = 0.2$ , usando come input gli errori su analisi statica e dinamica . . . . .	127
10.38	Risultati per guscio $h/a = 0.01$ , usando come input gli errori su analisi statica e dinamica (errore medio dei primi 10 modi di vibrare) . . . . .	127

10.39	Risultati per guscio $h/a = 0.1$ , usando come input gli errori su analisi statica e dinamica (errore medio dei primi 10 modi di vibrare) . . . . .	128
10.40	Risultati per guscio $h/a = 0.15$ , usando come input gli errori su analisi statica e dinamica (errore medio dei primi 10 modi di vibrare) . . . . .	128
10.41	Risultati per guscio $h/a = 0.2$ , usando come input gli errori su analisi statica e dinamica (errore medio dei primi 10 modi di vibrare) . . . . .	128

# Elenco delle figure

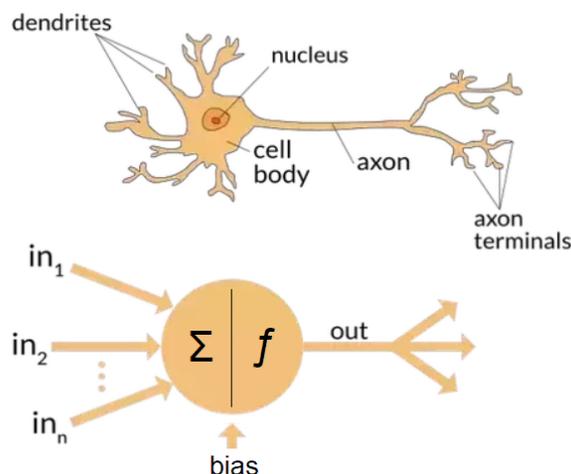
1.1	Neurone biologico vs neurone artificiale . . . . .	14
1.2	Campo di sforzi e spostamenti con ESL e LW [8] . . . . .	20
2.1	Barra a 2 nodi [8] . . . . .	22
2.2	Confronto dei campi di spostamento 1D (a sinistra) e 3D (a destra) [8] . . . . .	23
2.3	Approssimazione assiale tramite N (a sinistra) e nella sezione tramite F (a destra) [8] . . . . .	23
2.4	Schematizzazione della procedura di assemblaggio delle matrici. Si può notare un primo ciclo sugli indici $\tau$ ed $s$ per una data coppia $ij$ , una volta completato un nodo si agisce sull'elemento muovendosi per tutte le coppie $ij$ ed infine su tutti gli $N_e$ elementi. [8] . . . . .	24
2.5	Esempio curva BTD per $N = 4$ . . . . .	26
2.6	Neurone singolo input . . . . .	28
2.7	Layer di neuroni . . . . .	28
2.8	Rete . . . . .	29
2.9	Rete con tre layer . . . . .	29
2.10	Schema riassuntivo funzionamento rete neurale . . . . .	34
2.11	Curva BTD colta con un layer interno . . . . .	35
2.12	Confronto curve per diversi neuroni . . . . .	35
2.13	Confronto curve per diversi cicli di training . . . . .	36
2.14	Confronto curve per coppie input/output scelti a passi di "step" e random . . . . .	36
3.1	Sezione guscio di riferimento . . . . .	39
4.1	Tutti i risultati per $R = 5$ e laminazione $0^\circ/90^\circ/0^\circ$ . . . . .	46
4.2	BTD per $R = 5$ e laminazione $0^\circ/90^\circ/0^\circ$ . . . . .	47
4.3	BTD per $R = 2$ e laminazione $0^\circ/90^\circ/0^\circ$ . . . . .	49
4.4	Tutti i risultati per $R = 5$ e laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ . . . . .	50
4.5	BTD per $R = 5$ e laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ . . . . .	51
4.6	BTD per $R = 2$ e laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ . . . . .	52
4.7	BTD per $h = 0.1$ e laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ . . . . .	53
4.8	BTD per guscio con 2 lati incastrati, $R = 5$ e laminazione $0^\circ/90^\circ/0^\circ/90^\circ$ . . . . .	55
4.9	Coordinate casi particolari . . . . .	57
4.10	Tutti i risultati per $R = 5$ e laminazione $0^\circ/90^\circ/0^\circ$ . . . . .	59
4.11	BTD per $R = 5$ e laminazione $0^\circ/90^\circ/0^\circ$ . . . . .	60
4.12	Deviazione standard per $R = 5$ e laminazione $0^\circ/90^\circ/0^\circ$ . . . . .	61
5.1	Confronti tra i modi per $R = 5$ . . . . .	65
5.2	Particolari per modelli a pochi gradi di libertà . . . . .	66
6.1	Grafici per $a/h = 2$ . . . . .	77
6.2	Grafici per $a/h = 100$ . . . . .	78
6.3	Grafici per $h/a = 0.25$ e $R/a = 1$ . . . . .	80
6.4	Grafici per $h/a = 0.1$ e $R/a = 1$ . . . . .	81
6.5	Grafici per $h/a = 0.01$ e $R/a = 1$ . . . . .	82

6.6	Grafici per $h/a = 0.25$ e $R/a = 20$ . . . . .	83
6.7	Grafici per $h/a = 0.1$ e $R/a = 20$ . . . . .	84
6.8	Grafici per $h/a = 0.01$ e $R/a = 20$ . . . . .	85
7.1	Curve BTD ricavate utilizzando il modello LD3 come riferimento per la valutazione dell'errore . . . . .	90
7.2	Confronto curve per i modi di vibrare dal 1 al 4 . . . . .	91
7.3	Confronto curve per i modi di vibrare dal 5 al 10 . . . . .	92
8.1	Previsioni BTD per diversi spessori . . . . .	94
8.2	Curva BTD prevista per $h/a = 0.4$ utilizzando un intervallo di training che non comprende tale spessore . . . . .	94
8.3	Curva BTD prevista per $h/a = 0.1$ , con in input i valori di laminazione, condizioni al contorno e spessore. Tre strati interni con 15/10/10 neuroni . . . . .	96
8.4	Curva BTD prevista per $h/a = 0.1$ , con in input i valori di laminazione, condizioni al contorno e spessore. Tre strati interni con 15/25/25 neuroni . . . . .	96
8.5	Curva BTD prevista per $h/a = 0.05$ , con in input i valori di laminazione, condizioni al contorno e spessore. Tre strati interni con 15/25/25 neuroni . . . . .	97
8.6	Curva BTD per un guscio con caratteristiche utilizzate in fase di training . . . . .	98
8.7	Curva BTD prevista per un guscio con caratteristiche non presenti in fase di training . . . . .	99
9.1	Curva BTD prevista per $h/a = 0.15$ . . . . .	102
9.2	Frequenze previste per $h/a = 0.15$ . . . . .	102
9.3	Curva BTD prevista per $h/a = 0.1$ . . . . .	103
9.4	Frequenze previste per $h/a = 0.1$ . . . . .	103
9.5	Previsione dei primi due modi di vibrare . . . . .	104
9.6	Previsione dei primi due modi di vibrare . . . . .	105
9.7	Previsione dei primi due modi di vibrare . . . . .	106
9.8	Previsione curve BTD per guscio con $h/a = 0.01$ . . . . .	107
9.9	Risultati per guscio con $h/a = 0.01$ . . . . .	108
9.10	Previsione curve BTD per guscio con $h/a = 0.1$ . . . . .	109
9.11	Risultati per guscio con $h/a = 0.1$ . . . . .	110
9.12	Previsione curve BTD per guscio con $h/a = 0.15$ . . . . .	111
9.13	Risultati per guscio con $h/a = 0.15$ . . . . .	112

# Capitolo 1

## Introduzione

### 1.1 Machine Learning



**Figura 1.1:** Neurone biologico vs neurone artificiale

Il cervello umano possiede la rete neurale più grande e complicata che si possa studiare. È infatti vero che all'interno di ognuno di noi è sviluppata una rete con un numero proporzionale a  $10^{11}$  neuroni interconnessi tra di loro. Ognuno di questi neuroni è composto da una grande varietà di tessuti ed elementi chimici e ha la complessità propria di un microprocessore. Il nostro cervello contiene delle informazioni che ci vengono trasmesse alla nascita e con le quali conviviamo da sempre, mentre altre le impara con l'esperienza. È proprio la parola esperienza che sta alla base del Machine Learning e delle reti neurali. Infatti, le analogie tra la rete biologica ed artificiale sono molte. Si può immaginare un bambino che deve scoprire ciò che gli piace mangiare e cosa no. Egli avrà diversi input che saranno, ad esempio, cioccolato verdure e pasta, ed un target ovvero il "buono". Con l'esperienza (cioè assaggiando di volta in volta), allenerà la propria rete, imparando a categorizzare ciò che è buono da ciò che è meno buono. Su questo principio si basano le reti neurali artificiali che non hanno fatto altro che copiare e semplificare quelle biologiche perché, come spesso accade, non c'è bisogno di inventare nulla. In figura 1.1 è riportato un esempio schematico di neurone biologico confrontato con un semplice neurone artificiale. Proprio studiando le

reti neurali naturali si è scoperto che le funzioni biologiche sono immagazzinate nei neuroni e nelle connessioni che ci sono tra di essi. L'apprendimento è stato visto come uno stabilire nuove connessioni tra i neuroni oppure andare a modificare quelle che erano già esistenti. Alcune connessioni, nelle reti biologiche, si formano alla nascita e non verranno mai più modificate, altre connessioni invece si modificano o si perdono. Si vedrà come anche nelle reti neurali il numero di neuroni e connessioni tra di essi porta alla rete la capacità di svolgere funzioni più o meno complesse. Lo sviluppo delle reti neurali come è conosciuto oggi ha una storia molto lunga contrariamente a quanto si possa immaginare. Infatti, pensando alle reti neurali ci si immagina immediatamente un concetto estremamente "moderno" ed attuale, ma non è così. La storia delle reti neurali (o in generale dell'intelligenza artificiale) inizia con gli studi di *Hermann von Helmholtz*, *Ernest Mach* e *Ivan Pavlov*. Questi scienziati, tra la fine del 1800 e l'inizio del 1900 gettarono le basi su argomenti riguardanti, ad esempio, l'apprendimento e il condizionamento in linea teorica, ma senza fornire alcuna teoria matematica dietro ai concetti da loro espressi. L'attuale visione di rete neurale fu proposta negli anni '40 dallo studio di *Warren McCulloch* e *Walter Pitts* che dimostrarono come, teoricamente, una rete neurale fosse in grado di svolgere qualsiasi funzione logica o aritmetica [24]. Questo studio fu il vero punto di partenza dello sviluppo delle reti neurali. La prima applicazione di una rete neurale dovette attendere circa un decennio quando *Frank Rosenblatt* inventò un modello di training per la rete neurale che era in grado di svolgere funzioni di riconoscimento [1]. Questa applicazione suscitò grande interesse sulla materia che fu frenato dal fatto che tale modello di training funzionava solo per un numero limitato di casistiche. Vi fu successivamente un periodo di scarsa ricerca in questo settore poiché sia le ricerche di Rosenblatt che le successive di *Bernard Widrow* e *Ted Hoff* [25] vennero accantonate per via della scarsità di applicazioni in cui potevano essere utilizzate. Inoltre, in quegli anni, la potenza di calcolo dei calcolatori non era sufficiente per effettuare sperimentazione. Durante gli anni 80, però, la tecnologia fece un grande passo in avanti e quindi la potenza di calcolo delle macchine aumentò notevolmente. Insieme aumentò di molto anche la ricerca riguardo le reti neurali. Oltre all'aspetto tecnologico, quello che portò ad una svolta nell'ambito delle reti neurali, furono due nuovi concetti che vennero introdotti negli anni. Il primo concetto, scoperto da un gran numero di ricercatori, fu quello dell'algoritmo di training della rete neurale di tipo *Backpropagation* (che verrà spiegato nel dettaglio in 2.3). Lo studio di importanza maggiore relativo a tale algoritmo venne compiuto da *David Rumelhart* e *James McClelland* [23]. Tale algoritmo fu in grado di soprassedere ai limiti che furono riscontrati negli anni '50. Il secondo concetto rivoluzionario fu quello che portò ad utilizzare la meccanica statistica per spiegare il comportamento di alcune reti neurali di tipo ricorrente.

Il Machine Learning, anche detto apprendimento automatico, è una branca dell'intelligenza artificiale. Il principio fondamentale su cui basa il suo funzionamento è quello, appunto, dell'apprendimento sulla base della propria esperienza. Si considera, infatti, che un algoritmo stia apprendendo se, sulla base della propria esperienza riesce a migliorare le sue prestazioni di volta in volta. Vi sono diverse tecniche di apprendimento sulla quale si può basare un algoritmo e variano in base al tipo di supervisione alla quale è sottoposto. Ritornando all'analogia con il bambino, anche l'apprendimento può essere paragonato a quello nel mondo reale. L'algoritmo di apprendimento supervisionato può essere visto come un programma sottoposto all'attenzione di un'insegnante che monitora i risultati che lo studente sta raggiungendo. Infatti, a tale algoritmo vengono forniti un dato numero di dati in input  $X$  tutti associati ad un dato di output  $Y$ . Il programma ha quindi il compito di andare a trovare la funzione che lega input ed output:  $Y = f(X)$ . Un metodo di apprendimento non supervisionato può essere visto come uno studente autodidatta e ha come caratteristica quella di avere un certo numero di input, ma nessuno di questi associato ad un output. Il suo obiettivo è quello di trovare relazioni o schemi tra i vari dati che vengono analizzati senza utilizzare una categorizzazione. Infine, vi è il metodo di apprendimento più utilizzato ovvero quello parzialmente supervisionato. In questo caso vi sono ancora un certo numero di dati in input, ma solo alcuni di questi viene etichettato con il corrispettivo valore di output.

L'interesse per questa tecnologia è dato dalla vastità di applicazioni che può ricoprire e dal grado di affidabilità e ripetibilità che può ottenere. I campi in cui una tecnologia Machine Learning può essere impiegati spaziano dal riconoscimento vocale alla guida autonoma, dalla categorizzazione delle email spam all'impiego nell'ingegneria. In letteratura è possibile trovare diverse ricerche su temi ingegneristici.

1. Un'interessante applicazione è quella proposta da [9] in cui viene sviluppata una rete neurale per il monitoraggio dei deterioramenti all'interno di una struttura. Ciò che rende competitiva l'idea proposta nell'articolo è il fatto di allenare la rete attraverso campioni di danneggiamenti generati analiticamente. L'alternativa di generare i campioni per vie sperimentali, distruttive o non, è costosa sia dal punto di vista delle tempistiche di ottenimento dati sia dal punto di vista degli oneri nel dover costruire modelli da distruggere. Elkordy conclude dicendo che attraverso lo sviluppo di questa tecnologia ha un buon potenziale visti i risultati ottenuti e paragonati a quelli sperimentali.
2. Nel campo dell'ingegneria strutturale sono molti gli studi che si sono compiuti fino a questo punto; ad esempio in [16] viene proposta una linea guida per la progettazione ed il training di una rete neurale per poter simulare un programma di analisi strutturale e per poter ridurre il tempo necessario affinché un processo di ottimizzazione converga verso una progettazione ottimale. Questa linea guida include anche la selezione del tipo di training da utilizzare ed il numero di nodi e strati interni da inserire. Rogers ha inoltre studiato il problema dell'ottimizzazione della forma della trave per minimizzare il peso della struttura sempre soddisfacendo i vincoli strutturali.
3. Nell'ambito dell'analisi agli elementi finiti, invece, quello che si è cercato di fare è stato riuscire a trovare un algoritmo in grado di generare una mesh ottimale per lo studio del problema. È quello che viene trattato in [14] in cui viene implementata una rete neurale basata sull'algoritmo di training tipo Back Propagation. La rete viene allenata per dare in output il numero di elementi finiti della mesh attraverso i propri output che sono, appunto, il numero di triangolini che vanno a comporre la discretizzazione del dominio ed i nodi. Per rendere più rapida la ricerca dei target imposti la rete viene implementata usando un metodo di parallelizzazione con l'obiettivo di andare a rendere più veloce l'algoritmo Back Propagation.

Impossibile non citare quello che mediaticamente parlando, oltre che scientificamente, è stato l'utilizzo più discusso del 2019. Infatti, la famosa foto del buco nero è stata ottenuta anche grazie ad un algoritmo di Machine Learning che è stato in grado di correggere le imperfezioni (dovute a rumore atmosferico, rumore termico, umidità, ...) nei dati raccolti dai telescopi puntati sul buco nero. Grazie a questo tipo di tecnologia è stato possibile gestire e correggere migliaia di terabyte di dati. Senza un algoritmo Machine Learning, forse, non ci sarebbe stata questa enorme scoperta che ha portato alla conferma della teoria della relatività di Einstein proposta nel 1915. L'enorme potenzialità di questa tecnologia ancora in fase di grande ricerca e sviluppo porta i campi di applicazione a spaziare dove la fantasia dello sviluppatore può arrivare, per questo motivo in questa introduzione non è possibile riportare tutti i risultati ai quali la comunità scientifica è giunta. Si consiglia la lettura di [2] per una più ampia panoramica, ma non globale, sul punto della situazione.

## 1.2 Modelli strutturali

Il metodo di analisi agli elementi finiti (FEM) è probabilmente il metodo più comune per andare a studiare, analizzare e progettare strutture di vario genere. I programmi che basano il loro funzionamento sulle analisi FEM utilizzano elementi 3D (tridimensionali), 2D (bidimensionali) e 1D (unidimensionali), con gli ultimi due che forniscono accuratezze più basse del primo. I modelli bidimensionali usati dai comuni codici commerciali fanno riferimento alle teorie classiche dell'analisi strutturale riportate in [15], [22] e [18]. Tali teorie non hanno un'accuratezza molto alta per via delle ipotesi che vi sono alla base della formulazione stessa. Una teoria strutturale ha un certo tipo di espansione lungo la coordinata  $z$ . Il numero di espansioni che va a definire tale teoria la rende più o meno accurata e definisce anche la complessità del problema che è possibile andare a risolvere. Infatti, aumentando il numero di espansioni, aumentano anche i possibili problemi che è consentito andare a trattare. Avere molti termini di espansioni porta quindi ad avere una teoria molto accurata che, però, è molto costosa dal punto di vista computazionale. Come si vedrà più nel dettaglio nel corso della tesi, nell'analisi agli elementi finiti, l'ordine di espansione del modello va a definire il numero di gradi di libertà del problema e, quindi, la sua complessità [8]. È quindi una scelta dell'utente che decide quale teoria utilizzare decidere se scarificare accuratezza per lasciar spazio a ricerche della soluzione più rapide oppure avere un calcolo più lento, ma a fronte di risultati più complessi e/o più precisi. Per scegliere i termini di espansione che vadano a descrivere il mio problema si può ricorrere a due differenti approcci. Il primo, detto *assiomatico*, va ad introdurre le espansioni legandole a delle ipotesi fatte sulle caratteristiche meccaniche del caso in esame. Tali ipotesi rendono la soluzione meno accurata, ma permettono di avere tempi di risposta più rapidi. È questo il caso delle teorie classiche [15], [22] e [18]. Il secondo approccio è quello *asintotico*, invece, va ad introdurre delle espansioni matematiche rigorose lungo lo spessore del guscio o della piastra in esame, con accuratezza che è nota ed è riferita alla soluzione tridimensionale esatta [11], [5]. Naturalmente, come già in parte anticipato in questa introduzione, i modelli assiomatici (quindi le teorie classiche) sono più semplici di quelli asintotici, ma forniscono risultati meno accurati in un campo di applicazione più limitato. D'altro canto, però, i modelli asintotici possono avere una complessità tale da avere tempi di calcolo proibitivi e talvolta sono inutilizzabili. Vi è, infine, una sorta di terzo approccio per modellare il modello strutturale: il metodo asintotico/assiomatico (AAM) che viene utilizzato per andare a determinare la teoria BTM (Best Theory Diagram) [4]. Questo tipo di teoria tenta di andare ad identificare quali siano i termini di un'espansione a renderla accurata rispetto ad altri che sono invece meno influenti. Il lavoro di questa tesi si va proprio ad inserire nell'ambito della BTM in quanto, attraverso lo sviluppo di una rete neurale opportunamente allenata, si cercherà di ricavare dei modelli strutturali ridotti basandosi su tale teoria. Il vantaggio dell'utilizzo di una rete neurale, come si vedrà nei capitoli successivi, risiede nel poter ridurre le analisi da lanciare su un software FEM (ad esempio numero limitato di spessori della piastra) in virtù del fatto che la rete correttamente allenata sarà in grado di prevedere un vasto numero di soluzioni oltre a quelle utilizzate per il training.

Prima di entrare nello specifico del lavoro di questa tesi è utile richiamare le più comuni scelte di descrizione delle variabili per modellare le teorie per strutture multistrato, in quanto la totalità dei risultati riportati nei prossimi capitoli verterà su questo tipo di struttura. Le strutture multistrato sono comunemente definite anisotrope trasversalmente (ovvero in direzione dello spessore), in quanto ogni strato (anche chiamato layer) che le compone ha caratteristiche diverse, anche per via della diversa orientazione delle fibre. Per via di questa caratteristica anisotropa le proprietà meccaniche lungo lo spessore variano ad ogni livello e, per questo motivo, nasce un campo di spostamenti (detto comunemente  $u$ ) che può andare a cambiare rapidamente pendenza e direzione all'interfaccia tra due strati successivi. Questo fenomeno viene chiamato *zig-zag* o *ZZ*. Dal punto di vista degli sforzi è evidente che gli sforzi nel piano,  $\sigma_p = (\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$  possono essere generalmente discontinui tra uno strato e l'altro, ma per soddisfare requisiti di equilibrio lo sforzo

trasversale,  $\sigma_n = (\sigma_{xz}, \sigma_{yz}, \sigma_{zz})$  deve essere continuo anche nelle interfacce. Questo requisito viene chiamato continuità interlaminare (IC) dello sforzo trasversale. Le prime teorie furono modellate per descrivere strutture a singolo strato e materiale isotropo e possono essere convenzionalmente suddivise in due categorie:

1. teorie di Love, prima approssimazione (LFATs);
2. teorie di Love, seconda approssimazione (LASTs)

Le applicazioni della prima teoria vengono normalmente categorizzate come *teorie classiche (classic laminated theory CLT)*. Tale teoria assume che le normali alla superficie  $\Omega$  rimangono normali alla stessa nella configurazione deformata e non cambino in lunghezza. Tale assunzione implica il fatto che il taglio trasversale e lo sforzo normale siano trascurabili rispetto agli altri sforzi. Le prime estensioni delle teorie classiche, destinate alle strutture multistrato e includenti il taglio trasversale, sono conosciute come *first-order shear deformation theory FSDT*. Successivamente si raffinarono queste teorie per andare a formare le cosiddette *higher-order theories HOTs*, in cui si vanno a modellare i campi di spostamento tramite delle serie di espansioni, nel seguente modo:

$$u_i(x, y, z) = u_0 + zu_1 + z^2u_2 + \dots + z^N u_N$$

dove  $N$  è il numero che indica l'ordine di espansione e maggiore è l'ordine di espansione, maggiore è l'accuratezza dei risultati trovati.

A questo punto viene introdotta la teoria *Equivalent Single Layer (ESL)*. In tale teoria gli spostamenti di ogni strato della piastra o del guscio non cambiano tra i vari layers e possono presentare modelli a diversi ordini di espansione. Se si considera il modello ESL più semplice, ovvero quello al primo ordine di espansione, si può scrivere il campo di spostamenti come:

$$\begin{aligned} u_x &= u_{x_1} + zu_{x_2} \\ u_y &= u_{y_1} + zu_{y_2} \\ u_z &= u_{z_1} + zu_{z_2} \end{aligned} \quad (1.1)$$

In questo modello il primo termine è riferito alla superficie e prende il nome di ED1, ovvero Equivalent Single Layer per una formulazione agli spostamenti sviluppato al primo ordine di espansione in  $z$ . Se da tale modello elimino il termine lineare in  $u_z$  riottengo il modello FSDT citato in precedenza. È possibile andare a riscrivere il modello al primo ordine in forma compatta nel seguente modo:

$$\mathbf{u} = F_0 \mathbf{u}_0 + F_1 \mathbf{u}_1 = F_\tau \mathbf{u}_\tau$$

in cui si è imposto che:

$$\begin{aligned} \tau &= 0, 1 \\ F_0 &= 1 \\ F_1 &= z \\ \mathbf{u} &= (u_x, u_y, u_z) \\ \mathbf{u}_0 &= (u_{x_0}, u_{y_0}, u_{z_0}) \\ \mathbf{u}_1 &= (u_{x_1}, u_{y_1}, u_{z_1}) \end{aligned} \quad (1.2)$$

Naturalmente il modello ED1 è la base di partenza per poter andare a sviluppare teorie di ordine superiore aumentando, appunto, l'ordine di espansione della serie. In generale, quindi, i modelli EDN (con  $N$  che va ad indicare l'ordine di espansione), si possono scrivere come segue:

$$\begin{aligned} u_x &= u_{x_1} + zu_{x_2} + z^2u_{x_3} + z^3u_{x_4} + \dots + z^N u_{x_{N+1}} \\ u_y &= u_{y_1} + zu_{y_2} + z^2u_{y_3} + z^3u_{y_4} + \dots + z^N u_{y_{N+1}} \\ u_z &= u_{z_1} + zu_{z_2} + z^2u_{z_3} + z^3u_{z_4} + \dots + z^N u_{z_{N+1}} \end{aligned} \quad (1.3)$$

che, in forma compatta, diventa:

$$\mathbf{u} = F_0 \mathbf{u}_0 + F_1 \mathbf{u}_1 + \dots + F_N \mathbf{u}_N = F_\tau \mathbf{u}_\tau$$

Tuttavia i modelli visti finora, anche i più accurati, non permettono alle strutture multistrato di soddisfare la condizione di continuità tra uno strato e l'altro. Per venire incontro a tale mancanza è stata formulata la *teoria Zigzag* (il nome deriva dalla forma che assume il campo di spostamenti lungo lo spessore di una struttura multistrato, come in figura 1.2), che viene introdotta attraverso la funzione Murakami. Alla base delle teorie ZZ vi è l'idea che ogni strato possa avere il proprio modello di spostamento o modello di sforzi. Successivamente, all'interfaccia tra due strati, sono imposte condizioni di compatibilità ed equilibrio per ridurre il numero di variabili incognite. Attraverso questa funzione è possibile descrivere l'effetto zig-zag rimanendo nell'ambito Equivalent Single Layer senza dover introdurre il Layer Wise che, come si vedrà successivamente, aumenta il numero di incognite aumentando il numero di strati della piastra o del guscio in esame.

Tutte le teorie viste finora hanno come comune denominatore il fatto di considerare un numero di incognite che è indipendente dal numero di strati che compongono la struttura in esame. Se però è necessaria una risposta dettagliata di ogni singolo strato per descrivere ad esempio fenomeni locali allora è necessario utilizzare delle speciali HOTs in ognuno degli strati, parallelamente ad un aumento del numero di incognite che porta la complessità del problema ad aumentare. Infatti, in questo caso, il numero di incognite non è più indipendente dal numero di strati, ma bensì ne è proporzionale. Con la teoria *layer-wise LW* si include in maniera naturale l'effetto zigzag utilizzando le teorie CLT, FSDT o HOT ad ogni strato, considerandolo indipendente. La compatibilità degli spostamenti viene garantita attraverso l'applicazione di vincoli. Nel caso limite di un guscio o di una piastra monostrato le teorie ESL e LW risultano essere equivalenti. Se si considera un modello LW al primo ordine esso si può scrivere come:

$$\begin{aligned} u_x^k &= u_{x_1}^k + z u_{x_2}^k \\ u_y^k &= u_{y_1}^k + z u_{y_2}^k \\ u_z^k &= u_{z_1}^k + z u_{z_2}^k \end{aligned} \quad (1.4)$$

in tale formulazione l'apice  $k$  si riferisce al  $k$ -esimo strato che compone il guscio o la piastra in esame. In questo concetto si sostanzia la fondamentale differenza rispetto ai modelli ESL. Per le condizioni di compatibilità all'interfaccia si potrebbero utilizzare i polinomi di Lagrange, ma in generali vengono preferiti i polinomi di *Legendre*. L'espansione si scrive come segue:

$$\begin{aligned} u_x^k &= F_t u_{xt}^k + F_b u_{xb}^k \\ u_y^k &= F_t u_{yt}^k + F_b u_{yb}^k \\ u_z^k &= F_t u_{zt}^k + F_b u_{zb}^k \end{aligned} \quad (1.5)$$

I pedici  $t$  e  $b$  sono per riferire al *top* e al *bottom* della superficie dello strato in esame. Le funzioni  $F(\zeta_k)$ , in generale, assumono le seguenti forme:

$$F_t = \frac{P_0 + P_1}{2}; F_b = \frac{P_0 - P_1}{2}$$

Dove  $P_j(\zeta_k)$  è proprio il polinomio di Legendre di ordine  $j$  definito nell'intervallo  $-1 \leq \zeta_k \leq 1$ . I polinomi di Legendre sono definiti nel seguente modo:  $P_0 = 1, P_1 = \zeta_k, P_2 = \frac{3\zeta_k^2 - 1}{2}, P_3 = \frac{5\zeta_k^3 - 3\zeta_k}{2}, P_4 = \frac{35\zeta_k^4 - 15\zeta_k^2 + 3}{8}$ . Le funzioni di Legendre scritte in questo modo permettono di godere delle seguenti proprietà:

$$\zeta_k = \begin{cases} 1 : F_\tau = 1; F_b = 0; F_t = 0 \\ -1 : F_\tau = 0; F_b = 1; F_t = 0 \end{cases} \quad (1.6)$$

Grazie a queste proprietà non sono necessarie ulteriori equazioni per imporre le condizioni di  $C_z^0$ . A questo punto è possibile andare a scrivere la formulazione compatta del modello LW:

$$\mathbf{u}^k = F_t \mathbf{u}_t^k + F_b \mathbf{u}_b^k = F_\tau \mathbf{u}_\tau^k$$

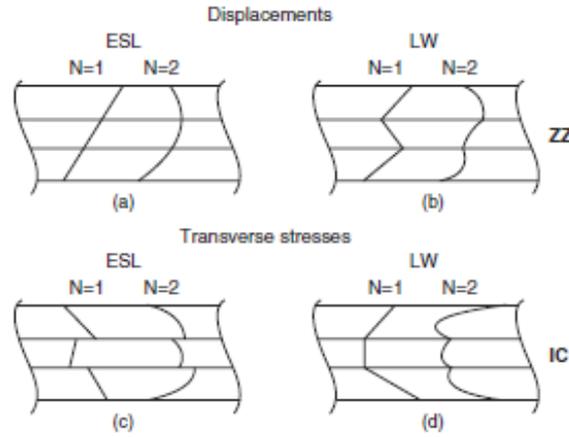
In questo caso  $\tau = t, b$ . Quella appena scritta è la teoria LD1 che, analogamente a quella ED1, si riferisce ad una teoria L (ovvero layerwise) con formulazione agli spostamenti ed al primo ordine di espansione. Anche in questo caso, naturalmente, partendo dalla teoria al primo ordine è possibile estendere le teorie LW ad ordini superiori come riportato di seguito:

$$\begin{aligned} u_x^k &= F_t u_{xt}^k + F_b u_{xb}^k + F_2 u_{x2}^k + \dots + F_N u_{xN}^k \\ u_y^k &= F_t u_{yt}^k + F_b u_{yb}^k + F_2 u_{y2}^k + \dots + F_N u_{yN}^k \\ u_z^k &= F_t u_{zt}^k + F_b u_{zb}^k + F_2 u_{z2}^k + \dots + F_N u_{zN}^k \end{aligned} \quad (1.7)$$

Per il caso generale la formulazione compatta assume la seguente forma:

$$\mathbf{u}^k = F_t \mathbf{u}_t^k + F_b \mathbf{u}_b^k + F_l \mathbf{u}_l^k = F_\tau \mathbf{u}_\tau^k$$

dove  $\tau = t, b$  e  $l = 2, 3, \dots, N$  con  $N$  ordine di espansione della serie. Per ulteriori approfondimenti



**Figura 1.2:** Campo di sforzi e spostamenti con ESL e LW [8]

consultare [8].

## Capitolo 2

# Modelli strutturali e reti neurali

### 2.1 Carrera Unified Formulation

La Carrera Unified Formulation (**CUF**) è una teoria innovativa attraverso la quale è possibile andare a scrivere le teorie unidimensionali (trave) e bidimensionali (guscio e piastra) attraverso una notazione compatta utilizzando i *Foundamental nucleus* (FN). Questi ultimi sono delle funzioni matematiche la cui forma è indipendente dalla teoria che si sta considerando ed è proprio in questa caratteristica che si trova la forza della CUF.

Per studiare la teoria si parte dalla derivazione della matrice di rigidezza per il caso di una barra (quindi caso unidimensionale). Si introducono le formulazioni per lo spostamento e la variazione virtuale:

$$\begin{aligned}u_y &= N_i(y)u_{y_i} \\ \delta u_y &= N_j(y)\delta u_{y_j}\end{aligned}\tag{2.1}$$

dove gli indici  $i$  e  $j$  indicano rispettivamente spostamento e variazione virtuale.  $N_i$  e  $N_j$  indicano invece le funzioni di forma e sono diverse per ogni caso considerato in base alla forma dell'elemento ed al numero di nodi interni. Infatti, nel caso generale, le 2.1 si scrivono nel seguente modo:

$$\begin{aligned}u_y = N_i(y)u_{y_i} &= \sum_{i=1}^{N_{NE}} N_i(y)u_{y_i} = N_1u_{y_1} + N_2u_{y_2} + \dots + N_{N_{NE}}u_{y_{N_{NE}}} \\ \delta u_y = N_j(y)\delta u_{y_j} &= \sum_{j=1}^{N_{NE}} N_j(y)\delta u_{y_j} = N_1\delta u_{y_1} + N_2\delta u_{y_2} + \dots + N_{N_{NE}}\delta u_{y_{N_{NE}}}\end{aligned}\tag{2.2}$$

dove  $NE$  rappresenta il numero di nodi presenti nell'elemento finito. Si noti che il caso più semplice di trave ad un elemento presenta due nodi. La deformazione e la relativa variazione virtuale assumono la seguente forma:

$$\begin{aligned}\epsilon &= N_{i,y}u_{y_i} \\ \delta\epsilon &= N_{j,y}\delta u_{y_j}\end{aligned}\tag{2.3}$$

dove  $N_{i,y} = dN/dy$ . La variazione del lavoro interno risulta essere:

$$\begin{aligned}\delta L_{int} &= \int_V \delta\epsilon^T \sigma dV \\ &= \int_V \delta\epsilon^T E\epsilon dV \\ &= \delta u_{y_j} \left( \int_V N_{j,y} E N_{i,y} dV \right) u_{y_i} \\ &= \delta u_{y_j} k^{ij} u_{y_i}\end{aligned}\tag{2.4}$$

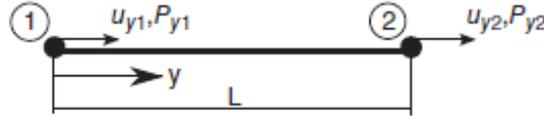
dove è stato possibile ricavare il nucleo fondamentale della barra, esso è:

$$k^{ij} = \int_V N_{j,y} E N_{i,y} dV$$

Il nucleo fondamentale non varia nè al variare del numero dei nodi  $N_N E$ , nè al variare delle funzioni di forma  $N_i$  e  $N_j$ . A questo punto è possibile andare ad esplicitare il nucleo fondamentale per il caso più semplice di barra con solamente due nodi così da poter ricavare la matrice di rigidezza per il problema in esame:

$$\begin{aligned} k^{11} &= \int_V N_{1,y} E N_{1,y} dV = \frac{1}{L} E \frac{1}{L} AL = \frac{EA}{L} \\ k^{12} &= \int_V N_{2,y} E N_{1,y} dV = -\frac{1}{L} E \frac{1}{L} AL = -\frac{EA}{L} \\ k^{21} &= \int_V N_{1,y} E N_{2,y} dV = -\frac{1}{L} E \frac{1}{L} AL = -\frac{EA}{L} \\ k^{22} &= \int_V N_{2,y} E N_{2,y} dV = \frac{1}{L} E \frac{1}{L} AL = \frac{EA}{L} \end{aligned} \quad (2.5)$$

In figura 2.1 è riportato il caso dell'esempio in esame. La matrice di rigidezza è a questo punto



**Figura 2.1:** Barra a 2 nodi [8]

facilmente determinabile utilizzando i nuclei fondamentali ed è uguale a:

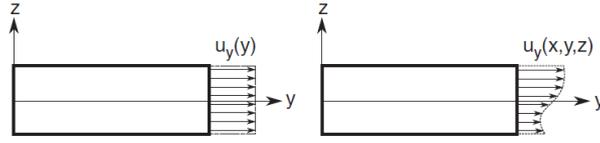
$$K = \begin{bmatrix} k^{11} & k^{12} \\ k^{21} & k^{22} \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.6)$$

Il risultato che si è ottenuto è lo stesso che si otterrebbe applicando la teoria classica. Finora è stata considerata una barra rigida senza nodi interni. Si può tuttavia ricavare, senza difficoltà aggiuntive, la matrice di rigidezza per qualsiasi tipo di configurazione. Infatti, si possono aggiungere quanti nodi sono necessari per lo scopo ed assegnare qualsiasi funzione di forma senza che il metodo visto perda in semplicità o smetta di funzionare. Per un approfondimento sulla generalizzazione del problema consultare [8]. Fino a questo punto della trattazione si è andati a considerare una barra rigida e, quindi, assimilabile ad un caso unidimensionale. Il campo di spostamento trasversale (lungo la sezione della barra) per il caso in esame sarebbe quindi costante. Per generalizzare lo studio ed arrivare ad estendere la CUF per qualsiasi caso (1D, 2D o 3D) si inizia a considerare il campo di spostamenti trasversale come non costante nella sezione della barra. In questo modo:

$$u(y) \rightarrow u(x, y, z)$$

In questo modo il campo di spostamenti è diventato tridimensionale. La differenza dei campi di spostamento, utilizzando tale assunzione, è rappresentata in figura 2.2. Ricordando la prima della 2.1 posso riscrivere il campo di spostamenti tridimensionale come:

$$u(x, y, z) = N_i(y) u_{y_i}(x, z) \quad (2.7)$$



**Figura 2.2:** Confronto dei campi di spostamento 1D (a sinistra) e 3D (a destra) [8]

Nel caso semplice visto in precedenza,  $N_i(y)$ , serviva per approssimare il campo di spostamenti lungo l'asse  $y$ . La stessa funzione la ricopre ora, con la differenza che adesso vi è anche la dipendenza dalle altre due coordinate. Tale dipendenza è messa in risalto da  $u_{y_i}(x, z)$  che può essere approssimato tramite una generica espansione per la direzione trasversale dell'elemento:

$$u_{y_i}(x, z) = F_\tau u_{y_{\tau i}} = F_1(x, z)u_{y_{1i}} + F_2(x, z)u_{y_{2i}} + \dots + F_m(x, z)u_{y_{Mi}} \quad (2.8)$$

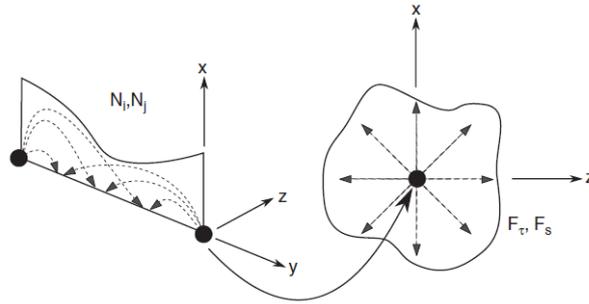
Dove  $M$  è il numero dei termini di espansione. Il campo di spostamenti completo può essere ottenuto andando a combinare 2.7 e 2.8, ottenendo così la seguente relazione:

$$u(x, y, z) = N_i(y)u_{y_i}(x, z) = N_i(y)F_\tau(x, z)u_{y_{\tau i}} \quad (2.9)$$

Anche in questo caso si può andare a ricavare la variazione virtuale utilizzando l'indice  $j$  al posto dell'indice  $i$  (come prima) e l'indice  $s$  al posto dell'indice  $\tau$ . Ottengo così:

$$\delta u(x, y, z) = N_j(y)F_s(x, z)\delta u_{y_{sj}} \quad (2.10)$$

Attraverso queste notazioni si indica che le funzioni di forma  $N$  espandono la soluzione dai nodi all'asse della barra, mentre i coefficienti di espansione  $F$  espandono la soluzione dai nodi alla sezione trasversale della barra. Questo meccanismo è rappresentato in figura 2.3. La potenza della CUF sta nel fatto che l'approccio al calcolo delle matrici FEM non varia rispetto a quello semplice introdotto in precedenza. Utilizzando questa notazione, sulla base di quello che è stato



**Figura 2.3:** Approssimazione assiale tramite  $N$  (a sinistra) e nella sezione tramite  $F$  (a destra) [8]

detto, si possono ricavare le formulazioni per lo sforzo e la deformazione:

$$\begin{aligned} \varepsilon(x, y, z) &= \mathbf{b}N_i(y)F_\tau(x, z)u_{y_{\tau i}} \\ \sigma(x, y, z) &= \mathbf{C}\mathbf{b}N_i(y)F_\tau(x, z)u_{y_{\tau i}} \end{aligned} \quad (2.11)$$

Dove  $\mathbf{b}$  è una matrice contenente operatori differenziali (per questo motivo  $\mathbf{b}$  viene detto operatore differenziale), mentre  $\mathbf{C}$  contiene i coefficienti di rigidità del materiale. Nel caso di campo di

spostamenti non costante lungo la sezione sforzo e deformazione sono dei vettori:

$$\begin{aligned}\varepsilon_{x,y} &= \frac{\delta u_y}{\delta x} + \frac{\delta u_x}{\delta y} \neq 0 \\ \varepsilon_{z,y} &= \frac{\delta u_y}{\delta z} + \frac{\delta u_z}{\delta y} \neq 0\end{aligned}\quad (2.12)$$

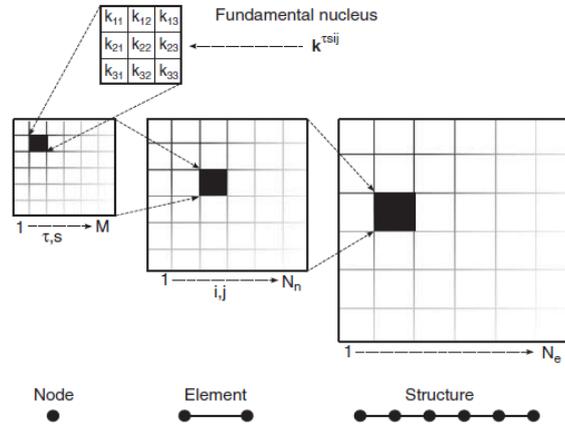
La variazione virtuale assume la forma:

$$\delta \boldsymbol{\varepsilon}(x, y, z) = \mathbf{b}N_j(y)F_s(x, z)\delta u_{sj} \quad (2.13)$$

Per il caso di una trave con campo di spostamenti tridimensionale e studiabile nelle tre direzioni, l'approccio per il calcolo della matrice di rigidezza non varia. In questo caso la matrice di rigidezza è di dimensione  $3 \times 3$  ed assume la seguente forma:

$$\mathbf{k}^{tsij} = \int_V F_s(x, z)N_j \mathbf{b}^T \mathbf{C} \mathbf{b} N_i(y) F_\tau(x, z) dV \quad (2.14)$$

L'ultimo passo nella trattazione della Carrera Unified Formulation riguarda l'assemblaggio delle matrici. Data la sua natura, la CUF, comporta avere matrici che si vanno a scomporre su quattro indici  $(i, j, \tau, s)$ . Tuttavia, anche se questa operazione potrebbe risultare complessa, può essere facilmente implementata andando ad agire prima sugli indici più "interni" e, poi, sugli indici esterni. Nello specifico si lavora prima su  $\tau$  ed  $s$ , ovvero sulla traslazione dal nodo alla sezione trasversale, per una data coppia di valori  $ij$  e lo si fa per tutte gli  $M$  termini dell'espansione. Dopodiché, si agisce allo stesso modo su tutti gli  $N_n$  termini di  $i$  e  $j$ . Infine, si va ad agire su tutti gli  $N_e$  elementi che compongono la struttura. In questo modo si vanno a comporre le matrici. Il procedimento viene schematizzato in figura 2.4.



**Figura 2.4:** Schematizzazione della procedura di assemblaggio delle matrici. Si può notare un primo ciclo sugli indici  $\tau$  ed  $s$  per una data coppia  $ij$ , una volta completato un nodo si agisce sull'elemento muovendosi per tutte le coppie  $ij$  ed infine su tutti gli  $N_e$  elementi. [8]

Quanto trattato finora per una modellazione FEM unidimensionale può essere facilmente esteso, utilizzando gli indici introdotti dalla CUF, per qualsiasi modello agli elementi finiti (sia bidimensionale che tridimensionale). Quello che andranno a cambiare sono le dipendenze di funzioni di forma e funzioni di espansione. Nello specifico:

$$\begin{aligned}1D &\rightarrow N_i(y) \\ 2D &\rightarrow N_i(x, y) \\ 3D &\rightarrow N_i(x, y, z)\end{aligned}\quad (2.15)$$

Le funzioni di espansione saranno invece:

$$\begin{aligned} 1D &\rightarrow F_\tau(y, z) \\ 2D &\rightarrow F_\tau(z) \\ 3D &\rightarrow 1 \end{aligned} \tag{2.16}$$

A questo punto è possibile andare a definire il campo di spostamento per qualsiasi modello FEM utilizzato mantenendo sempre la stessa notazione:

$$\begin{aligned} 1D &\rightarrow \mathbf{u}(x, y, z) = N_i(y)F_\tau(x, z) \\ 2D &\rightarrow \mathbf{u}(x, y, z) = N_i(x, y)F_\tau(z) \\ 3D &\rightarrow \mathbf{u}(x, y, z) = N_i(x, y, z) \end{aligned} \tag{2.17}$$

Maggiori dettagli vengono forniti in [8]

## 2.2 Best Theory Diagram

La Best Theory Diagram (BTD) è un'estensione della CUF. Tramite questa teoria ci si pone l'obiettivo di andare a studiare l'influenza dei vari termini di espansione di un modello di riferimento sulla qualità del risultato. Il campo di spostamenti derivante dal modello di riferimento, che deve essere sufficientemente accurato, viene considerato come campo di spostamento vero. A partire da questa assunzione si andranno a confrontare i modelli ridotti (ovvero quei modelli strutturali a cui, partendo dal riferimento, sono stati tolti dei termini di espansione) con quello di partenza per valutare l'errore e quindi l'importanza del termine. Attraverso questa analisi si vuole andare a vedere se all'interno di un'espansione possono esserci dei termini ininfluenti tali per cui il costo computazionale da loro derivato possa essere eliminato. Oppure, al contrario, si vuole vedere se alcuni termini di espansione hanno un'importanza tale da non poter mai essere trascurati.

A questo punto si procede con l'analisi "step-by-step" del funzionamento di tale teoria. La BTD è un risultato della MAAA ("*mixed axiomatic-asymptotic approach*") che a sua volta deriva dalla CUF come espansione. Per andare a valutare l'influenza di ogni termine si utilizza una procedura a step:

- Vengono fissati i dati del problema quali carichi, condizioni al contorno e materiali;
- viene scelto un set di variabili in output che vogliamo valutare, ad esempio sforzo o campo di spostamenti;
- si utilizza la CUF per generare le equazioni di governo per la teoria considerata;
- si fissa una teoria di riferimento per andare poi a caratterizzare l'accuratezza dei modelli ridotti;
- ogni termine della teoria di partenza viene in turno disattivato (ovvero non considerato) e per tutte le combinazioni possibili;
- si valuta se l'assenza di tale termine ha cambiato radicalmente il risultato oppure no;
- alla fine dell'analisi si trova qual è il modello strutturale ridotto migliore per una data applicazione.

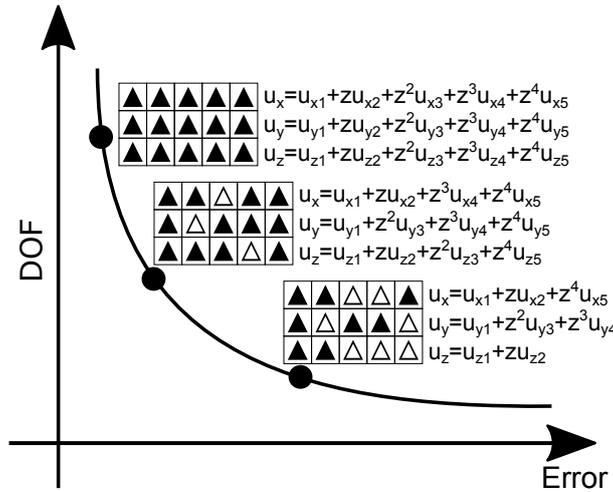
Viene utilizzata una notazione grafica come quella rappresentata in tabella 2.1 nella quale il triangolo nero significa termine attivo, mentre il triangolo bianco significa termine disattivato.

Attraverso il metodo appena descritto è possibile andare a determinare un diagramma in grado

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	▲	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	△
10	▲	▲	▲	▲	▲	△	▲	△	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	▲	△	△	△
8	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.82			RF <sub>2</sub> = 0.58			RF <sub>3</sub> = 0.67			RF <sub>4</sub> = 0.27		

**Tabella 2.1:** Esempio tabella utilizzata per MAAA, nello specifico per la ricerca di una curva BTB

di presentare il minimo numero di elementi utilizzabili in un modello agli elementi finiti. Tale diagramma, inoltre, chiarisce quali sono i termini che non sono trascurabili in un modello ridotto. La MAAA ci permette di ottenere un grafico che mette in relazione i gradi di libertà con l'errore commesso per tutte le combinazioni possibili. È possibile, partendo da tale grafico, andare a determinare una curva che unisce il minimo errore per ogni grado di libertà. Tale curva è solitamente iperbolica ed è proprio la curva Best Theory Diagram. Un esempio di BTB viene riportato in figura 2.5 in cui sono riportati tre modelli strutturali ridotti che presentano sempre meno gradi di libertà. Si nota come l'errore sia sempre maggiore man mano che i DOFs diminuiscono, indice di un'accuratezza del modello via via più bassa. Questa curva cambia se si cambiano i dati del



**Figura 2.5:** Esempio curva BTB per N = 4

problema come la geometria considerata, il carico e le condizioni al contorno. Inoltre, al variare delle variabili di output considerate, varierà anche la curva. I punti che compongono la BTB sono quindi il minimo errore ottenibile per un dato numero di termini considerati in quanto nascono dal confronto di tutte le migliaia di possibilità che il modello di partenza fornisce. Attraverso questa curva è possibile andare, partendo da un modello arbitrario che si troverà al di sopra della BTB, a fissare un valore di errore e ridurre il numero di termini e quindi ridurre il costo computazionale (faccio questo muovendomi verticalmente nel diagramma). Al contrario è anche possibile andare a fissare il numero di termini e quindi il costo computazionale e andare ad agire sull'accuratezza del risultato (mi muovo orizzontalmente sul diagramma).

Nel corso di questa tesi viene sempre utilizzato (salvo diversa indicazione) un modello di riferimento  $N = 4$  come il seguente:

$$\begin{aligned} u_x &= u_{x_1} + zu_{x_2} + z^2u_{x_3} + z^3u_{x_4} + z^4u_{x_5} \\ u_y &= u_{y_1} + zu_{y_2} + z^2u_{y_3} + z^3u_{y_4} + z^4u_{y_5} \\ u_z &= u_{z_1} + zu_{z_2} + z^2u_{z_3} + z^3u_{z_4} + z^4u_{z_5} \end{aligned} \quad (2.18)$$

Tali modelli presentano 15 incognite nodali e quindi  $2^{15}$  gradi di libertà da considerare. L'errore viene valutato andando a confrontare il modello intero di partenza con il modello ridotto e si calcola nel seguente modo:

$$\epsilon[\%] = \frac{|u_z - u_z^{N=4}|}{|u_z^{N=4}|} 100 \quad (2.19)$$

L'errore, partendo dal modello strutturale di partenza, viene normalizzato (tipicamente) con la seguente espressione:  $\epsilon = 100 \frac{|u_z - u_z^{N=4}|}{|u_z^{N=4}|}$ . Per maggiori approfondimenti consultare [8].

## 2.3 Reti neurali

Il processo di ricerca delle curve Best Theory Diagram al variare dei parametri di input, nel corso della tesi, verrà a mano a mano automatizzato. Per fare questo si sono utilizzate le reti neurali che sono di seguito presentate.

Il termine rete neurale deriva da una sorta di analogia tra le reti neurali cerebrali (quindi reti biologiche) e quelle artificiali. Le differenze sono molte e spaziano dalla velocità di elaborazione dati alla numerosità dei componenti e delle connessioni tra di essi. In questa fase ci si concentrerà però sulle analogie tra reti biologiche ed artificiali. Ogni elemento dei due tipi di reti è un "dispositivo" di calcolo molto semplice e il numero di neuroni e connessioni determina la possibilità per una rete di svolgere operazioni più o meno complesse. Insieme ad altri elementi, sono proprio il numero di neuroni e le tipologie di connessioni che, come si vedrà nel seguito, portano una rete a funzionare più o meno bene, essere più o meno veloce ed alla possibilità di svolgere funzioni più o meno complesse.

L'elemento più semplice per una rete neurale è il neurone a singolo input ed è rappresentato in figura 2.6 (questa immagine e le altre presenti in questo capitolo sono tratte dal libro [12]). Esso è in grado di gestire un solo input "p" (di tipo scalare) che viene moltiplicato per un vettore scalare "w" detto peso. Il vettore risultante viene mandato al sommatore che riceve un secondo input (1) che viene moltiplicato per il termine "b" detto bias. L'output del sommatore "n" viene mandato alla funzione di trasferimento "f" che produce uno scalare "a" che è l'output del neurone. Per questo semplice caso, l'output è dato da una funzione di  $n$  con  $n = wp + b$ .

La funzione di trasferimento può essere di diverso tipo in base all'esigenza del problema analizzato. Essa può essere lineare o non lineare, ciò che regola la scelta di una funzione rispetto ad un'altra è la necessità di soddisfare alcuni criteri piuttosto che altri. Ci sono moltissime funzioni di trasferimento, viene di seguito riportata la funzione utilizzata nel corso della tesi:

**Hyperbolic Tangent Sigmoid** abbreviata con la sigla *tansig*. L'output del neurone è dato dalla seguente relazione:

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$$

Tale funzione di trasferimento è in grado di modulare l'output del sommatore in un intervallo compreso tra 0 ed 1. Questo tipo di funzione di trasferimento è particolarmente indicata per i problemi di predizione e meno performante per i problemi di classificazione. Essendo che, nel corso di tutta la tesi, ci si concentrerà sempre su operazioni di predizione (di errori o di modelli

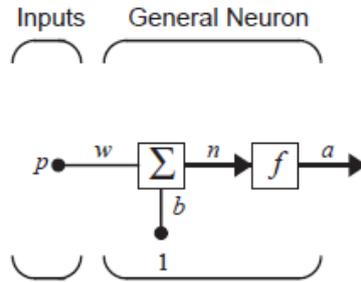


Figura 2.6: Neurone singolo input

strutturali), questa sarà la funzione di trasferimento utilizzata.

Il neurone a singolo input, come detto, è l'elemento più semplice possibile per un una rete neurale, ma è anche l'elemento base. Esso si può complicare aggiungendo scalari in input ed ottenendo un vettore  $p$  contenente gli  $R$  elementi che lo compongono. Ogni elemento del vettore viene pesato dagli elementi della matrice dei pesi  $\mathbf{W}$  e la funzione che regola l'output del neurone è la seguente:  $n = \mathbf{W}p + b$ .

Un solo neurone, seppur con molti input, non è in grado di risolvere funzioni molto complesse. All'inizio del capitolo, infatti, si era parlato di numero di collegamenti e numero di elementi che compongono una rete. Per questo motivo sono necessari decine di neuroni che lavorano contemporaneamente gestendo gli input del problema (figura 2.7). Ogni neurone ha il proprio bias e la propria funzione di trasferimento e tutti gli elementi sono collegati attraverso la matrice dei pesi ai neuroni. La funzione di trasferimento può essere diversa da neurone a neurone, se necessario. È importante notare che ogni elemento del vettore di input è collegato ad ognuno dei neuroni tramite le diverse matrici dei pesi. In generale il numero di input è diverso dal numero di neuroni presenti nel layer.

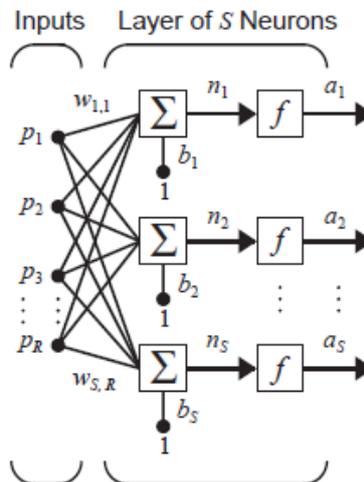


Figura 2.7: Layer di neuroni

Infine, una rete è la composizione di più o meno layers a seconda della complessità del problema che si sta analizzando. Il principio di funzionamento è lo stesso visto per il singolo layer; in questo caso vi sono gli input dei layer interni sono gli output dei layer precedenti. Possono essere distinte due famiglie di layer:

- Hidden layers: sono i layer della rete che non danno come output l'output della rete;
- Output layer: è il layer che da come output l'output della rete e quindi il risultato.

Una schematizzazione generale di una rete è riportata in figura 2.8 al quale è associata la seguente funzione per ricavare l'output:

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3)$$

Il numero di output che il problema richiede è il vincolo per la scelta del numero di neuroni dell'output layer. Allo stesso modo, il numero di input del problema determina il numero di input del primo layer. Per quanto riguarda gli hidden layers, invece, non vi è una soluzione univoca.

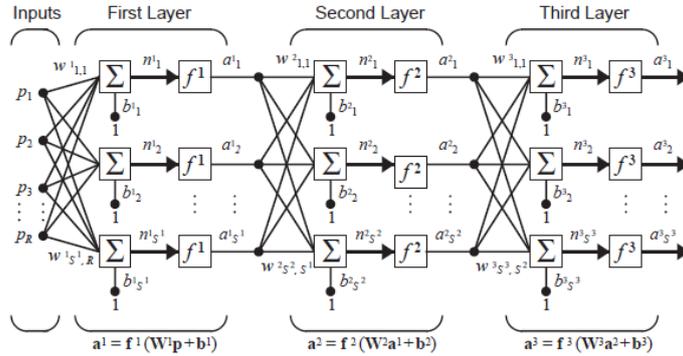


Figura 2.8: Rete

In questo capitolo è stato riportato in breve il funzionamento generale di una semplice rete neurale, per approfondimenti e dettagli maggiori consultare [12].

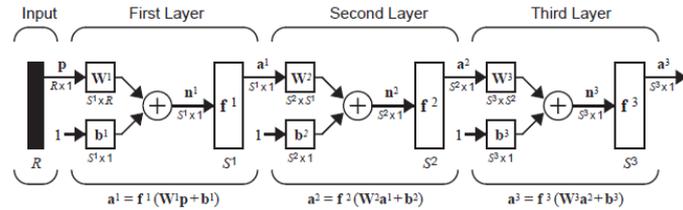


Figura 2.9: Rete con tre layer

Fino a questo momento si è discusso di come è strutturata una rete neurale dal punto di vista concettuale. Per fare in modo che funzioni è necessario che questa venga allenata per far sì che risponda alle esigenze per cui è stata pensata. Ci sono moltissimi modi diversi per poter allenare una rete e sono principalmente suddivisi in base al tipo di funzione che devono andare a svolgere, ma in questa sezione ci si focalizzerà sull'algorithmo di training chiamato *Backpropagation*. Per lo studio di tale algorithmo si procede utilizzando lo schema riportato in figura 2.8, ma andandolo a modificare leggermente per avere una notazione più breve ed otteniamo lo schema in figura 2.9. Come già visto, l'output della rete è dato dalla seguente relazione:

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3)$$

dove

$$a^0 = p$$

ovvero l'input del primo layer è proprio l'input del problema, mentre

$$a = \mathbf{a}^M$$

cioè l'output dell'ultimo layer è l'output della rete.

L'algoritmo, per poter essere allenato, deve essere fornito da coppie di dati del tipo:

$$(p_1, t_1), (p_2, t_2), \dots, (p_Q, t_Q)$$

dove  $p$  sta ad indicare il generico input della rete, mentre  $t$  è il suo target. Il target è il risultato a cui noi vogliamo tendere una volta che la rete risulta essere allenata. L'algoritmo utilizza il *mean square error* (errore quadratico medio) per il calcolo dell'errore e tenta quindi di minimizzarlo andando a variare i parametri della rete.

$$F(x) = E[e^2] = E[(t - a)^2]$$

dove  $x$  rappresenta il vettore contenente i pesi ed i bias. Si può generalizzare per una rete con più di un output nel seguente modo:

$$F(x) = E[\mathbf{e}^T \mathbf{e}] = E[(\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})]$$

È possibile approssimare l'errore quadratico medio come segue:

$$F(x) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k) \mathbf{e}(k)$$

Dove è apparso l'errore quadratico medio all'iterazione  $k$ .

I pesi e i bias vengono aggiornati attraverso le seguenti funzioni:

$$\begin{aligned} w_{i,j}^m(k+1) &= w_{i,j}^m(k) - \alpha \frac{\partial F}{\partial w_{i,j}^m} \\ b_i^m(k+1) &= b_i^m(k) - \alpha \frac{\partial F}{\partial b_i^m} \end{aligned} \quad (2.20)$$

$\alpha$  è il rateo di apprendimento della rete ed è un numero compreso tra 0 e 1. Se è troppo piccolo la convergenza sarà lenta, mentre se è grande la soluzione può oscillare o divergere. La scelta del valore del rateo di apprendimento è fortemente basata sul tipo di problema e sull'esperienza di chi progetta la rete neurale. A questo punto non rimane che trovare le espressioni per il calcolo delle derivate parziali. Siccome l'errore è una funzione indiretta dei pesi nei layers interni, si ricorre alla *regola della catena* per la determinazione delle derivate. Se si suppone di avere una funzione  $f$  che è una funzione esplicita della sola variabile  $n$  e di volerla derivare rispetto ad una terza variabile  $w$ , la regola della catena ci dice che:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \times \frac{dn(w)}{dw}$$

Utilizzando questo concetto le derivate parziali trovate nelle equazioni 2.20 si possono scrivere come segue:

$$\begin{aligned} \frac{\partial F}{\partial w_{i,j}^m} &= \frac{\partial F}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \\ \frac{\partial F}{\partial b_i^m} &= \frac{\partial F}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m} \end{aligned} \quad (2.21)$$

Il secondo termine è facilmente determinabile in quanto l'input netto di un determinato layer è una funzione esplicita dei pesi e bias di quel layer. Quindi si ottiene che, partendo dalla definizione di input:

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m$$

derivando si ottiene:

$$\begin{aligned}\frac{\partial n_i^m}{\partial w_{i,j}^m} &= a_j^{m-1} \\ \frac{\partial n_i^m}{\partial b_i^m} &= 1\end{aligned}\tag{2.22}$$

A questo punto si va a definire la *sensibilità* come  $s_i^m = \frac{\partial F}{\partial n_i^m}$ .

Ora, se si riportano i risultati fin qui ottenuti nelle equazioni 2.20, si ricava:

$$\begin{aligned}w_{i,j}^m(k+1) &= w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \\ b_i^m(k+1) &= b_i^m(k) - \alpha s_i^m\end{aligned}\tag{2.23}$$

Che in forma matriciale diventa:

$$\begin{aligned}\mathbf{W}_{i,j}^m(k+1) &= \mathbf{W}_{i,j}^m(k) - \alpha s_i^m \mathbf{a}_j^{m-1} \\ \mathbf{b}_i^m(k+1) &= \mathbf{b}_i^m(k) - \alpha s_i^m\end{aligned}\tag{2.24}$$

Fino a questo momento non è ancora stato giustificato il nome Backpropagation che viene attribuito a questo algoritmo di allenamento per le reti neurali. A tal proposito rimane da determinare la sensibilità. Anche in questo caso verrà utilizzata la regola della catena. Per prima cosa è necessario andare a scrivere la matrice Jacobiana al fine di ricercare la relazione di ricorrenza della sensibilità:

$$\frac{\partial \mathbf{n}^{m+1}}{\mathbf{n}^m} = \begin{bmatrix} \frac{\partial n_1^{m+1}}{n_1^m} & \frac{\partial n_1^{m+1}}{n_2^m} & \dots & \frac{\partial n_1^{m+1}}{n_{S^m}^m} \\ \frac{\partial n_2^{m+1}}{n_1^m} & \frac{\partial n_2^{m+1}}{n_2^m} & \dots & \frac{\partial n_2^{m+1}}{n_{S^m}^m} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial n_{S^m+1}^{m+1}}{n_1^m} & \frac{\partial n_{S^m+1}^{m+1}}{n_2^m} & \dots & \frac{\partial n_{S^m+1}^{m+1}}{n_{S^m}^m} \end{bmatrix}$$

Se si considera l'elemento  $i,j$  della matrice, si ottiene che:

$$\frac{\partial n_i^{m+1}}{n_j^m} = \frac{\partial(\sum_{l=1}^{S^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1})}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{i,j}^{m+1} f^m(n_j^m)$$

La matrice Jacobiana può essere scritta come:

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \mathbf{W}^{m+1} \mathbf{F}^m(\mathbf{n}^m)$$

dove

$$\mathbf{F}^m(\mathbf{n}^m) = \begin{bmatrix} f^m(n_1^m) & 0 & \dots & 0 \\ 0 & f^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & f^m(n_{S^m}^m) \end{bmatrix}$$

A questo punto è possibile scrivere la formula della ricorrenza per la sensibilità come segue:

$$s^m = \frac{\partial F}{\partial \mathbf{n}^m} = \left( \frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right)^T \frac{\partial F}{\partial \mathbf{n}^{m+1}} = \mathbf{F}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \frac{\partial F}{\partial \mathbf{n}^{m+1}} = \mathbf{F}(\mathbf{n}^m) (\mathbf{W}^{m+1})^T s^{m+1}$$

Da quest'ultima relazione è evidente dov'è la propagazione all'indietro, in quanto la sensibilità viene propagata dall'ultimo layer fino al primo. L'ultimo elemento che rimane da determinare per completare l'algoritmo che si andrà ad utilizzare nel seguito è il punto di partenza della sensibilità, ovvero quella nell'ultimo layer. Questa viene determinata nel seguente modo:

$$s_i^m = -2(t_i - a_i) f^m(n_i^M)$$

che può essere espresso in forma matriciale come:

$$\mathbf{s}^m = -2\mathbf{F}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a})$$

In questo capitolo è stata data una panoramica generale dell'algoritmo che verrà utilizzato nei capitoli successivi, per ulteriori approfondimenti consultare [12].

### 2.3.1 Utilizzo della rete neurale per la determinazione della BTD

Prima di entrare nel merito di come viene allenata una rete neurale (verrà riportato il solo processo diretto, ovvero quello con in input i modelli strutturali e come training l'errore) è utile andare a capire come funziona il meccanismo della rete. Per prima cosa si ottengono, dalle analisi agli elementi finiti, gli input ed i training che serviranno per andare ad allenare la rete neurale. Naturalmente, a seconda del tipo di problema, gli input ed i target saranno diversi. A questo punto si va a fornire alla rete i dati che si sono ottenuti in modo da poter effettuare il training. Quello che si otterrà a seguito di questa fase è una rete allenata in grado di fornire ad un dato input un output che sia il più possibile vicino ad i target forniti in fase di training. Le fasi che sono appena state descritte si trovano sintetizzate nel grafico in figura 2.10.

A questo capitolo ci si concentra sulla determinazione delle curve BTD, a seguito di un training della rete, utilizzando un numero ridotto di dati in input. Come visto nelle sezioni precedenti, il modello  $N = 4$  possiede un numero molto elevato di incognite ( $2^{15}$  o  $2^{12}$  a seconda del problema analizzato). Tale numero di incognite si traduce in grandi tempi di calcolo, che arrivano anche alla decina di minuti, dovuti all'ingente costo computazionale. Nell'ottica di un'applicazione di calcolo più complessa questo tempo di calcolo potrebbe diventare insostenibile.

Attraverso l'utilizzo di una rete neurale opportunamente allenata è possibile andare a ridurre notevolmente i tempi di calcolo e la necessità di svolgere simulazioni con programmi agli elementi finiti. Infatti, fornendo alla rete un numero finito di coppie input/output è possibile, in linea teorica, andare a ricostruire tutte le curve BTD desiderate. Il vantaggio sta nel fatto dover fare le simulazioni solo a monte e poi utilizzare tali simulazioni per allenare la rete in tutte le applicazioni successive.

Nelle sezioni successive si illustrerà proprio il processo di allenamento della rete neurale. L'algoritmo utilizzato è quello illustrato nel capitolo 2.3, mentre il numero di layers, le funzioni di trasferimento e i cicli di allenamento dovranno essere ricercati iterativamente.

### 2.3.2 Allenamento della rete neurale

In questa sezione si è andati a studiare come i diversi parametri che determinano il training della rete siano influenti sul risultato finale. Nel seguito di questo paragrafo verranno illustrati i passaggi effettuati per l'allenamento della rete di un guscio quadrato di spessore  $h/a = 0.01$ , ma le considerazioni fatte sono state eseguite per diversi spessori.

Per prima cosa si è andati a scegliere il numero di strati interni alla rete, in quanto proprio il loro numero e il quantitativo di collegamenti tra di essi ed i neuroni va a determinare la complessità delle operazioni che sono in grado di svolgere. Il caso con un solo layer sembrerebbe funzionare per il semplice problema della determinazione delle curve BTD, ma in seguito si è visto come al complicarsi del problema (inserimento dell'influenza dello spessore tra gli input), un solo strato interno nella rete non riesca a riprodurre in modo sufficientemente fedele la curva. In figura 2.11 è riportato un esempio di curva BTD per un guscio con  $h/a = 0.15$  ricavato a partire da altri spessori noti (questo processo verrà analizzato nel dettaglio nel paragrafo 8, qui è riportato l'andamento solo a scopo illustrativo). Si nota come la curva derivante dalle FEM, ovvero quella esatta, e la curva ricavata dalla rete neurale siano completamente staccate. Si è quindi aumentato il numero

di strati interni per migliorare le soluzioni. Quello che si è notato è che aumentando molto gli strati interni (7/9) la soluzione non migliora più nei problemi trattati, ma il tempo computazionale aumenta molto. Si è scelto quindi di utilizzare tre strati interni.

Il secondo parametro da trattare è stato quello relativo alla scelta del numero di neuroni per ogni strato interno. Anche in questo caso si è partiti utilizzando pochi neuroni e si sono a mano a mano aumentati; gli andamenti ottenuti sono riportati in figura 2.12.

Analogamente al caso precedente, la curva colta con la rete più semplice (5 neuroni) sembrerebbe essere la migliore insieme a quella colta con 10 neuroni (riportata in figura 2.14a). Infatti, le altre curve presentano sempre errori più o meno negativi nell'intono dello zero e questa è una situazione che si cerca di evitare. Come per il caso del numero di strati interni, anche qui aumentando la complessità del problema (nello specifico aggiungendo l'influenza dello spessore) un numero basso di neuroni non è sufficiente a garantire un risultato accettabile. Per questo motivo la rete utilizzata avrà 10 neuroni per ogni strato interno. Aumentare di molto questo numero, nei problemi trattati non porta benefici sostanziali (talvolta addirittura peggiora il risultato), ma aumenta il costo computazionale.

La rete neurale utilizza un certo numero di coppie di dati input/output per allenarsi. Queste coppie di dati vengono suddivise per il training, la validazione e il testing della rete. Nella fase di training è possibile impostare un numero di cicli di iterazione all'interno dei quali la rete prende delle coppie di dati sempre diversi e fornisce come risultato la rete allenata. Alla fine del ciclo, la rete fornita che darà il minor errore sarà proprio la rete da utilizzare. Aumentando il numero di cicli di iterazione aumenta la specializzazione della rete. Si è andati allora a studiare come questo parametro influisce sui risultati ottenuti dalla rete, gli andamenti sono riportati in figura 2.13.

Si osserva che un'eccessiva specializzazione della rete comporta risultati controproducenti e, oltre ad aumentare notevolmente il costo computazionale, peggiora anche il risultato che si ottiene.

L'ultima osservazione la si fa sulla metodologia di scelta delle coppie di dati input/output a partire dai risultati completi provenienti dalla FEM. Fino a questo momento il codice prelevava le coppie in modo ordinato a seconda del numero di dati richiesti. Ovvero, se si voleva allenare la rete utilizzando il 50% dei dati il codice ne prendeva uno ogni due; se si voleva utilizzare un terzo dei dati ne prelevava uno ogni tre e così via. Il codice è stato modificato in modo da prelevare il numero di coppie prestabilito in modo random, non seguendo quindi degli step ordinati. I risultati ottenuti sono risultati essere decisamente migliori della soluzione precedente, come riportato in figura 2.14. Il problema riscontrato con questa metodologia di scelta delle coppie di dati risiede nel fatto che, essendo scelte in maniera casuale, esse sono diverse di volta in volta. Questo implica che le soluzioni non saranno identiche se si ripete l'analisi per più di una volta. Quello che si è osservato, però, è che mediamente la soluzione è sempre migliore quando ottenuta utilizzando i passi di scelta random.

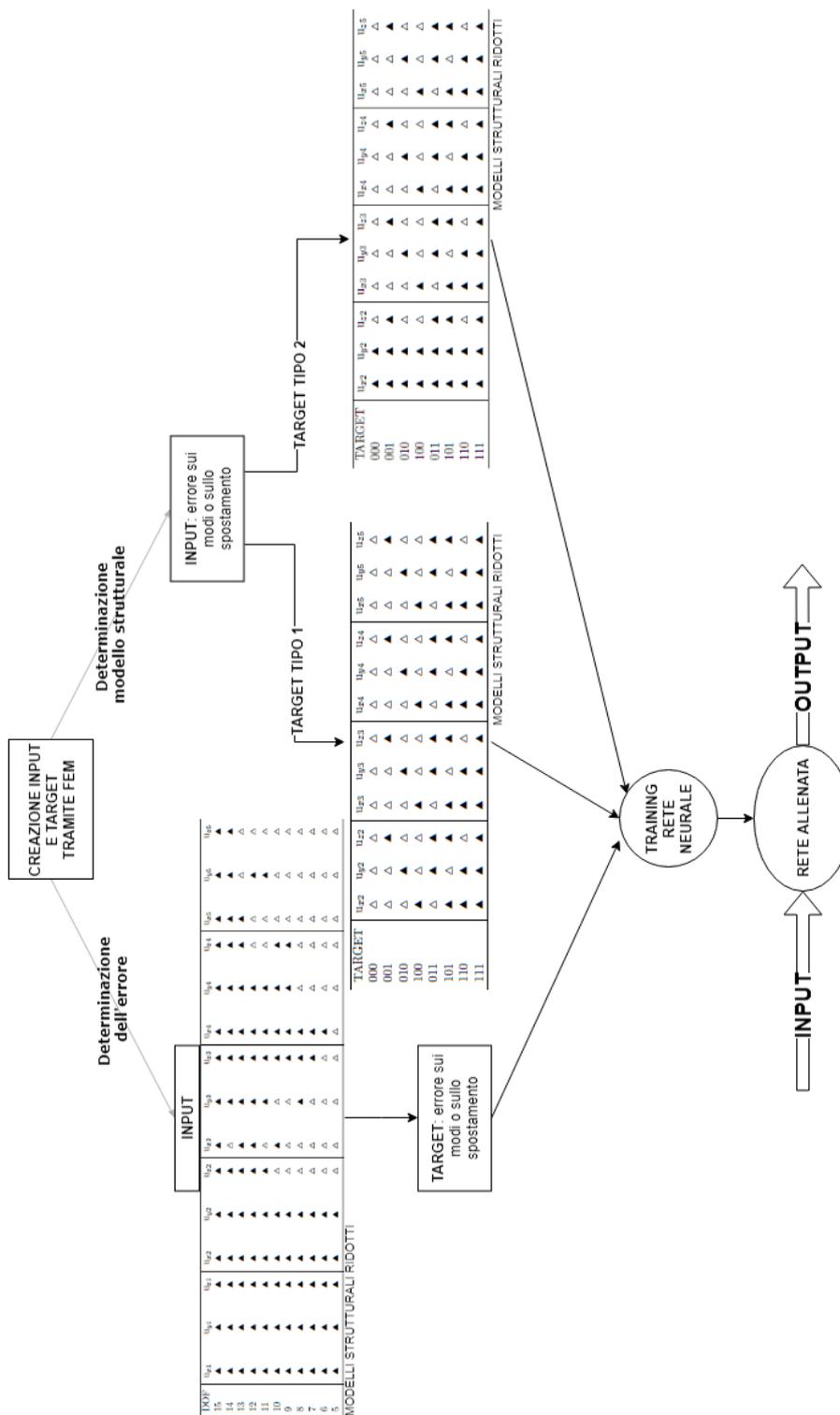


Figura 2.10: Schema riassuntivo funzionamento rete neurale

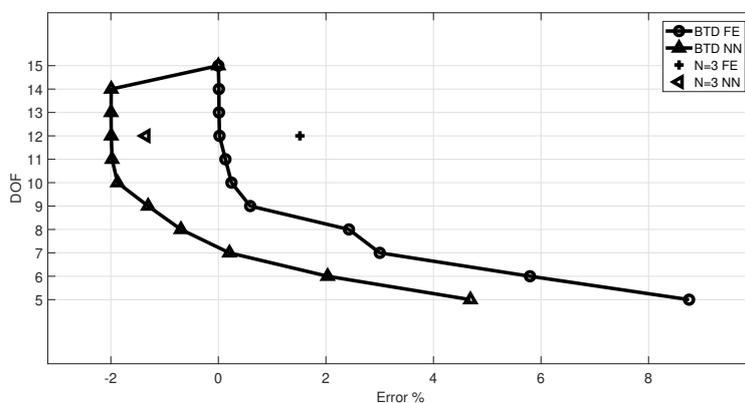


Figura 2.11: Curva BTM colta con un layer interno

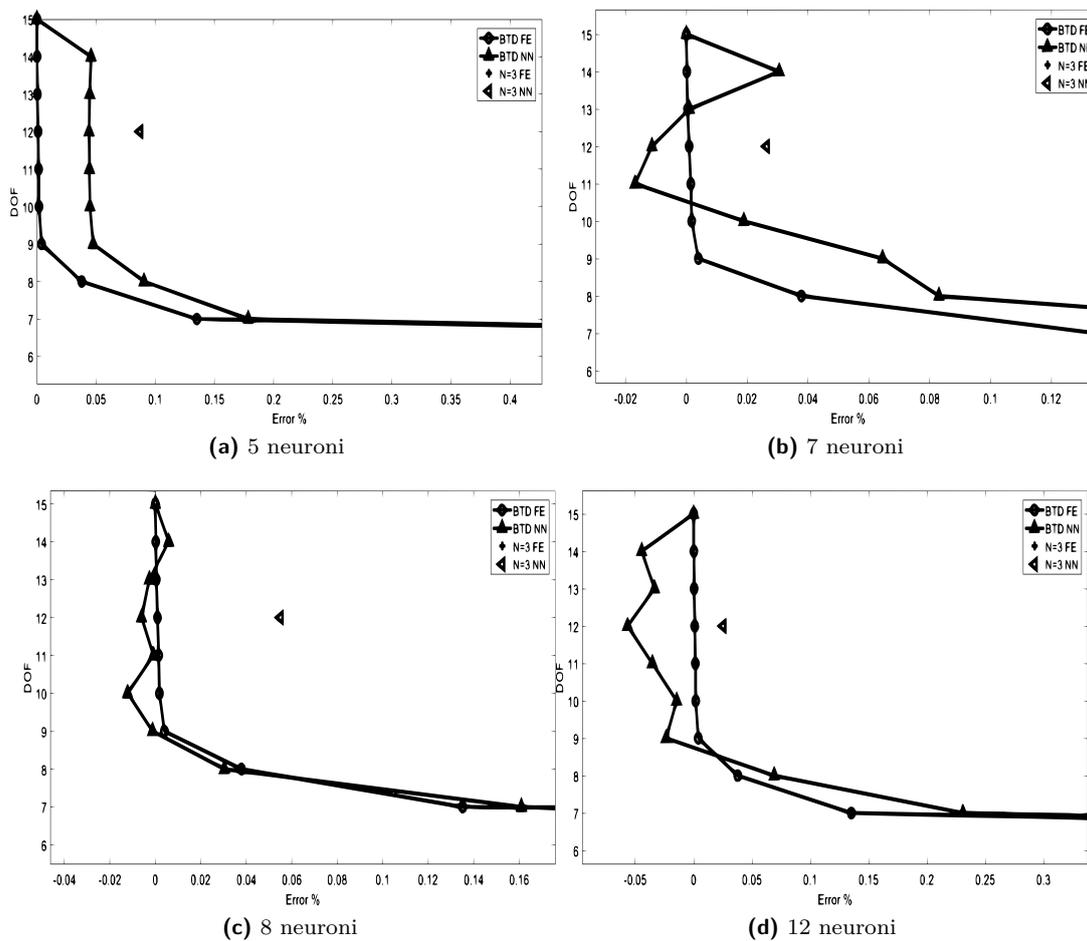


Figura 2.12: Confronto curve per diversi neuroni

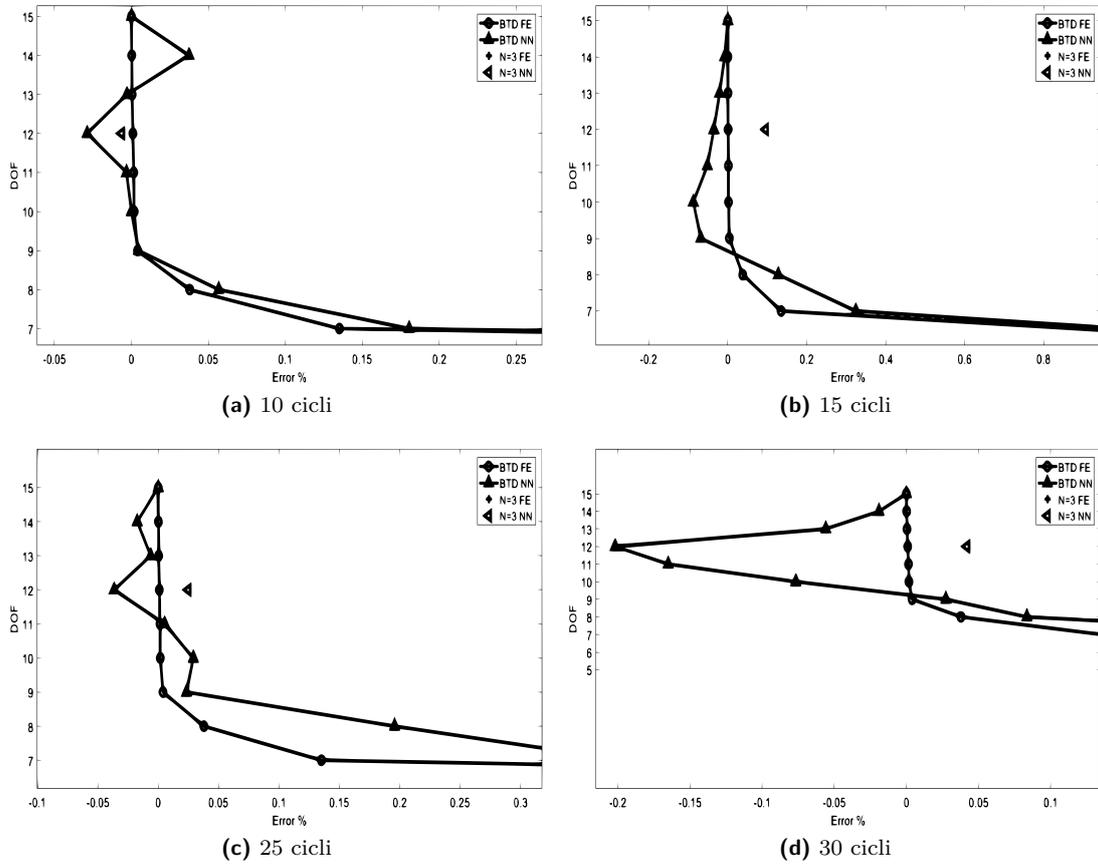


Figura 2.13: Confronto curve per diversi cicli di training

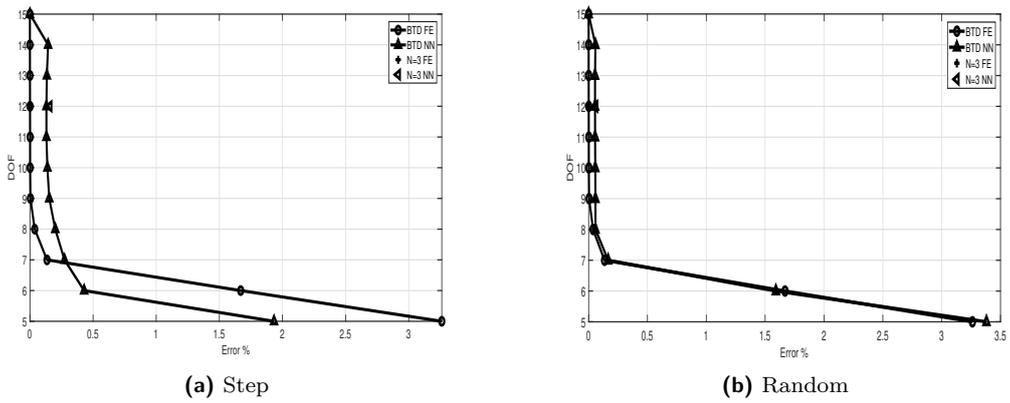


Figura 2.14: Confronto curve per coppie input/output scelti a passi di "step" e random

**Parte I**

**Risultati preliminari**



## Capitolo 3

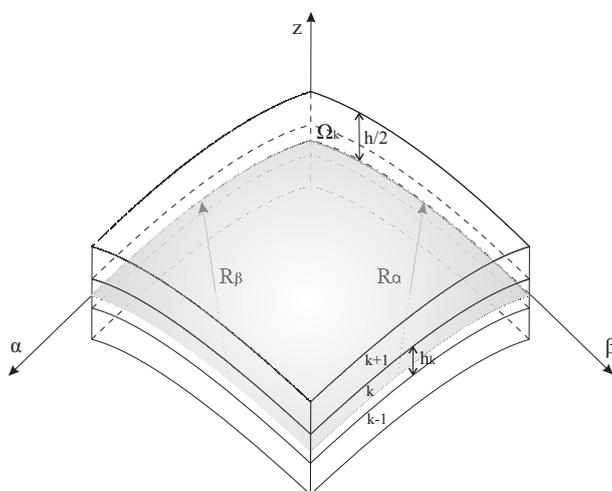
# Analisi statica e dinamica di gusci multistrato

Nella prima parte della tesi il lavoro verterà su una sessione di training dei software che verranno utilizzati nel corso di tutto il lavoro.

Il primo obiettivo da raggiungere è quello di riuscire a riprodurre le tabelle che si trovano nelle pubblicazioni [6] e [19].

Attraverso queste tabelle verranno utilizzati diversi tipi di analisi presenti nel codice, nello specifico analisi 101 relativa alla statica e analisi 103 relativa all'analisi modale.

Entrambe le analisi fanno riferimento alla figura 3.1. La prima analisi che si riporta in questa



**Figura 3.1:** Sezione guscio di riferimento

sezione è quella relativa ad un guscio di lati unitari, semplicemente appoggiato su tutti e quattro i lati. Il guscio in esame è multistrato con laminazioni orientate a  $0^\circ/90^\circ/0^\circ$  e caricato sulla faccia superiore da un carico bisinusoidale che segue la seguente legge:

$$p = p^0 \sin\left(\frac{\pi x n}{l}\right) \sin\left(\frac{\pi y m}{l}\right)$$

$R/a$  e  $a/h$  variano per tenere in considerazione curvature più o meno profonde del guscio e vari spessori.

La Tabella 3.1 riporta lo spostamento trasversale comparato con una soluzione 3D e un modello analitico basato su un modello layer-wise al quarto ordine. Si precisa che l'accuratezza di un modello con  $N = 4$  decresce con lo spessore del guscio. Il secondo caso va a trattare una diversa

Model	R/a = 5			R/a = 2		
3D [13]	-	0.7325	1.549	-	0.6087	1.482
LD4 [7]	0.1036	0.7325	1.5494	0.0208	0.6087	1.4824
ED4	0.1036	0.6975	1.4562	0.0208	0.5868	1.4046
a/h	100	10	5	100	10	5

**Tabella 3.1:**  $0/90/0$ ,  $\bar{u}_z (z = 0) = 100u_z E_T h^3 / (\bar{p}_z a^4)$ .

laminazione; nello specifico si è aggiunto uno strato portando il guscio ad avere la seguente orientazione delle fibre:  $0^\circ/90^\circ/0^\circ/90^\circ$ . Per questa analisi tutti gli altri dati sono rimasti invariati andando però ad aggiungere due valori di  $R/a$ , nello specifico 100 e 50. La tabella 3.2 riporta nuovamente gli spostamenti trasversali confrontati con modelli della letteratura.

Model	R/a = 100	R/a = 50	R/a = 5		
3D [13]	-	-	-	0.7408	1.495
LD4 [7]	-	-	0.1067	0.7408	1.4951
ED4	0.7248	0.7252	0.1067	0.7055	1.3717
a/h	10	10	100	10	5

**Tabella 3.2:**  $0/90/0/90$ ,  $\bar{u}_z (z = 0) = 100u_z E_T h^3 / (\bar{p}_z a^4)$ .

L'ultima analisi numerica del training relativo all'analisi statica propone un guscio a quattro strati, nuovamente orientati a  $0^\circ/90^\circ/0^\circ/90^\circ$ , ma questa volta con i lati paralleli a  $\beta$  incastrati e gli altri liberi. Tutti gli altri parametri geometrici sono gli stessi dei casi analizzati precedentemente. La tabella 3.3 riporta lo spostamento trasversale del modello con ordine di espansione pari a  $N = 4$ .

Model	R/a = 1		
ED4	0.0255	0.4206	1.1890
a/h	100	10	5

**Tabella 3.3:**  $0/90/0/90$ ,  $\bar{u}_z (z = 0) = 100u_z E_T h^3 / (\bar{p}_z a^4)$ .

A questo punto, una volta riprodotti i risultati dell'analisi statica, si procede con un'analisi di tipo modale. In questo caso si è cercato di confrontare la prima frequenza normalizzata di diverse teorie variando il raggio di curvatura. Lo schema a cui fare riferimento, sia per gli assi che per il tipo di guscio, è sempre quello in figura 3.1. In questo caso, rispetto al paper di riferimento, gli output non sono risultati essere uguali sino alla terza cifra decimale in tutti i casi. Lo scarto è comunque stato molto basso e tale differenza è stata imputata alla diversa versione del software utilizzata per fare le analisi.

Nell'ultima simulazione di questa fase introduttiva di training al software si vanno a confrontare le varie teorie al variare del raggio di curvatura e dello spessore della piastra. Tutti gli altri dati rimangono invariati.

Model	R/a = 1	R/a = 2	R/a = 5	R/a = 10	Plate
HSDT [21]	-	-	12.060	11.860	11.790
FSDT [20]	16.115	13.382	12.372	12.215	12.162
CLT [3]	17.820	15.878	15.233	15.136	15.104
L1	15.717	12.868	11.803	11.636	11.580
ED4	15.796	13.007	11.972	11.810	11.756
ED1	16.198	13.645	12.715	12.572	12.523
FSDT	16.250	13.709	12.784	12.642	12.593
CLT	18.654	16.267	15.376	15.238	15.191

**Tabella 3.4:**  $E_L/E_T=25$ ,  $G_{LT}/E_T=0.5$ ,  $G_{TT}/E_T=0.2$ ,  $\nu_{LT}=\nu_{TT}=0.25$ ,  $a/h=10$  e  $m=1$ ,  $n=1$ . Mesh 16x16. Laminazione  $0^\circ/90^\circ/0^\circ$ .

R/h	R/a=1			R/a=10		
	2.5	4	20	25	40	200
L1	5.399	8.637	42.961	4.787	7.263	23.976
ED4	5.386	8.622	42.964	4.761	7.369	24.350
ED2	5.401	8.642	43.042	5.074	7.937	26.094
FSDT	5.414	8.658	43.054	4.995	7.855	26.122
CLT	6.309	9.276	43.420	5.432	8.690	32.511

**Tabella 3.5:**  $E_L/E_T=25$ ,  $G_{LT}/E_T=0.5$ ,  $G_{TT}/E_T=0.2$ ,  $\nu_{LT}=\nu_{TT}=0.25$ ,  $a/h=10$  e  $m=1$ ,  $n=1$ . Mesh 16x16. Laminazione  $0^\circ/90^\circ/0^\circ$ , Vincolato/libero/vincolato/libero



## Capitolo 4

# Best Theory Diagrams per caso statico e dinamico

Nella seconda fase del training si andranno a replicare i risultati contenuti nei file [19] e [17] riguardanti la *BTD*. La prima analisi che verrà fatta sarà di tipo statico, mentre la seconda di tipo dinamico.

### 4.1 Risultati analisi statica

I risultati numerici che vengono riportati nel seguito fanno riferimento ad un guscio multistrato come quello illustrato nella sezione 3. I dati non variano e si considera la figura 3.1 come guscio di riferimento.

Nel seguito verranno visualizzati grafici che avranno sull'asse delle ordinate i gradi di libertà che andranno da 5 a 15 oppure 12. Il limite inferiore è dovuto al fatto che modelli con meno di 5 gradi di libertà forniscono errori decisamente alti. Il limite superiore, in alcune analisi, viene abbassato a 12 gradi di libertà per verificare di avere un costo computazionale minore ottenendo gli stessi risultati significativi.

#### 4.1.1 Guscio semplicemente appoggiato $0^\circ/90^\circ/0^\circ$

La prima analisi che si effettua fa riferimento alla tabella 3.1 in cui sono riportati gli spostamenti lungo l'asse  $z$  (fare riferimento alla figura 3.1) per un modello con  $N = 4$  confrontato con una teoria 3D è una soluzione analitica basata sul layer-wise al quarto ordine. Un modello con  $N = 4$  ha la precisione dei risultati che tende a diminuire al crescere dello spessore, ma in questa prima analisi lo scopo è quello di andare a valutare l'importanza dei singoli termini piuttosto che la precisione assoluta del risultato.

Nei grafici 4.1 vengono riportati tutti gli errori per un dato numero di gradi di libertà. Non verrà riportato per tutte le analisi in quanto non è interessante la variazione tra un caso e l'altro.

I grafici 4.2 e 4.3 mostrano la BTD per i determinati dati del problema. Nelle tabelle 4.1, 4.2 e 4.3 sono riportati i termini che vengono mantenuti nel modello per formare la curva BTD tramite dei triangoli neri e con dei triangoli bianchi i termini non presi in considerazione.

Si può chiaramente notare che per dare il minimo errore in output i termini  $u_{x_1}$ ,  $u_{y_1}$  e  $u_{z_1}$  siano presenti in tutti i modelli dando sempre un  $RF_0 = 1$  (il termine RF indica il rapporto tra i termini non attivi rispetto a quelli totali). È anche interessante notare come i termini al primo e secondo ordine delle espansioni diventino via via meno importanti all'aumentare dello spessore lasciando il comportamento duale ai termini del terzo e quarto ordine. I termini al quarto ordine, comunque,

assumono un'importanza notevole solo per il guscio spesso ( $a/h = 5$ ).

Prendendo in considerazione le curve BTD si può notare che la teoria FSDT restituisce un errore più basso, a pari gradi di libertà, della curva soltanto nel caso di guscio molto sottile. Aumentando lo spessore il punto prima si unisce alla curva, ma poi da errori più consistenti della teoria BTD. Per quanto riguarda il modello ad  $N = 3$ , esso è sempre contenuto all'interno della curva fatta eccezione per gusci spessi.

Molto interessante è notare che sono necessari al più 6 gradi di libertà per fornire un errore inferiore all'1%, qualsiasi caso si stia trattando.

Infine, confrontando i risultati ottenuti tramite  $R/a = 5$  e  $R/a = 2$  si nota che non vi è una sostanziale differenza nelle curve BTD, sintomo che variare lo spessore del guscio influenza l'analisi di più che variare il raggio di curvatura.

#### 4.1.2 Guscio semplicemente appoggiato $0^\circ/90^\circ/0^\circ/90^\circ$

Il secondo caso di interesse varia rispetto al primo per quanto riguarda la laminazione, ma non nelle condizioni al contorno rimanendo sempre appoggiato su tutti e quattro i lati. Tutti gli altri parametri rimangono invariati rispetto al caso precedente; si vanno ad aggiungere all'analisi due valori di  $R/a$ , 50 e 100, per studiare più nel dettaglio l'influenza della curvatura. La tabella 3.2 riporta gli spostamenti comparati anche in questo caso a due modelli in letteratura. I risultati sono riportati nelle figure 4.4, 4.5, 4.6, 4.7 e nelle tabelle 4.4, 4.5, 4.6, 4.7 e 4.8.

Leggendo i grafici si nota che, come nel caso precedente, il modello FSDT fornisce risultati migliori della BTD per gusci sottili, ma che aumentando lo spessore, l'errore dato da tale teoria cresce notevolmente.

Un cambiamento sostanziale rispetto al caso precedente risiede nel più elevato numero di gradi di libertà necessari a fornire un risultato con errore inferiore all'1%. Infatti, è evidente come, escluso il caso di guscio sottile, siano necessari più di 10 gradi di libertà. In generale quindi, con questo tipo di laminazione, si nota che l'errore restituito dalla teoria è più alto rispetto al caso precedente a parità di gradi di libertà.

Analogamente al caso precedente, anche qui, le variazioni dei raggi di curvatura non forniscono sostanziali differenze nei risultati come invece accade per diversi spessori.

In questo caso, andando a guardare il fattore RF, non vi sono particolari andamenti da seguire. Infatti la distribuzione dei termini nei vari ordini di espansione rimane pressochè la stessa anche aumentando lo spessore, fatta eccezione per i termini di quarto ordine. Anche in questo caso, infatti, l'importanza di questo contributo aumenta all'aumentare dello spessore del guscio.

#### 4.1.3 Guscio con lati incastrati/liberi $0^\circ/90^\circ/0^\circ/90^\circ$

L'ultima analisi che si è effettuata in questa prima parte fa riferimento ad un guscio incastrato nei due spigoli paralleli a  $\beta$  (si veda la figura 3.1) e gli altri due liberi. I risultati degli spostamenti a cui fare riferimento sono quelli riportati nella tabella 3.3.1. Fatta eccezione per le condizioni al contorno geometriche, gli altri dati sono uguali a quelli utilizzati nelle analisi precedenti.

Questa analisi è stata effettuata per mettere in evidenza l'influenza che hanno le condizioni al contorno geometriche sulla BTD. I risultati sono riportati nella figura 4.8 e nelle tabelle 4.9, 4.10 e 4.11.

#### 4.1.4 Casi particolari

Alla fine delle analisi, si è notato che alcuni casi trattati nelle sezioni 4.1.2 e 4.1.3 hanno comportamenti che si discostano dall'andamento generale delle altre analisi. Nello specifico, se si fa riferimento alle tabelle 4.6, 4.10 e 4.11 si nota che sono le uniche a non presentare il fattore RF pari ad 1.

Per questi casi allora si cambierà punto di studio della deformazione del guscio, attualmente posizionato nel centro del guscio, per monitorare l'andamento della soluzione. Nella figura 4.9 si possono vedere le coordinate dei punti 1 e 2 su cui si andranno a monitorare gli spostamenti lungo  $z$ .

Andando ad osservare le tabelle 4.12, 4.13 e 4.14 si nota come gli effetti dei primi termini dell'espansione continuino a non essere importanti nei casi a bassi gradi di libertà anche andando a variare il punto di osservazione dello spostamento lungo  $z$ . Nello specifico si nota che la tabella 4.12 è esattamente identica alla tabella 4.6 indice del fatto che anche avendo variato il punto di osservazione i termini di espansione importanti sono rimasti invariati. Negli altri 2 casi, invece, in generale cambia l'ordine dei termini che appaiono nella curva BTD anche se, come si nota confrontando la tabella 4.13 con la tabella 4.10, i rapporti RF rimangono pressoché invariati. Si precisa inoltre che per i punti di analisi 1 e 2, per via della simmetria, gli spostamenti sono risultati essere gli stessi e per tanto vengono riportati solo quelli del punto 1. Osservando nel complessivo i risultati si nota che il caso con laminazione non simmetrica del tipo  $0^\circ/90^\circ/0^\circ/90^\circ$  e raggio di curvatura  $R = 5$  è l'unico in cui i termini all'ordine zero dell'espansione non sono fondamentali per costruire una curva BTD. Si conclude quindi che se in questo caso avessi considerato sempre presenti i primi tre termini per ridurre il costo computazionale dell'analisi, come si è fatto in alcune analisi e si farà nel seguito, avrei ottenuto dei risultati peggiori. Quest'ultima analisi è dunque servita a confermare tale affermazione e far escludere, per un'analisi statica di questo tipo, l'utilizzo di soli 12 termini di espansione.

## 4.2 Risultati analisi dinamica

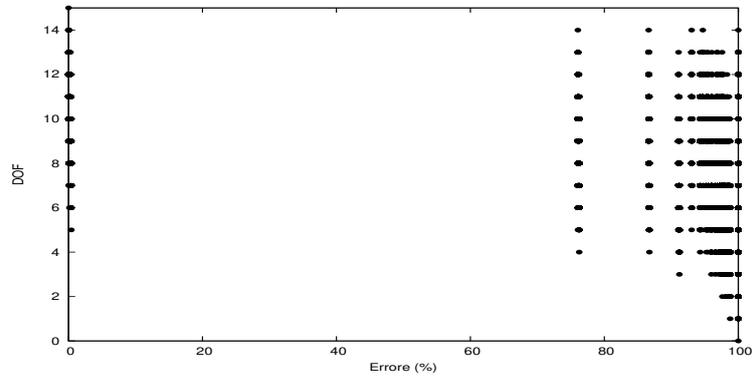
L'analisi effettuata in questa sezione fa ancora riferimento ad un guscio multistrato quadrato come quello riportato in figura 3.1, con le stesse proprietà del materiale e geometriche riportate nella sezione 3. Grazie alla simmetria del guscio, in questa parte dell'elaborato si utilizzerà interamente la condizione di simmetria con l'obiettivo di ridurre il costo computazionale dell'analisi. Inoltre, sempre con lo stesso intento, i termini costanti delle espansioni, ovvero  $u_{x_1}$ ,  $u_{y_1}$  e  $u_{z_1}$  verranno sempre considerati nel modello ridotto. Si può fare questo poichè come visto nella sezione 4.1 e in [19], a meno che in alcuni casi particolare, questi termini sono sempre presenti nel BTD. In questo modo il numero di gradi di libertà passa da 15 a 12 portando ad una drastica riduzione dei modelli da considerare (si passa da 32768 modelli a 4096) riducendo notevolmente il tempo per ogni analisi senza far perdere ai risultati finali informazioni rilevanti.

L'obiettivo di questa fase del training è famigliarizzare con il codice "MATLAB" che fornisce i risultati dell'analisi dinamica e replicare i risultati presenti in "MANCA CITAZIONE". Quello che verrà fatto è, per diverse condizioni geometriche, andare ad analizzare la curva BTD per le prime 10 frequenze e confrontare i modi attraverso il MAC. Il MAC è un parametro in grado di fornire qual è il livello di uguaglianza tra il modo di riferimento (in questo caso è il modo fornito dal modello  $N = 4$ ) ed il modo considerato. Se il valore di MAC è pari ad uno, allora i due modi confrontati sono uguali; quando è minore di uno si discostano tanto più MAC tende a zero. Questo tipo di confronto viene fatto per mezzo di una matrice cromatica in cui il colore nero indica modi uguali, colore bianco modi completamente diversi e le scale intermedie diversi gradi di uguaglianza.

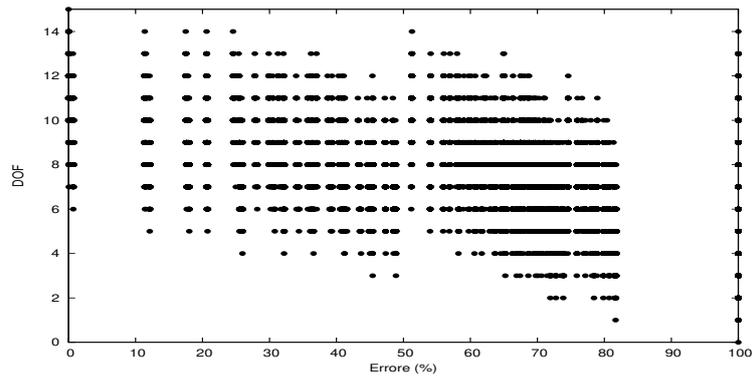
Altro importante fattore da considerare è la deviazione standard dalla media delle frequenze. Questo parametro viene tenuto in considerazione in quanto non ci si può ritenere soddisfatti di un risultato con errore sulla frequenza basso, ma una grande deviazione standard.

L'analisi effettuata è quella con raggio di curvatura  $R = 5$  e diversi spessori. La figura 4.10 riporta tutte le accuratèzze dei  $2^{12}$  modelli; ogni punto sulla griglia rappresenta una teoria e la sua precisione nel rilevare le prime dieci frequenze naturali. La figura 4.11 riporta invece le varie curve BTD per i diversi spessori confrontate con le teorie  $N = 2$ ,  $N = 3$ , FSDT PTD e TSDT.

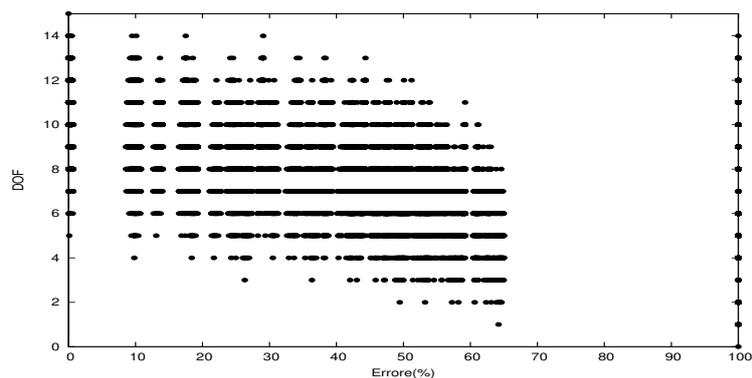
Per una più facile lettura, i casi 4.11b e 4.11d, sono stati riportati con la sola teoria  $N = 3$ . Nella figura 4.12 è invece riportata la deviazione standard dei vari casi. Infine, nelle tabelle 4.15, 4.16 e 4.17 sono riportati i termini che vanno a comporre le curve BTD. Le tabelle sono da leggersi come nei paragrafi precedenti e, avendo considerato sempre presenti i termini costanti delle espansioni, avranno sempre  $RF_0 = 1$ .



(a)  $h = 0.01$

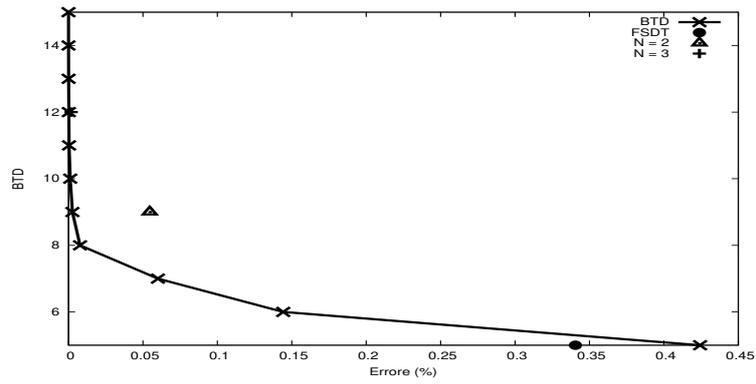


(b)  $h = 0.1$

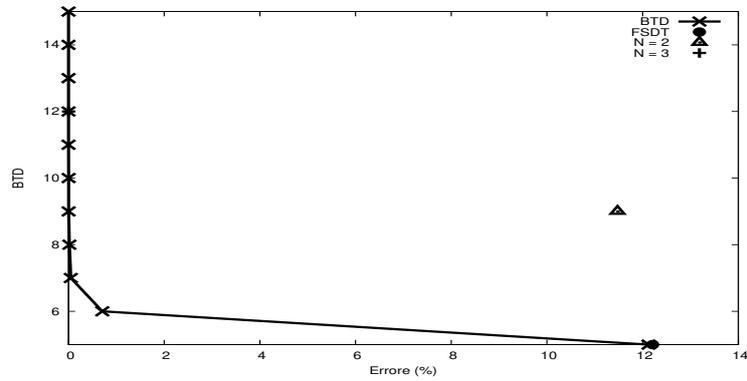


(c)  $h = 0.2$

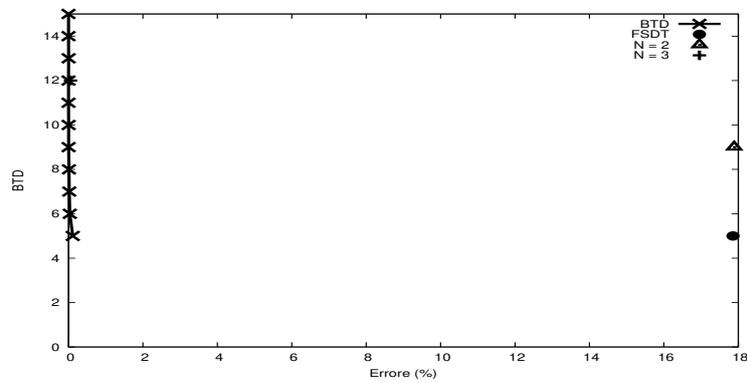
**Figura 4.1:** Tutti i risultati per  $R = 5$  e laminazione  $0^\circ/90^\circ/0^\circ$



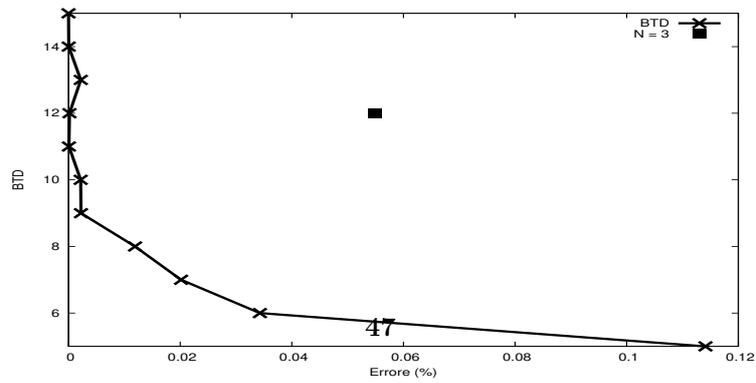
(a)  $h = 0.01$



(b)  $h = 0.1$



(c)  $h = 0.2$



(d)  $h = 0.2$ , senza FSDT e  $N = 2$

Figura 4.2: BTD per  $R = 5$  e laminazione  $0^\circ/90^\circ/0^\circ$

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	△	△	△	△
8	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	△	△	△	△
7	▲	▲	▲	▲	▲	▲	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.94			RF <sub>2</sub> = 0.64			RF <sub>3</sub> = 0.48			RF <sub>4</sub> = 0.27		

**Tabella 4.1:** Modello BTD per laminazione 0°/90°/0°, R = 5 e h = 0.01

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	▲	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	△
10	▲	▲	▲	▲	▲	△	▲	△	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.82			RF <sub>2</sub> = 0.58			RF <sub>3</sub> = 0.67			RF <sub>4</sub> = 0.27		

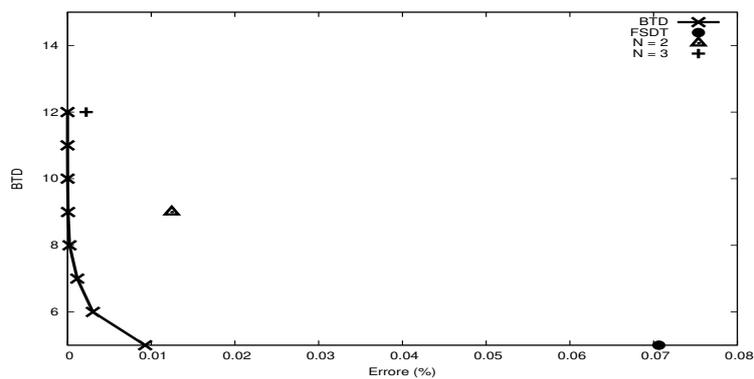
**Tabella 4.2:** Modello BTD per laminazione 0°/90°/0°, R = 5 e h = 0.1

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲	▲
12	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	▲	△	△
11	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	▲	△	△
10	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	△
9	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	▲	△
8	▲	▲	▲	△	▲	△	△	▲	▲	▲	△	△	△	▲	△
7	▲	▲	▲	△	▲	△	△	▲	▲	▲	△	△	△	△	△
6	▲	▲	▲	△	▲	△	△	△	▲	▲	△	△	△	△	▲
5	▲	▲	▲	△	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.45			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.39		

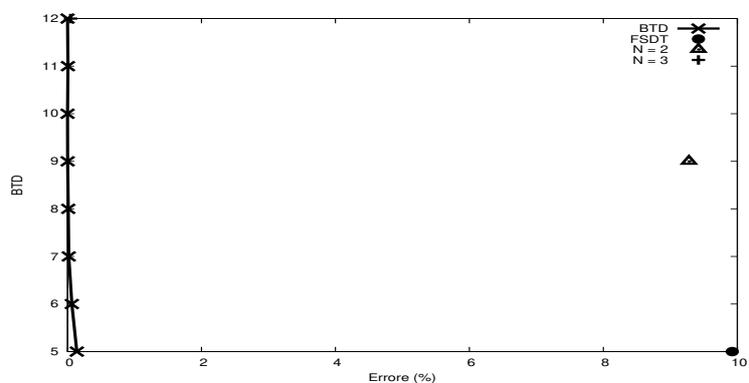
**Tabella 4.3:** Modello BTD per laminazione 0°/90°/0°, R = 5 e h = 0.2

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	▲	△	△
10	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.94			RF <sub>2</sub> = 0.61			RF <sub>3</sub> = 0.48			RF <sub>4</sub> = 0.30		

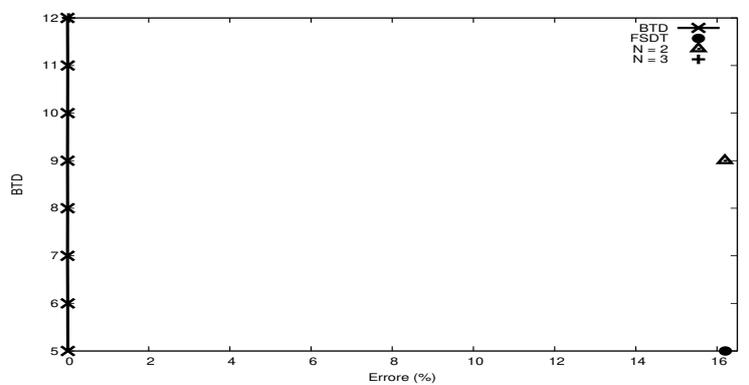
**Tabella 4.4:** Modello BTD per laminazione 0°/90°/0°/90°, R = 5 e h = 0.01



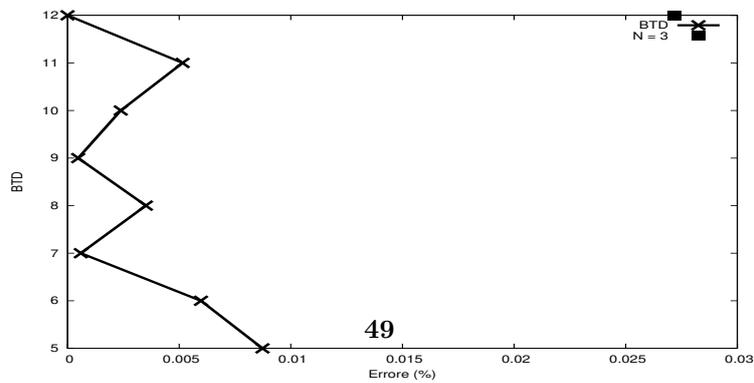
(a)  $h = 0.01$



(b)  $h = 0.1$

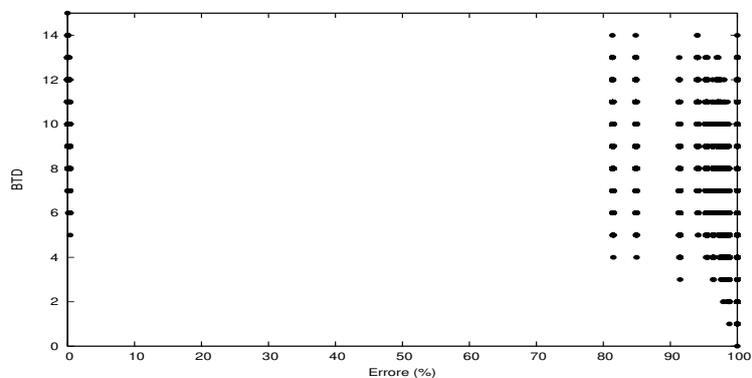


(c)  $h = 0.2$

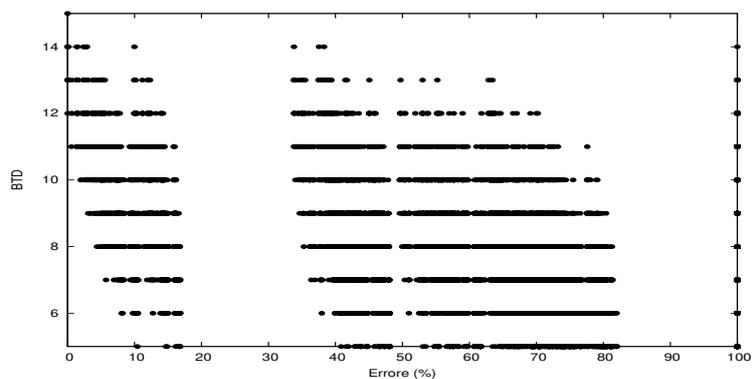


(d)  $h = 0.2$ , senza FSDT e  $N = 2$

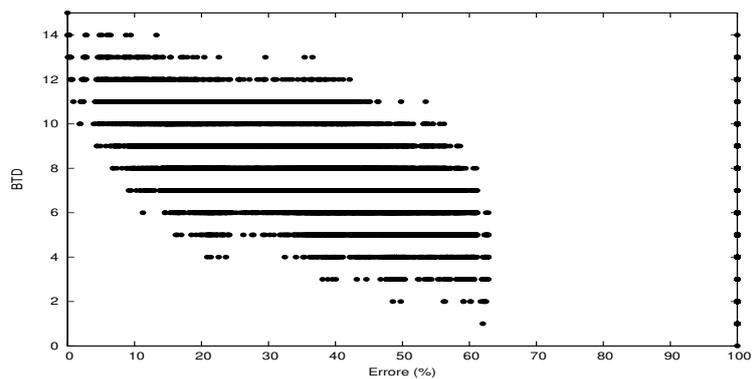
Figura 4.3: BTD per  $R = 2$  e laminazione  $0^\circ/90^\circ/0^\circ$



(a)  $h = 0.01$

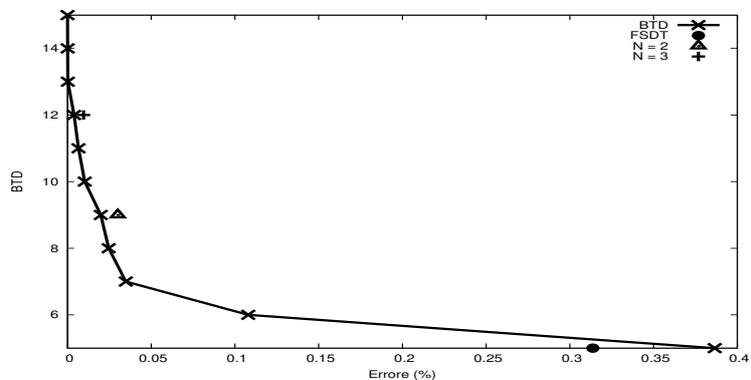


(b)  $h = 0.1$

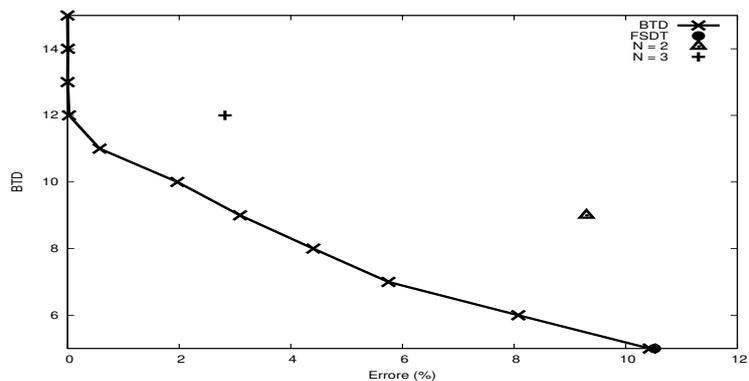


(c)  $h = 0.2$

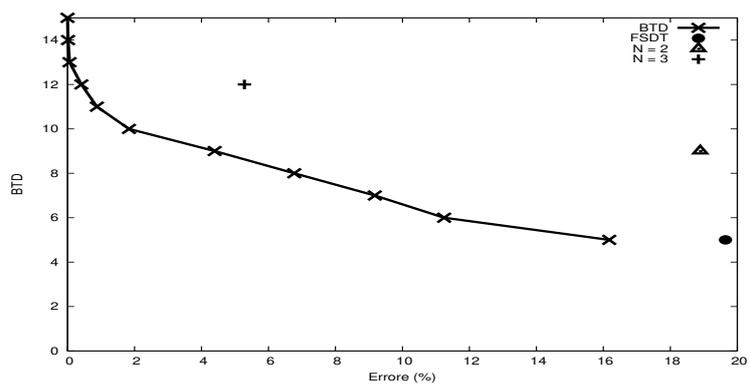
Figura 4.4: Tutti i risultati per  $R = 5$  e laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$



(a)  $h = 0.01$

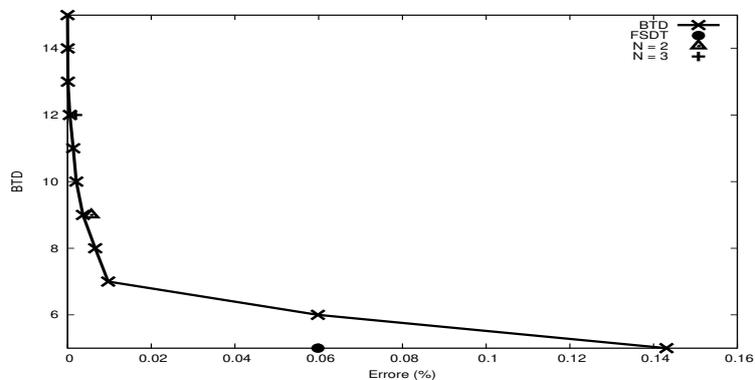


(b)  $h = 0.1$

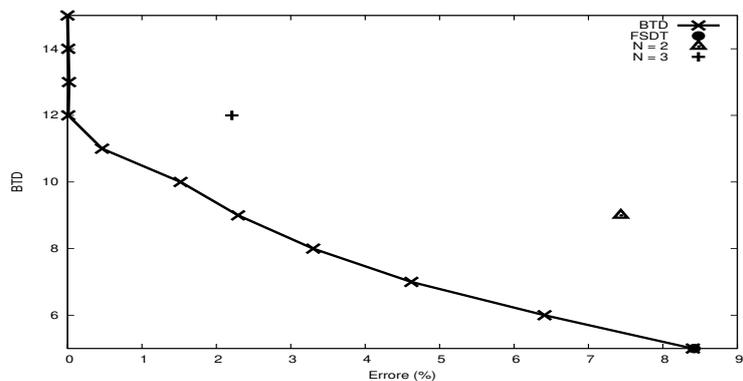


(c)  $h = 0.2$

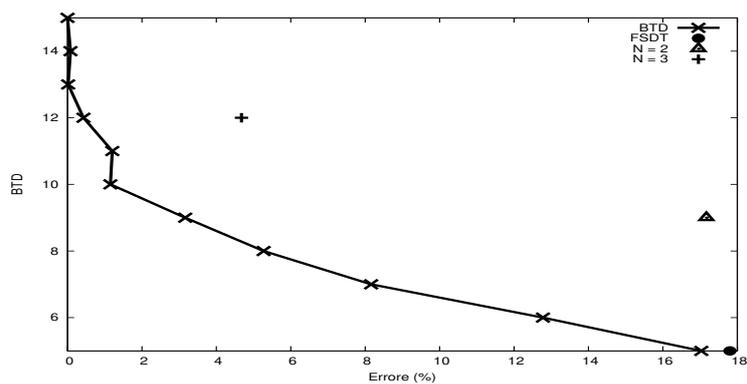
Figura 4.5: BTD per  $R = 5$  e laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$



(a)  $h = 0.01$

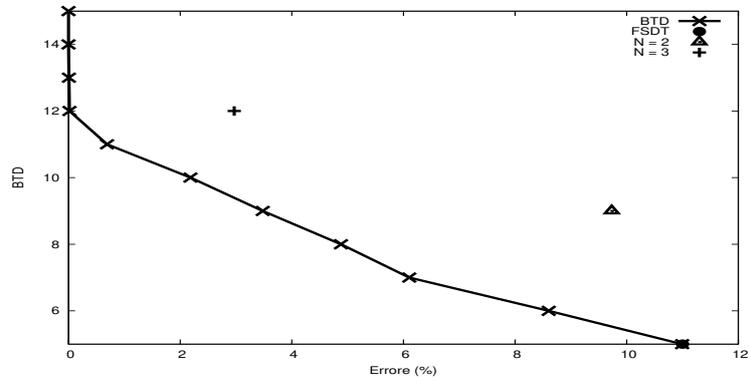


(b)  $h = 0.1$

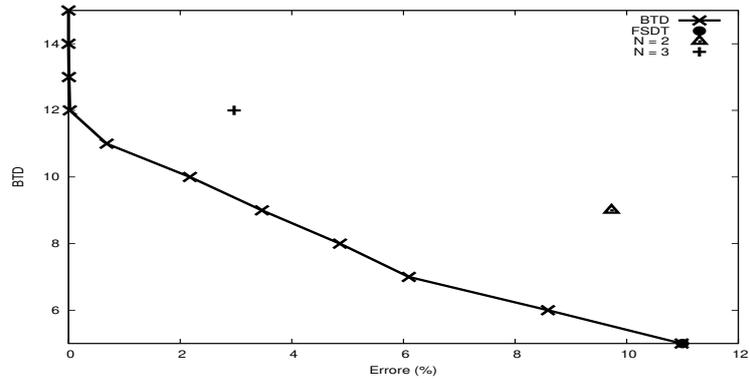


(c)  $h = 0.2$

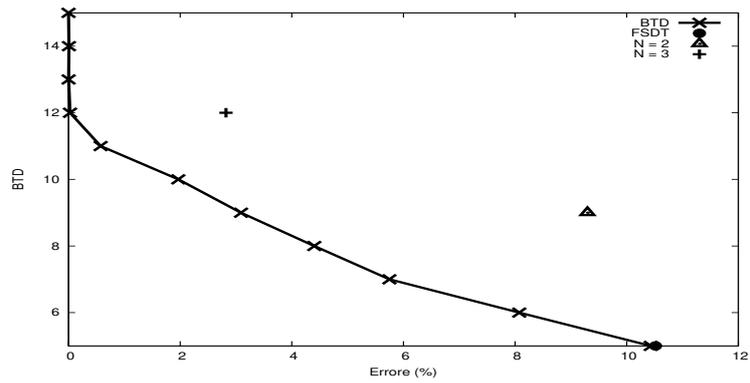
Figura 4.6: BTD per  $R = 2$  e laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$



(a)  $h = 0.01$



(b)  $h = 0.1$



(c)  $h = 0.2$

Figura 4.7: BTD per  $h = 0.1$  e laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
12	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	▲
11	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
10	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	△	▲	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	△	△
8	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	△	▲	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.76			RF <sub>2</sub> = 0.52			RF <sub>3</sub> = 0.64			RF <sub>4</sub> = 0.42		

**Tabella 4.5:** Modello BTM per laminazione 0°/90°/0°/90°, R = 5 e h = 0.1

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
12	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
10	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	△	▲	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	△	△
8	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	△	△	△
7	△	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	△	△
6	△	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
5	△	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 0.91			RF <sub>1</sub> = 0.79			RF <sub>2</sub> = 0.55			RF <sub>3</sub> = 0.67			RF <sub>4</sub> = 0.42		

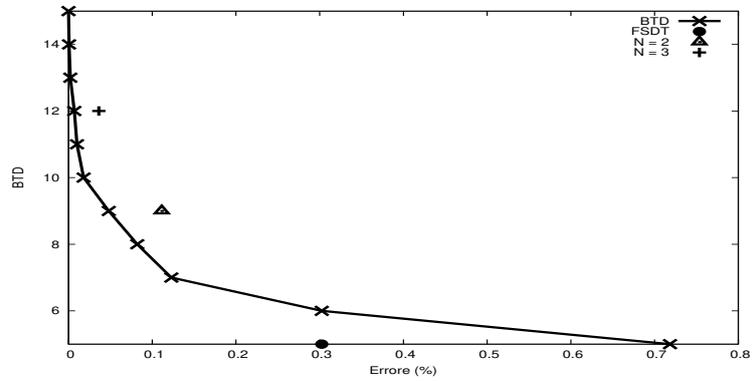
**Tabella 4.6:** Modello BTM per laminazione 0°/90°/0°/90°, R = 5 e h = 0.2

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
12	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
10	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	△	▲	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	△	△
8	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	△	▲	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.76			RF <sub>2</sub> = 0.55			RF <sub>3</sub> = 0.61			RF <sub>4</sub> = 0.42		

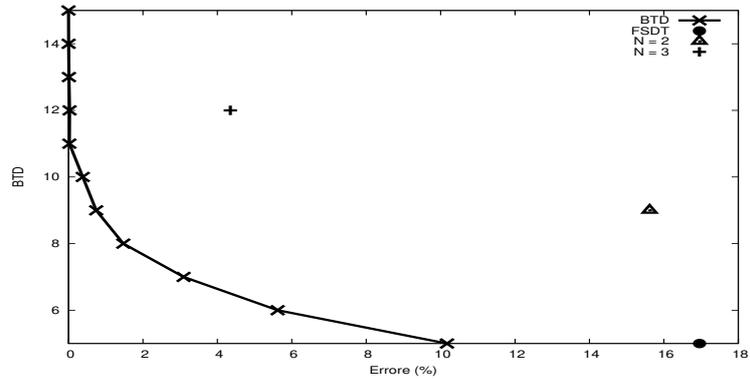
**Tabella 4.7:** Modello BTM per laminazione 0°/90°/0°/90°, R = 50 e h = 0.1

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲	▲
12	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
10	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	△	▲	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	△	△
8	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	△	▲	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.55			RF <sub>3</sub> = 0.61			RF <sub>4</sub> = 0.45		

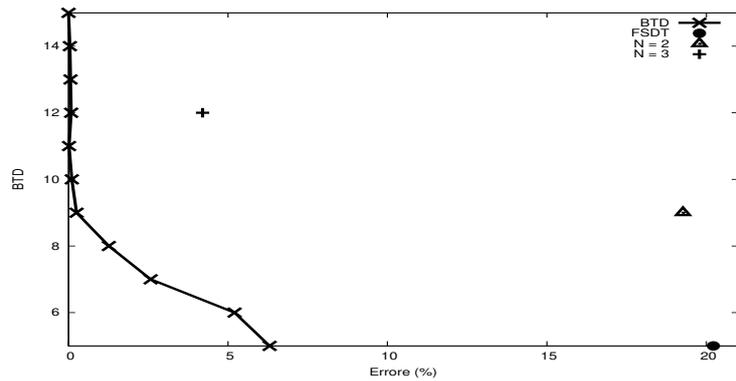
**Tabella 4.8:** Modello BTM per laminazione 0°/90°/0°/90°, R = 100 e h = 0.1



(a)  $h = 0.01$



(b)  $h = 0.1$



(c)  $h = 0.2$

Figura 4.8: BTD per guscio con 2 lati incastrati,  $R = 5$  e laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	△	△
11	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	▲	△	△
10	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△	△	▲	△	△
9	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△	△	▲	△	△
8	▲	▲	▲	▲	▲	▲	△	△	▲	▲	△	△	△	△	△
7	▲	▲	▲	▲	▲	▲	△	△	▲	△	△	△	△	△	△
6	▲	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.55			RF <sub>3</sub> = 0.61			RF <sub>4</sub> = 0.45		

**Tabella 4.9:** Modello BTM per guscio con 2 lati incastrati, laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$ ,  $R = 5$  e  $h = 0.01$

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
13	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲
12	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲	▲	▲	△
11	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
10	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	△	△
8	▲	▲	▲	▲	▲	△	▲	△	△	▲	△	△	▲	△	△
7	△	▲	▲	▲	▲	△	▲	△	△	▲	△	△	▲	△	△
6	△	▲	▲	▲	▲	△	▲	△	△	▲	△	△	▲	△	△
5	△	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 0.91			RF <sub>1</sub> = 0.76			RF <sub>2</sub> = 0.55			RF <sub>3</sub> = 0.64			RF <sub>4</sub> = 0.48		

**Tabella 4.10:** Modello BTM per guscio con 2 lati incastrati, laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$ ,  $R = 5$  e  $h = 0.1$

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
12	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△
11	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△
10	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	▲	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	△	△
8	▲	▲	▲	▲	△	△	▲	△	△	▲	▲	△	▲	△	△
7	▲	▲	▲	▲	△	△	▲	△	△	▲	▲	△	▲	△	△
6	▲	▲	▲	△	△	△	▲	△	△	▲	▲	△	△	△	△
5	△	▲	▲	△	△	△	▲	△	△	▲	▲	△	△	△	△
	RF <sub>0</sub> = 0.97			RF <sub>1</sub> = 0.61			RF <sub>2</sub> = 0.58			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 4.11:** Modello BTM per guscio con 2 lati incastrati, laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$ ,  $R = 5$  e  $h = 0.2$

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
12	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
10	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	△	△
8	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	△	▲	△	△
7	△	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	△	△
6	△	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
5	△	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 0.91			RF <sub>1</sub> = 0.79			RF <sub>2</sub> = 0.55			RF <sub>3</sub> = 0.67			RF <sub>4</sub> = 0.42		

**Tabella 4.12:** Modello BTM per laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$ ,  $R = 5$  e  $h = 0.2$ . Caso nel punto 1

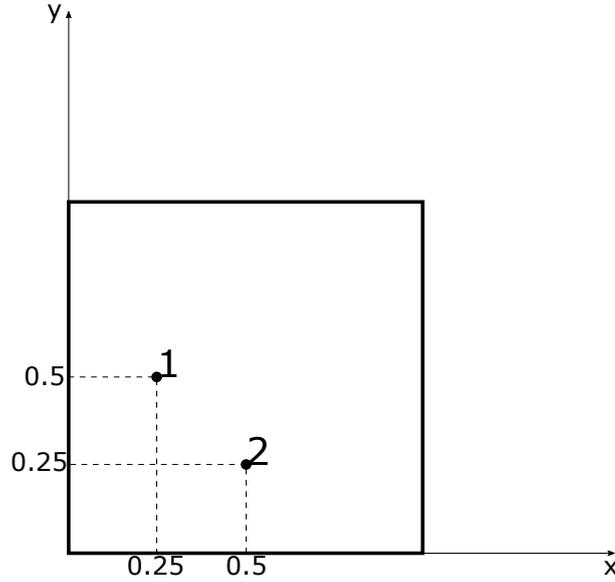


Figura 4.9: Coordinate casi particolari

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
13	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△
12	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△
11	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲	▲	▲	△
10	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲	▲	△	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	▲	▲	△	△
8	▲	▲	▲	▲	▲	△	▲	△	△	▲	△	▲	▲	△	△
7	△	▲	▲	▲	▲	△	▲	△	△	▲	△	▲	▲	△	△
6	△	▲	▲	▲	▲	△	▲	△	△	▲	△	▲	▲	△	△
5	△	▲	▲	▲	▲	△	△	△	△	▲	△	▲	▲	△	△
	RF <sub>0</sub> = 0.91			RF <sub>1</sub> = 0.79			RF <sub>2</sub> = 0.55			RF <sub>3</sub> = 0.64			RF <sub>4</sub> = 0.45		

Tabella 4.13: Modello BTD per guscio con 2 lati incastrati, laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$ ,  $R = 5$  e  $h = 0.1$ . Caso nel punto 1

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
12	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲	▲	△	▲
11	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲	▲	△	▲
10	▲	▲	▲	△	▲	△	▲	▲	△	▲	▲	▲	▲	△	△
9	▲	▲	▲	▲	▲	△	▲	△	△	▲	▲	▲	▲	△	△
8	▲	▲	▲	△	▲	△	▲	△	△	▲	▲	▲	▲	△	△
7	▲	▲	▲	△	△	△	▲	△	△	▲	▲	▲	▲	△	△
6	△	▲	▲	△	△	△	▲	△	△	▲	▲	▲	▲	△	△
5	△	▲	▲	△	△	△	▲	△	△	▲	▲	▲	▲	△	△
	RF <sub>0</sub> = 0.94			RF <sub>1</sub> = 0.55			RF <sub>2</sub> = 0.61			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.48		

Tabella 4.14: Modello BTD per guscio con 2 lati incastrati, laminazione  $0^\circ/90^\circ/0^\circ/90^\circ$ ,  $R = 5$  e  $h = 0.1$ . Caso nel punto 2

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	△	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.88			RF <sub>2</sub> = 0.61			RF <sub>3</sub> = 0.58			RF <sub>4</sub> = 0.27		

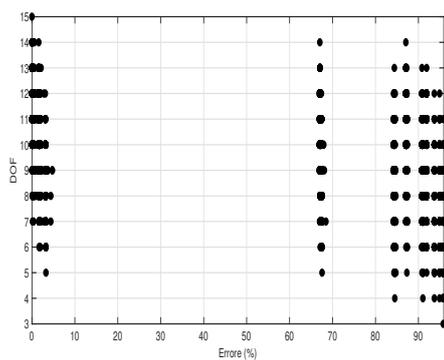
**Tabella 4.15:** Modello BTD analisi dinamica, laminazione 0°/90°/0°, R = 5 e h = 0.01.

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	▲	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	▲	▲	▲
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	▲
10	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	▲	▲
9	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	▲	△
8	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	▲
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.79			RF <sub>2</sub> = 0.40			RF <sub>3</sub> = 0.64			RF <sub>4</sub> = 0.51		

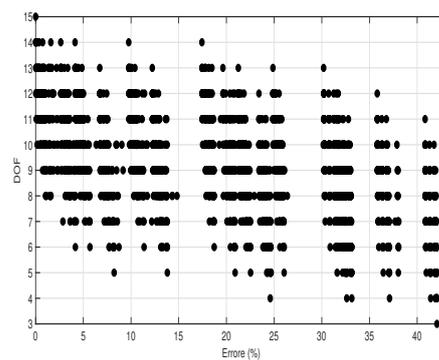
**Tabella 4.16:** Modello BTD analisi dinamica, laminazione 0°/90°/0°, R = 5 e h = 0.1.

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲
13	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲	▲
12	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	▲	△	▲	▲
11	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	▲	△	▲	▲
10	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	▲	▲
9	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	▲	△
8	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

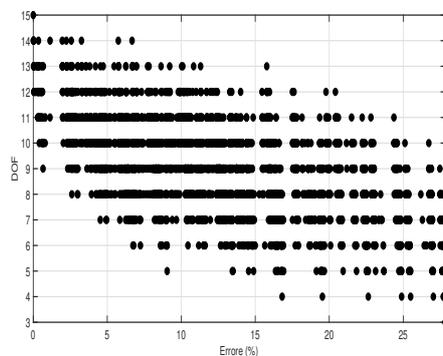
**Tabella 4.17:** Modello BTD analisi dinamica, laminazione 0°/90°/0°, R = 5 e h = 0.2.



(a)  $h = 0.01$

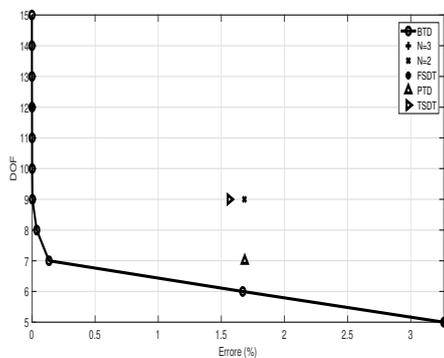


(b)  $h = 0.1$

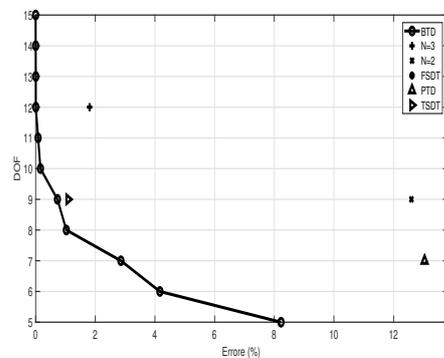


(c)  $h = 0.2$

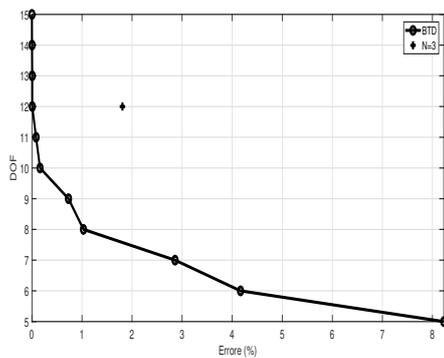
**Figura 4.10:** Tutti i risultati per  $R = 5$  e laminazione  $0^\circ/90^\circ/0^\circ$



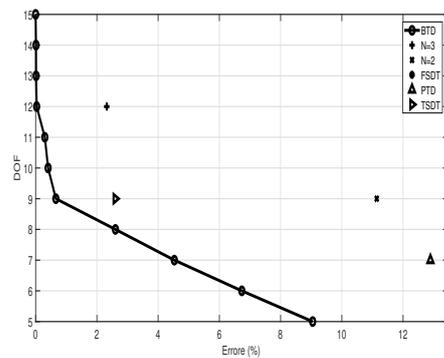
(a)  $h = 0.01$



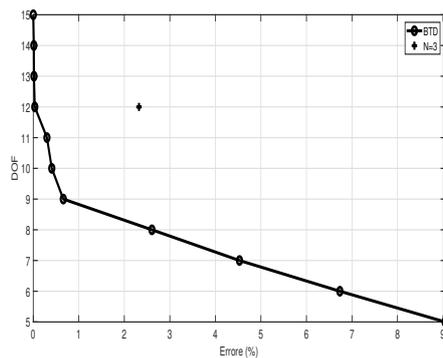
(b)  $h = 0.1$



(c)  $h = 0.1$  con  $N = 3$



(d)  $h = 0.2$



(e)  $h = 0.2$  con  $N = 3$

Figura 4.11: BTd per  $R = 5$  e laminazione  $0^\circ/90^\circ/0^\circ$

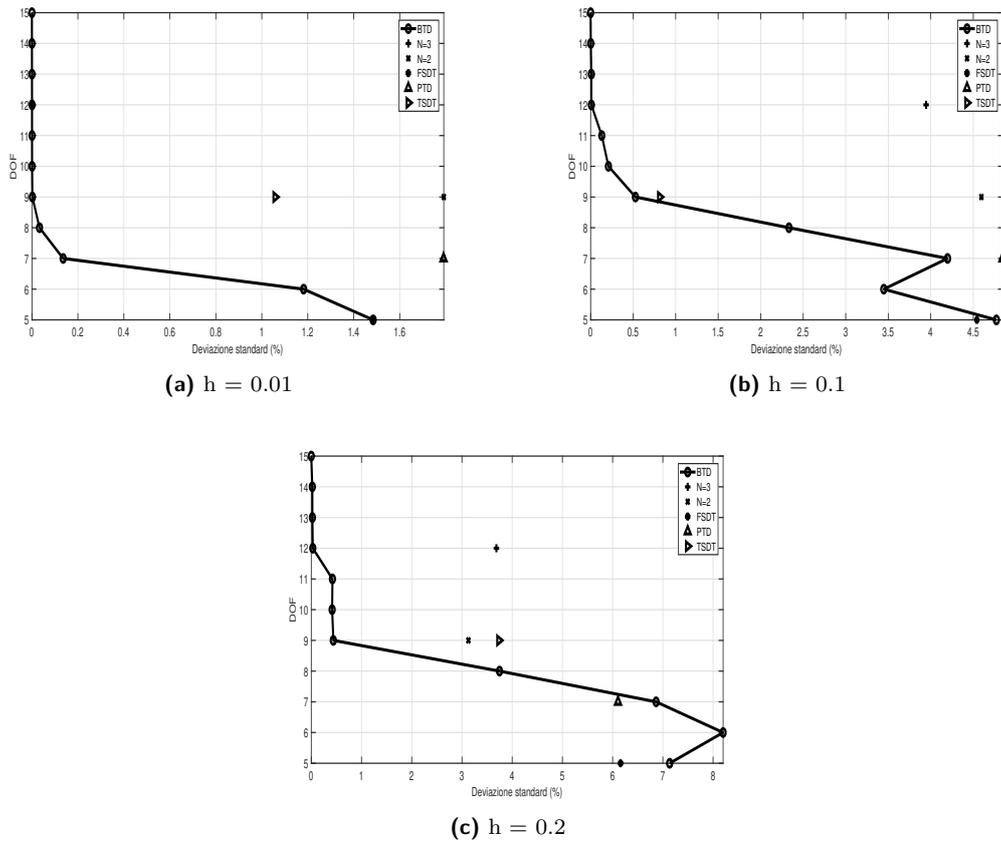


Figura 4.12: Deviazione standard per  $R = 5$  e laminazione  $0^\circ/90^\circ/0^\circ$



## Capitolo 5

# Valutazione dell'errore BTD

L'obiettivo di questa parte di elaborato nasce a seguito della constatazione che il codice utilizzato nella sezione 4.2 confonde, per alcuni casi, il modo di vibrare di due frequenze vicine. Questo nasce dal fatto che spesso, se non sempre, due modi di vibrare adiacenti, hanno uno scarto in termini di Hertz molto piccolo e tale per cui il codice può confondere i due modi ed invertirli. Questo fatto è ben visibile nelle tabelle del MAC in cui sulla diagonale principale si dovrebbero visualizzare i quadratini neri ad indicare che il modo con il modello ad  $N = 4$  e il modo con il modello ridotto sono uguali. Analizzando i risultati si nota però che non sempre è la diagonale a presentare i quadratini neri.

Altro effetto che si vuole provare a contrastare è la presenza di modi relativi a frequenze anche molto diverse che si assomigliano, fatto che porta ad un grafico del MAC non preciso e ad una valutazione della bontà del modello ridotto più difficile.

Si è pensato quindi di andare a modificare l'attuale valutazione dell'errore che vige sul codice MatLab. Fino a questo momento l'errore che veniva restituito era relativo ad una singola frequenza del modo preso in considerazione nel modello ridotto rispetto alla frequenza dello stesso modo nel modello con  $N = 4$ , ovvero:

$$\epsilon = \frac{f_i^{N=4} - f_i^*}{f_i^{N=4}}$$

In questo modo si è in qualche modo vincolati alla frequenza di ogni modo di vibrare. L'idea è stata proprio quella di andare a svincolarsi da questa dipendenza dal modo di vibrare.

L'errore medio, nel nuovo codice, verrà quindi valutato con la seguente relazione:

$$\frac{\sum f_i^*}{\sum f_i^{N=4}}$$

La parte di codice in cui viene calcolato l'errore è riportato di seguito:

---

```
1 ...
2 A = importdata('BTD.dat');
3 B = importdata('FREQUENCIES.dat');
4 N4 = sum(B);
5 for jj = 1:2^tau
6 for j = 1:nmodes
7 A(1,2+tau+j) = 0;
8 fstar(jj,j) = B(j)-((A(jj,2+tau+j)*B(j))/100);
9 MAC(jj,j,:) = A(jj,(3+tau+nmodes+(j-1)*nmodes):(2+tau+nmodes+(j)*nmodes));
10 MAC_mean(jj) = MAC_mean(jj) + MAC(jj,j,j)/nmodes;
11 f_mean(jj) = f_mean(jj)+fstar(jj,j);%/nmodes;
```

```

12 end
13 f_mean(jj) = abs((1-f_mean(jj)/N4)*100);
14 ...

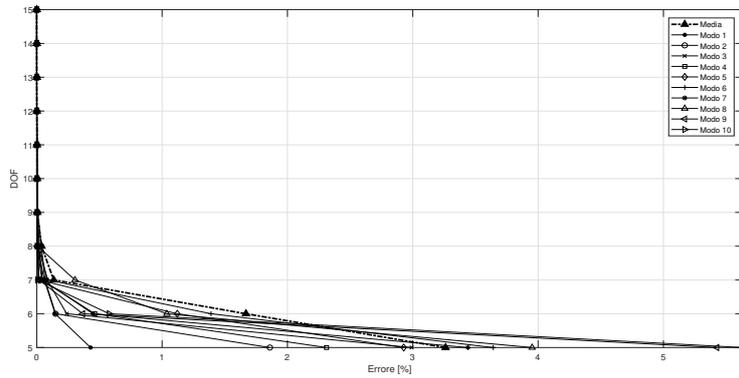
```

Una seconda fase di analisi dell'errore ha portato a concentrarsi sul fatto che la Best Theory Diagram viene tracciata tenendo conto della media dell'errore dovuta ai modi analizzati (finora sono stati sempre considerati 10 modi). Questo accorgimento risulta molto comodo in quanto si ha una sola tabella ed un solo grafico che sono di facile lettura, ma non si è in grado di vedere se un modo ha un grande errore rispetto ad un altro. A tal proposito viene nel seguito presentato un confronto tra le Best Theory Diagram per ogni modo di vibrare e, tenendo conto dell'errore medio, si analizza quali sono i modi che portano ad un grande aumento dell'errore.

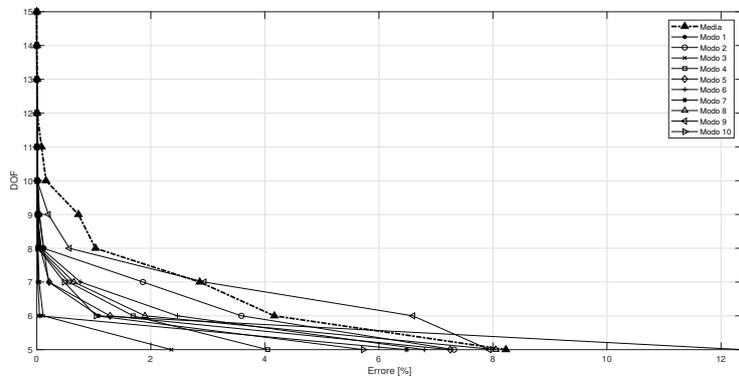
Per fare questa analisi si utilizzano i risultati ottenuti nella sezione 4.2. Analizzando i grafici si può osservare che non c'è una legge univoca che descriva il comportamento dei vari modi infatti si può notare che nella figura 5.1a l'errore massimo per il modello a cinque gradi di libertà è dato dal decimo modo di vibrare mentre nella figura 5.1b si nota come sia dato dal primo modo. La stessa imprevedibilità dell'errore avviene anche per gli altri modelli a numero maggiore di gradi di libertà attivi. In tutti i casi riportati in figura 5.1 si può notare un errore molto basso per modelli con un numero pari o superiore ai dieci gradi di libertà. Ciò che appare evidente è come aumentando lo spessore, per tutti i modi, è necessario un numero sempre maggiore di gradi di libertà poiché l'errore sia vicino allo zero. Infatti, nella figura 5.1a, che fa riferimento ad un guscio con spessore  $h = 0.01$ , per otto gradi di libertà l'errore prodotto da tutti i modi è vicino allo zero e con sette l'errore prodotto dal modo 8 è ancora sotto all'unità percentuale. Nella figura 5.1c, in questo caso si fa riferimento ad un guscio di spessore  $h = 0.2$ , per otto gradi di libertà si ha un errore dato dal modo 10 maggiore dell'unità percentuale, mentre per sette gradi di libertà sempre il modo 10 produce un errore del 6%. Osservando questo confronto si può affermare che l'importanza dei termini di ordine superiore cresce all'aumentare dello spessore e che quindi per uno spessore elevato non è possibile utilizzare modelli molto ridotti e questo indipendentemente dal modo di vibrare che si vuole studiare. Infatti, non essendo presente una legge valida per cui sia possibile predirre qual'è il modo che dà l'errore più alto con un dato numero di gradi di libertà, bisogna attenersi al fatto che mediamente questo aumenta all'aumentare dello spessore.

Osservando ancora la figura 5.1 ed in particolare la figura 5.2 si nota che per ogni caso è presente l'andamento della curva BTD ottenuta con la media degli errori. Ciò che si osserva è che, mediamente, la curva ha errori più alti di quelli prodotti dal singolo modo di vibrare. Infatti, dove l'errore diventa consistente (ovvero si passa da un errore  $\ll 1$  a  $< 1$ ) si osserva chiaramente la curva generale stare a destra (quindi ad errori più elevati) rispetto al fascio delle curve dei singoli modi. Questa tendenza rimane vera fino a sei gradi di libertà, mentre per il modello che ne presenta solo cinque l'errore dato dalla media dei modi si posiziona circa nel mezzo. In prima analisi ci si potrebbe aspettare che, facendo la media degli errori ottenuti per i 10 modi ad ogni numero di grado di libertà, tale errore fosse proprio il valore riportato sulla curva della BTD media. Questo non si verifica poiché ogni modo ha, nelle tabelle e nei grafici riportati, i propri modelli ridotti migliori. Per poter accomodare i 10 modi, il modello globale inevitabilmente non può prendere di ognuno di questi 10 il migliore e quindi l'errore sale.

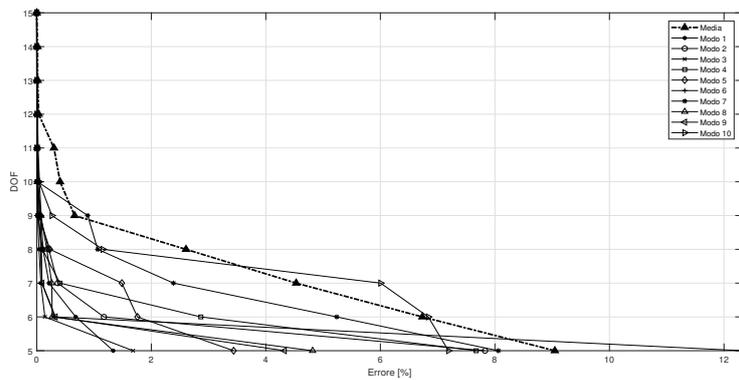
L'ultima osservazione viene fatta confrontando i grafici nella figura 5.1 con quelli in figura 4.11. Andando a concentrarsi con i gli errori percentuali relativi ai modelli con solo cinque gradi di libertà attivi si può notare che in tutti e tre i casi vi è un solo modello che alza di molto la media degli errori di tutti gli altri modelli. Infatti nel caso in figura 5.1a il modo 10 ha un errore circa del 6% contro alla media di tutti i modi che è circa del 3.5%. Nel caso in figura 5.1b questo fenomeno è ancora più evidente in quanto l'errore del modo 1 è superiore al 12% contro la media dei tutti i modi che è circa dell' 8%. Infine, nella figura 5.1c l'errore del modo 6 è superiore al 12% invece quello medio del 9%.



(a)  $h = 0.01$



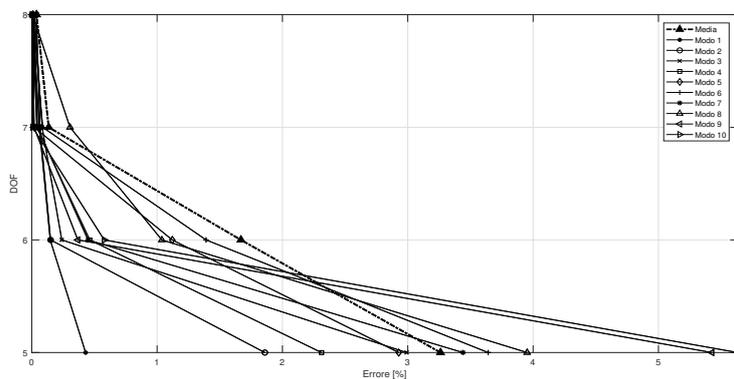
(b)  $h = 0.1$



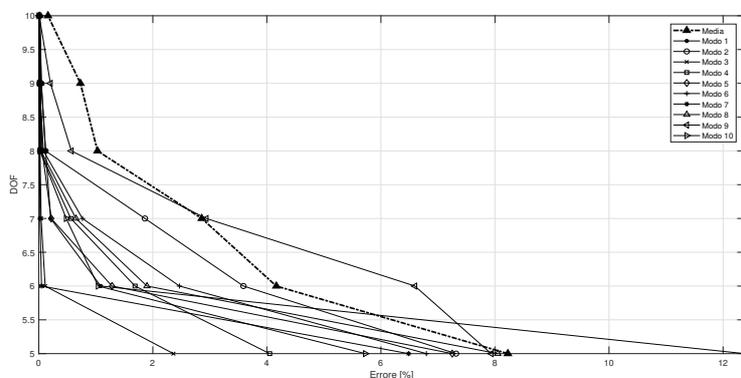
(c)  $h = 0.2$

**Figura 5.1:** Confronti tra i modi per  $R = 5$

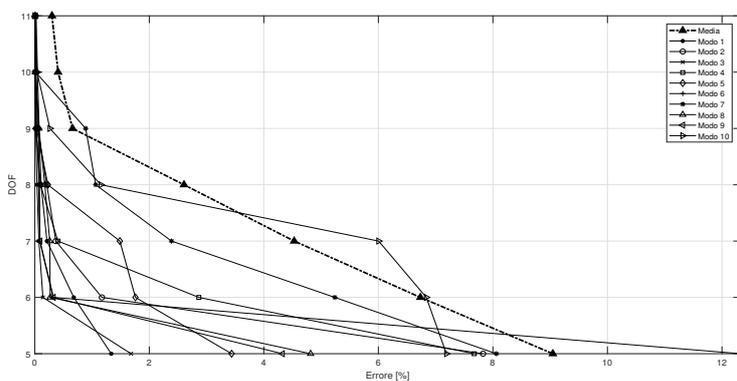
A questo punto è interessante andare a confrontare le tabelle delle Best Theory Diagram ottenute nel paragrafo 4.2 con quelle ottenute per ognuno dei 10 modi nei tre casi analizzati.



(a)  $h = 0.01$



(b)  $h = 0.1$



(c)  $h = 0.2$

**Figura 5.2:** Particolari per modelli a pochi gradi di libertà

Verranno confrontati i casi che riportano maggiore differenza tra gli andamenti dell'errore in funzione dei gradi di libertà. Il primo caso di interesse è quello per  $h = 0.1$  in cui si confrontano

i modi 1 e 9, riportati nelle tabelle 5.11 e 5.19 con il Best Theroy Diagram riferito alla media degli errori di tutti i modi riportato nella tabella 4.16. Si nota che l'andamento del modo 1 è del tutto simile a quello della media dei modi escluso il caso con cinque gradi di libertà in cui l'errore del modo 1 ha un valore molto più alto della media con cinque gradi di libertà. Se si osservano i termini attivi nel modello BTD del modo 1 si nota che il termine  $u_{x_4}$  è spento a differenza della media in cui è ancora attivo a spese del termine  $u_{x_2}$ . Per questo modo, quindi, non è possibile utilizzare efficacemente un modello ridotto a cinque gradi di libertà. Se si osservano poi gli altri modelli a gradi di libertà più elevati si nota il modo 1 ha errori più bassi della media di tutti i modi. Evidentemente, però, questa scelta di termini attivi, per i modi superiori al primo da mediamente errori elevati e quindi non viene scelto dal codice come modello BTD globale. Per quanto riguarda il modo 9 invece il discorso è diverso in quanto esso ha un andamento dell'errore completamente diverso e peggiore, per quanto riguarda i bassi gradi di libertà, del modello globale. Nonostante questo è interessante notare come per cinque gradi di libertà i modelli ridotti siano identici.

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
13	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	△
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	△	▲	▲	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	△	△	▲	△
9	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	△	△	△	△
8	▲	▲	▲	▲	▲	▲	△	△	▲	▲	△	△	△	△	△
7	▲	▲	▲	▲	▲	▲	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.1:** Modello BTD h = 0.01, modo 1

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	△	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	▲	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.2:** Modello BTD h = 0.01, modo 2

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	△	▲
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	△	△
11	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	△	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.3:** Modello BTD  $h = 0.01$ , modo 3

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	△	▲	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	▲	△	▲	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	▲	△	▲	△
9	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	▲	△	△	△
8	▲	▲	▲	▲	▲	▲	△	△	▲	▲	△	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.4:** Modello BTD  $h = 0.01$ , modo 4

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	△
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	▲	▲	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	▲	△	△	▲	▲	△	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.5:** Modello BTD  $h = 0.01$ , modo 5

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	△
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	▲	▲	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.6:** Modello BTD  $h = 0.01$ , modo 6



DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	△	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△
5	▲	▲	▲	▲	▲	△	△	△	△	△	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.11:** Modello BTM h = 0.1, modo 1

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	△	▲
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	▲
10	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	▲
9	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	▲
8	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.12:** Modello BTM h = 0.1, modo 2

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	▲
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	▲
10	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	▲
9	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	△	△	△	▲
8	▲	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	▲
7	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	▲
6	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.13:** Modello BTM h = 0.1, modo 3

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	▲	▲	▲	▲
11	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	△	▲	▲	▲
10	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	△	▲	▲	▲
9	▲	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	▲	▲	▲
8	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	▲	▲	▲
7	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	▲	▲	△
6	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	▲	▲	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	▲	▲	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.14:** Modello BTM h = 0.1, modo 4



DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	▲
11	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	△	△	▲	▲
10	▲	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	▲	▲
9	▲	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	▲	△
8	▲	▲	▲	▲	▲	△	△	▲	△	▲	△	△	△	▲	△
7	▲	▲	▲	▲	▲	△	△	▲	△	▲	△	△	△	△	△
6	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	△	▲	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.19:** Modello BTM h = 0.1, modo 9

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲
12	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	▲	△	▲
11	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	△	△	▲
10	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	▲
9	▲	▲	▲	△	▲	▲	△	△	▲	▲	▲	△	△	△	▲
8	▲	▲	▲	△	▲	△	△	△	▲	▲	▲	△	△	△	▲
7	▲	▲	▲	△	▲	△	△	△	△	▲	▲	△	△	△	▲
6	▲	▲	▲	△	▲	△	△	△	△	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	▲	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.20:** Modello BTM h = 0.1, modo 10

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	△	▲
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	▲
10	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	△	△	△	▲
9	▲	▲	▲	△	▲	▲	△	▲	△	▲	▲	△	△	△	▲
8	▲	▲	▲	△	▲	▲	△	△	△	▲	▲	△	△	△	▲
7	▲	▲	▲	△	▲	△	△	△	△	▲	▲	△	△	△	▲
6	▲	▲	▲	△	▲	△	△	△	▲	▲	△	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.21:** Modello BTM h = 0.2, modo 1

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲
13	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲
12	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲
11	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	▲
10	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	▲
9	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	▲
8	▲	▲	▲	△	▲	△	△	△	▲	▲	▲	△	△	△	▲
7	▲	▲	▲	△	▲	△	△	△	△	▲	▲	△	△	△	▲
6	▲	▲	▲	△	▲	△	△	△	△	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	▲	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.22:** Modello BTM h = 0.2, modo 2

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	▲
11	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	△	△	▲	▲
10	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	△	△	△	▲
9	▲	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	▲
8	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	▲
7	▲	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	▲
6	▲	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△
5	▲	▲	▲	▲	△	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.23:** Modello BTM h = 0.2, modo 3

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	▲	▲
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	△	▲
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	▲
10	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	▲
9	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	▲
8	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	▲
7	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.24:** Modello BTM h = 0.2, modo 4

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲
13	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲
12	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	▲	△	▲
11	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	△	△	△	▲
10	▲	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	▲
9	▲	▲	▲	▲	▲	△	△	△	▲	▲	▲	△	△	△	▲
8	▲	▲	▲	△	▲	△	△	△	▲	▲	▲	△	△	△	▲
7	▲	▲	▲	△	▲	△	△	△	▲	▲	▲	△	△	△	▲
6	▲	▲	▲	△	▲	△	△	△	▲	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	▲	▲	▲	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.25:** Modello BTM h = 0.2, modo 5

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲
12	▲	▲	▲	△	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲
11	▲	▲	▲	△	▲	△	▲	▲	▲	△	△	▲	▲	▲	▲
10	▲	▲	▲	▲	▲	△	▲	▲	△	△	△	▲	▲	▲	△
9	▲	▲	▲	△	▲	△	△	▲	▲	△	△	△	▲	▲	▲
8	▲	▲	▲	△	▲	△	△	▲	▲	△	△	△	△	▲	▲
7	▲	▲	▲	△	▲	△	△	▲	△	△	△	▲	△	▲	△
6	▲	▲	▲	△	▲	△	△	▲	△	△	△	△	△	▲	△
5	▲	▲	▲	△	▲	△	△	▲	△	△	△	△	△	▲	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.26:** Modello BTM h = 0.2, modo 6

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	▲
12	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	△	▲
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	▲
10	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	△	▲
9	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	△	▲
8	▲	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	△	△
7	▲	▲	▲	△	△	△	△	△	▲	▲	▲	△	△	▲	△
6	▲	▲	▲	△	△	△	△	▲	△	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	△	△	▲	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.27:** Modello BTM h = 0.2, modo 7

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	▲
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	▲
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	▲
9	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	△	▲
7	▲	▲	▲	▲	▲	▲	△	△	△	▲	▲	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
5	▲	▲	▲	△	▲	△	△	△	△	▲	▲	△	△	△	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.28:** Modello BTM h = 0.2, modo 8

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲
13	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲
12	▲	▲	▲	▲	▲	▲	▲	▲	△	▲	△	▲	▲	▲	△
11	▲	▲	▲	△	▲	▲	▲	▲	△	▲	△	▲	▲	▲	△
10	▲	▲	▲	△	▲	△	▲	▲	△	▲	△	▲	▲	▲	△
9	▲	▲	▲	△	▲	△	▲	▲	△	△	△	▲	▲	▲	△
8	▲	▲	▲	△	▲	▲	△	▲	△	△	△	▲	△	▲	△
7	▲	▲	▲	△	▲	△	△	▲	△	△	△	▲	△	▲	△
6	▲	▲	▲	△	△	△	△	▲	△	▲	△	△	△	▲	△
5	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△	▲	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.29:** Modello BTM h = 0.2, modo 9

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
13	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	△	△	▲	▲
12	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	▲
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	▲	△
10	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	▲	▲
9	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	▲	△
8	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	▲	△
7	▲	▲	▲	△	△	△	△	▲	▲	▲	▲	△	△	▲	△
6	▲	▲	▲	△	▲	△	△	△	▲	▲	▲	△	△	▲	△
5	▲	▲	▲	△	▲	△	△	△	▲	▲	▲	△	△	▲	△
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.73			RF <sub>2</sub> = 0.42			RF <sub>3</sub> = 0.76			RF <sub>4</sub> = 0.42		

**Tabella 5.30:** Modello BTM h = 0.2, modo 10

## Capitolo 6

# Sandwich

In questo capitolo si andranno a trattare, così come fatto per i laminati, gusci e piastre sandwich. Per prima cosa si andranno a confrontare le soluzioni analitiche che si trovano in [10] con le soluzioni fornite dal "MUL2 code" per verificare quanto la soluzione approssimata dal programma si avvicini a quella in forma chiusa riportata sulla pubblicazione.

In [10] viene utilizzato il seguente campo di spostamenti:

$$\begin{aligned}u &= u_0 + z\theta_x + z^2u_0^* + z^3\theta_x^* \\v &= v_0 + z\theta_y + z^2v_0^* + z^3\theta_y^* \\w &= w_0 + C_1z\theta_z + C_1z^2w_0^* + C_2z^3\theta_z^*\end{aligned}\tag{6.1}$$

Dove quando  $C_1 = C_2 = 1$  il modello di spostamenti ha 12 gradi di libertà e viene definito "HOST 12" (Higher-Order Shear deformation Theory con 12 gradi di libertà), corrisponde al campo di spostamenti ad  $N = 3$ , mentre se  $C_2 = 0$  il campo di spostamento diventa "HOST 11" e, infine, se  $C_1 = C_2 = 0$  il campo di spostamenti diventa "HOST 9", ovvero con soli 9 gradi di libertà. Un confronto diretto è possibile effettuarlo andando ad osservare i valori ottenuti per  $N = 3$  con quelli del modello HOST 12. Per completezza vengono riportati tutti i modelli.

Il primo caso analizzato è quello di una piastra sandwich, aventi le facce composte da due strati, antisimmetrica con laminazione  $0^\circ/90^\circ/\text{core}/0^\circ/90^\circ$ . Il materiale delle facce ha le seguenti proprietà:

$E_1 = 19 \cdot 10^6 \text{ lb/in}^2$ ,  $E_2 = 1.5 \cdot 10^6 \text{ lb/in}^2$ ,  $E_2 = E_3$ ,  $G_{12} = G_{23} = 1 \cdot 10^6 \text{ lb/in}^2$ ,  $G_{13} = 0.9 \cdot 10^6 \text{ lb/in}^2$ ,  $\nu_{12} = \nu_{13} = 0.22$ ,  $\nu_{23} = 0.49$ ,  $\rho = 0.057 \text{ lb/in}^3$ .

Il cuore ha invece le seguenti proprietà:

$E_1 = E_2 = E_3 = 1000 \text{ lb/in}^2$ ,  $G_{12} = G_{23} = G_{13} = 500 \text{ lb/in}^2$ ,  $\nu_{12} = \nu_{13} = \nu_{23} = 0$ ,  $\rho = 0.003403 \text{ lb/in}^3$ .

Lo spessore è dato dal seguente rapporto  $t_c/t_f = 10$ , ovvero il cuore ha spessore che è 10 volte maggiore di quello della singola faccia. Il rapporto dei lati  $a/b = 1$  indica una piastra quadrata. Infine, le frequenze sono normalizzate secondo la seguente relazione:

$$\bar{\omega} = \frac{\omega b^2}{h} \sqrt{(\rho/E_2)_f}$$

Nella tabella 6.1 sono riportati sia risultati presenti in [10], ovvero quelli in forma chiusa, che quelli ottenuti dal codice e quindi in forma approssimata. Osservando la tabella 6.1 è possibile notare come per spessori della piastra molto bassi, il codice abbia una soluzione approssimata pressoché identica a quella in forma chiusa riportata. In particolare si deve fare il confronto tra HOST 12 ed  $N = 3$  e tra il modello FOST ed FSDT poiché sono modelli identici. All'aumentare dello spessore, invece, le soluzioni tendono a diventare mano a mano diverse per via del fatto che a

a/h	FOST	HOST9	HOST11	HOST12	CLT	FSDT	N = 2	N = 3	N = 4
2	5.3279	1.1942	1.2138	1.2138	12.1226	5.7782	5.7222	1.1997	1.1968
4	9.2003	2.1356	2.1471	2.1471	15.0398	9.8064	9.6977	2.1385	2.1365
10	13.9936	4.9543	4.9603	4.9603	16.3219	14.4924	14.2650	4.9553	4.9544
20	15.6067	8.7330	8.7372	8.7372	16.5329	15.9818	15.7010	8.7333	8.7330
30	15.9785	11.2296	11.2329	11.2329	16.5728	16.3181	16.0240	11.2299	11.2297
40	16.1157	12.7928	12.7954	12.7954	16.5869	16.4416	16.1425	12.7932	12.7930
50	16.1806	13.7791	13.7812	13.7812	16.5934	16.4998	16.1983	13.7796	13.7795
60	16.1262	14.4212	14.4229	14.4229	16.5970	16.5317	16.2289	14.4219	14.4218
70	16.2378	14.8549	14.8564	14.8564	16.5991	16.5511	16.2475	14.8557	14.8556
80	16.2519	15.1585	15.1597	15.1597	16.6043	16.5675	16.2633	15.1721	15.1720
90	16.2615	15.3777	15.3788	15.3788	16.6015	16.5723	16.2679	15.3786	15.3785
100	16.2685	15.5405	15.5415	15.5415	16.6021	16.5785	16.2738	15.5414	15.5414

**Tabella 6.1:** Frequenze adimensionali per una piastra sandwich non simmetrica ( $0^\circ/90^\circ/\text{core}/0^\circ/90^\circ$ )

spessore crescente i termini di ordine elevato diventano sempre più importanti e, dunque, potrebbe essere utile utilizzare un modello con numero di espansioni maggiore (quindi di gradi di libertà) per tornare ad approssimare perfettamente la soluzione in forma chiusa. È da sottolineare il fatto che l'errore massimo commesso con il modello  $N = 3$  è del 1.2%. Per tanto l'approssimazione data dal codice risulta essere ancora molto buona.

A questo punto si va ad analizzare la BTD per i due casi estremi ( $a/h = 2$  e  $a/h = 100$ ) per vedere come i modelli ridotti rispondono in caso di piastra sandwich. Osservando le figure 6.1 e 6.2 si nota immediatamente come gli errori sono molto più alti rispetto a quelli trovati nella sezione 4.2, indice del fatto che la presenza del cuore, con modulo elastico, modulo a taglio e densità molto più bassi porta ad avere la necessità di lavorare con termini di espansione molto grandi. Confrontando poi le due immagini si nota come la piastra con spessore maggiore ha frequenze molto male approssimate dalla BTD già per modelli con 10 gradi di libertà. Questo fatto è molto in contrapposizione con quanto visto nella sezione 4.2 dove con 10 gradi di libertà si aveva ancora errore molto prossimo allo zero. Il fatto è tanto più evidente quanto più si vanno a togliere gradi di libertà, fino ad arrivare ad un errore del 70% per i modelli a 5 gradi di libertà. Osservando invece la figura 6.2 si può notare che la curva BTD riesce a meglio approssimare le frequenze ottenute con il modello  $N = 4$ , avendo errori molto bassi fino a modelli con 8 gradi di libertà. Anche in questo caso l'errore diventa molto grande a 5 gradi di libertà dove tocca circa il 30%. Questo fatto dimostra ancora una volta come, aumentando lo spessore, aumenta l'importanza dei termini di espansione elevati e diventa difficile adottare modelli ridotti. Tale aspetto lo si può ben vedere confrontando la figura 6.1a, dove vi è un'alta densità di soluzioni con errori molto elevati, con la figura 6.2a dove le soluzioni sono più distribuite, avendo una buona presenza ad errori relativamente bassi.

Inoltre, è possibile andare a fare dei confronti sugli errori percentuali che si ottengono con i modelli HOST con numero di gradi di libertà inferiore a 12 e i modelli ridotti ottenuti sulla curva BTD per lo stesso numero di gradi di libertà. L'errore commesso dai modelli HOST verrà calcolato con la seguente formulazione:

$$\frac{|\bar{\omega}^* - \bar{\omega}|}{\bar{\omega}}$$

dove  $\bar{\omega}^*$  è la pulsazione adimensionalizzata del modello con numero di gradi di libertà inferiore a 12. Nel caso  $a/h = 2$  il modello FOST ha un errore pari al 3.39%, il modello HOST9 dello 1.6% e il modello HOST11 dello 0%. Dalla curva BTD possiamo osservare che il modello a 5 gradi di libertà ha un errore del 70.46%, il modello a 9 gradi di libertà del 50.09% ed, infine, quello con 11 gradi di libertà del 10.56%. Come già detto per la piastra spessa la Best Theory Diagram non riesce a trovare modelli ridotti efficaci.

Nel caso  $a/h = 100$  i modelli in forma chiusa danno i seguenti errori:

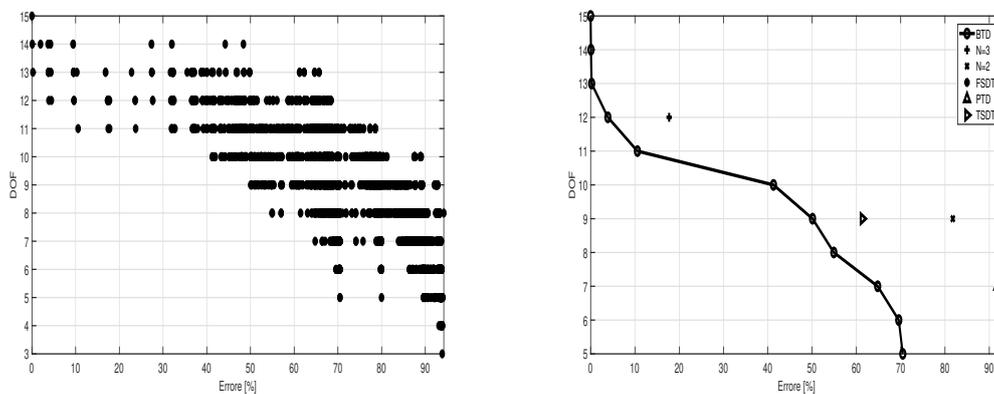
- $FOST = 4.68\%$ ;

- $HOST9 = 0.006\%$ ;
- $HOST11 = 0\%$ .

I modelli ridotti del BTD invece danno i seguenti risultati:

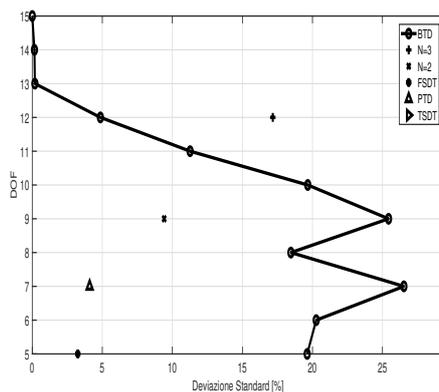
- $N5 = 34.36\%$ ;
- $N9 = 0.09\%$ ;
- $N11 = 0.03\%$ .

In questo caso è chiaramente visibile come gli errori dati dai modelli ridotti della BTD, escluso quello per 5 gradi di libertà, siano molto bassi indice del fatto che i modelli ridotti approssimano bene la soluzione in forma chiusa. È da tenere in conto, inoltre, che le curve BTD analizzate finora utilizzano come modello di riferimento quello con  $N = 4$  che, come visto nella tabella 6.1, presenta una leggera variazione nei valori rispetto ai modelli in forma chiusa.



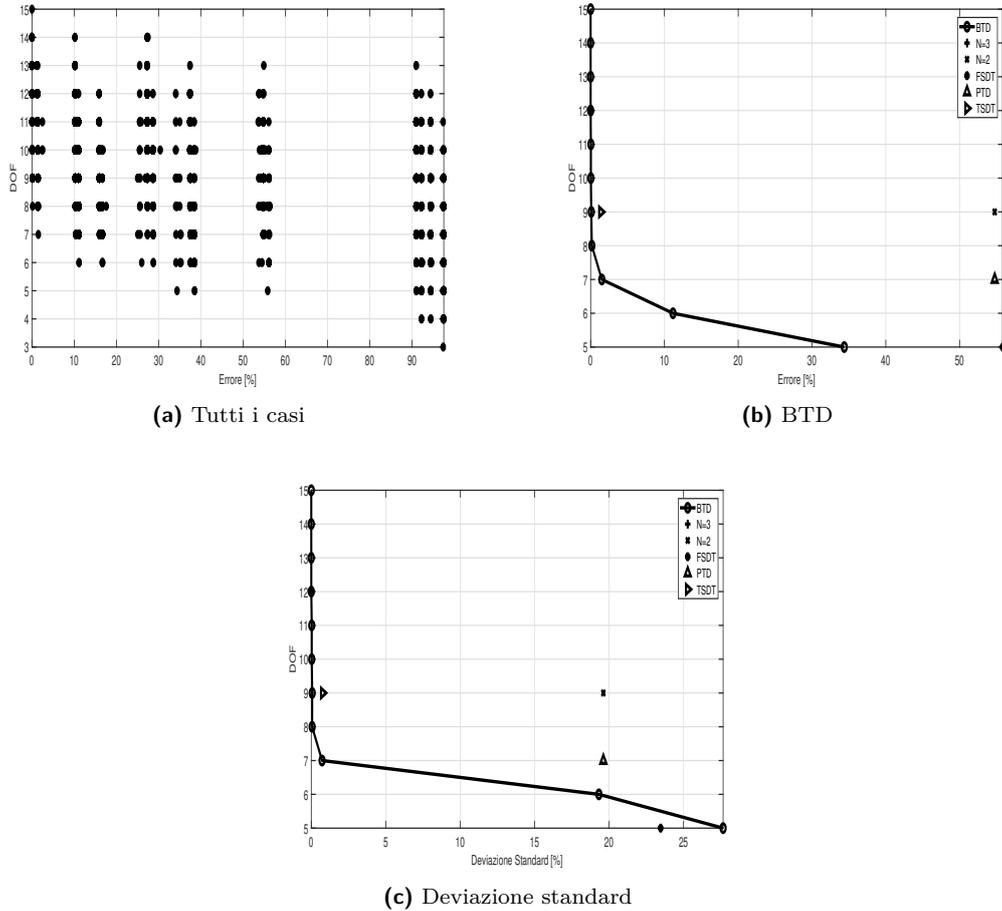
(a) Tutti i casi

(b) BTD



(c) Deviazione standard

Figura 6.1: Grafici per  $a/h = 2$


**Figura 6.2:** Grafici per  $a/h = 100$ 

La seconda analisi riportata in questo capitolo non è più relativa ad una piastra, ma bensì ad un guscio sandwich. Il guscio è di tipo sferico con raggi di curvatura variabili, ma sempre  $R_\alpha = R_\beta$ . I dati relativi ai materiali della facce e del cuore sono gli stessi dell'analisi della piastra. L'adimensionalizzazione della pulsazione viene effettuata tramite la seguente relazione:

$$\bar{\omega} = \frac{\omega b^2 \sqrt{(\rho/E_2)_f}}{h}$$

I risultati sono riportati nella tabella 6.2 e, anche in questo caso, riportano le frequenze ottenute con i modelli in forma chiusa e quelle ottenute con il "MUL2 code". Anche in questo caso i risultati sono molto precisi e vanno oltre la seconda cifra decimale di accuratezza per il guscio più sottile. Quando lo spessore aumenta i risultati hanno precisione che va a calare mantenendo comunque un'accuratezza molto elevata (errori dell'ordine dello 0.4 – 0.5%).

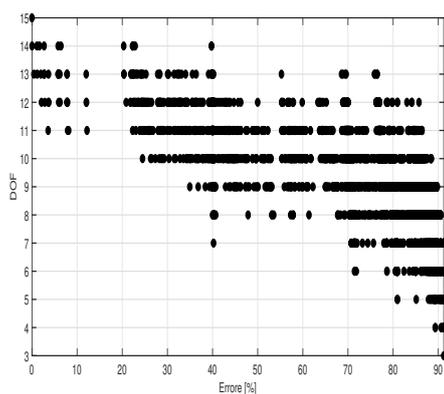
Come fatto nel caso precedente si va ad analizzare la BTD per i due casi estremi ed ogni spessore. Il primo caso analizzato è quello per  $R/a = 1$  i grafici sono quelli nelle figure 6.3, 6.4 e 6.5. Analizzando qualitativamente gli andamenti delle curve BTD si nota subito quanto detto nell'analisi precedente, ovvero che andando a diminuire lo spessore diminuisce anche l'errore percentuale commesso per ogni modello ridotto. Si può anche notare come il grafico della deviazione standard

presenti valori molto elevati anche per modelli ridotti con alto numero di gradi di libertà nel caso riportato in figura 6.3c, mentre assume valori molto più accettabili nel caso di figura 6.4c. Nei primi due casi non vi è sostanziale differenza tra le distribuzioni di tutte le soluzioni in quanto entrambe presentano alta densità ad elevati valori di errore. Ciò che si nota nel caso in figura 6.4a è una presenza di pallini dai 15 ai 7 gradi di libertà in corrispondenza di errore relativamente basso coerente con quanto riportato nel grafico 6.4b. Il caso con  $h/a = 0.01$  conferma quanto dedotto fino a questo punto presentando una distribuzione di tutte le soluzioni con densità abbastanza alta verso errori relativamente piccoli e, infatti, l'errore massimo ottenuto dalla corrispondente curva BTD riportata in figura 6.5b è inferiore al 5%. Dal caso di guscio con  $h/a = 0.25$  al caso con  $h/a = 0.01$  vi è stata una riduzione dell'errore massimo pari circa al 94%.

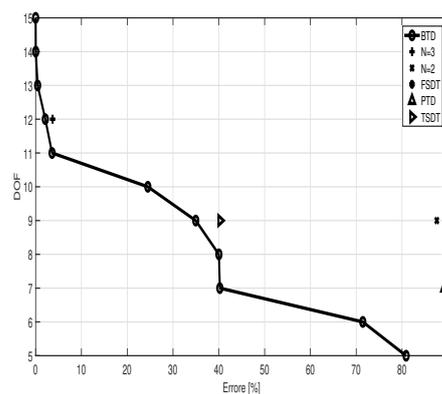
A questo punto si passa a studiare il guscio con  $R/a = 20$  i cui risultati sono riportati nelle figure RIFERIMENTO, 6.7 e 6.8. Le considerazioni generali sono le stesse fatte per il caso precedente, con una sostanziale differenza. Si nota che con il raggio di curvatura molto più elevato anche gli errori sono cresciuti. Questo lo si può facilmente notare confrontando la figura 6.8b con la figura 6.5b. Si osserva infatti che a parità di spessore il caso con raggio di curvatura maggiore presenta errori per modelli con bassi gradi di libertà decisamente più elevati. Tale comportamento lo si poteva intuire andando ad analizzare la figura 6.2b che riporta l'andamento BTD di una piastra. Come già visto, l'errore massimo si attesta intorno al 35%, ovvero un valore ancora maggiore al caso con  $R/a = 20$ . Essendo la piastra un guscio con raggio di curvatura infinito, si può concludere che la precisione della BTD diminuisce con l'aumentare del raggio di curvatura del guscio.

R/a	h/a	HOST 12	HOST 11	HOST 9	FOST	N = 4	N = 3	N = 2	FSDT	CLT
1	0.01	123.56674	123.56958	123.5846	123.65504	123.56343	123.56371	123.63144	123.66615	127.30283
	0.1	12.93889	12.97389	12.98830	17.05499	12.93063	12.93150	17.04142	17.18725	18.83401
	0.25	5.22250	5.23312	5.23586	9.16973	5.21681	5.21878	9.28547	9.35689	12.85923
2	0.01	66.33009	66.33180	66.34089	66.50321	66.32833	66.32839	66.48999	66.56259	68.69567
	0.1	8.00227	8.02226	8.02745	14.89910	7.99558	7.99575	15.07017	15.27144	17.05303
	0.25	3.29400	3.30245	3.29904	9.18443	3.28727	3.28766	9.56196	9.66024	14.40329
3	0.01	46.11017	46.11132	46.11727	46.35985	46.10860	46.10864	46.35162	46.45781	47.87903
	0.1	6.51682	6.52845	6.52902	14.41078	6.51057	6.51096	14.63424	14.84937	16.65706
	0.25	2.72672	2.73190	2.72516	9.19246	2.71844	2.71942	9.63423	9.73830	14.74908
4	0.01	36.17988	36.18072	36.18489	36.49785	36.17902	36.17906	36.49317	36.62888	37.66327
	0.1	5.89249	5.89989	5.89810	14.23136	5.88644	5.88698	14.47508	14.69543	16.51253
	0.25	2.49188	2.49522	2.48674	9.19573	2.48276	2.48411	9.66136	9.76751	14.87537
5	0.01	30.44502	30.44566	30.44870	30.82233	30.44435	30.44438	30.81976	30.98083	31.77421
	0.1	5.57671	5.58176	5.57866	14.14671	5.57073	5.57137	14.40018	14.62302	16.44453
	0.25	2.37420	2.37649	2.36703	9.19732	2.36459	2.36616	9.67425	9.78139	14.93480
10	0.01	20.33751	20.33775	20.33837	20.89829	20.33721	20.33725	20.90043	21.13823	21.44686
	0.1	5.12201	5.12341	5.11823	14.03223	5.11612	5.11691	14.29906	14.52530	16.35276
	0.25	2.20634	2.20697	2.19597	9.19953	2.19595	2.19787	9.69178	9.80023	15.01502
20	0.01	16.87017	16.87025	16.86979	17.54220	16.87002	16.87007	17.54653	17.82941	17.93790
	0.1	5.00124	5.00160	4.99583	14.00331	4.99537	4.99621	14.27356	14.50065	16.32961
	0.25	2.16211	2.16228	2.15084	9.20010	2.15150	2.15352	9.69622	9.80500	15.03525
Plate	0.01	15.54147	15.54147	15.54053	16.26848	15.54138	15.54143	16.27380	16.57852	16.60214
	0.1	4.96028	4.96028	4.95430	13.99365	4.95441	4.95527	14.26504	14.49242	16.32188
	0.25	2.14715	2.14715	2.13556	9.20029	2.13646	2.13851	9.69771	9.80636	15.03984

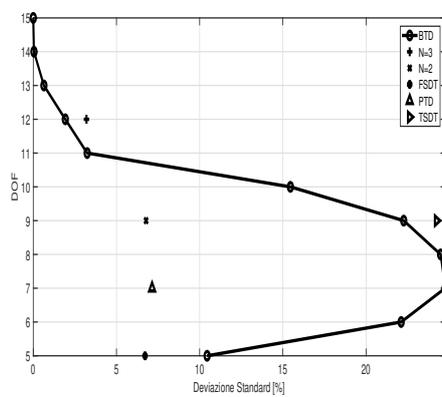
**Tabella 6.2:** Frequenze adimensionali per un guscio sandwich non simmetrico ( $0^\circ/90^\circ/\text{core}/0^\circ/90^\circ$ )



(a) Tutti i casi



(b) BT



(c) Deviazione standard

Figura 6.3: Grafici per  $h/a = 0.25$  e  $R/a = 1$

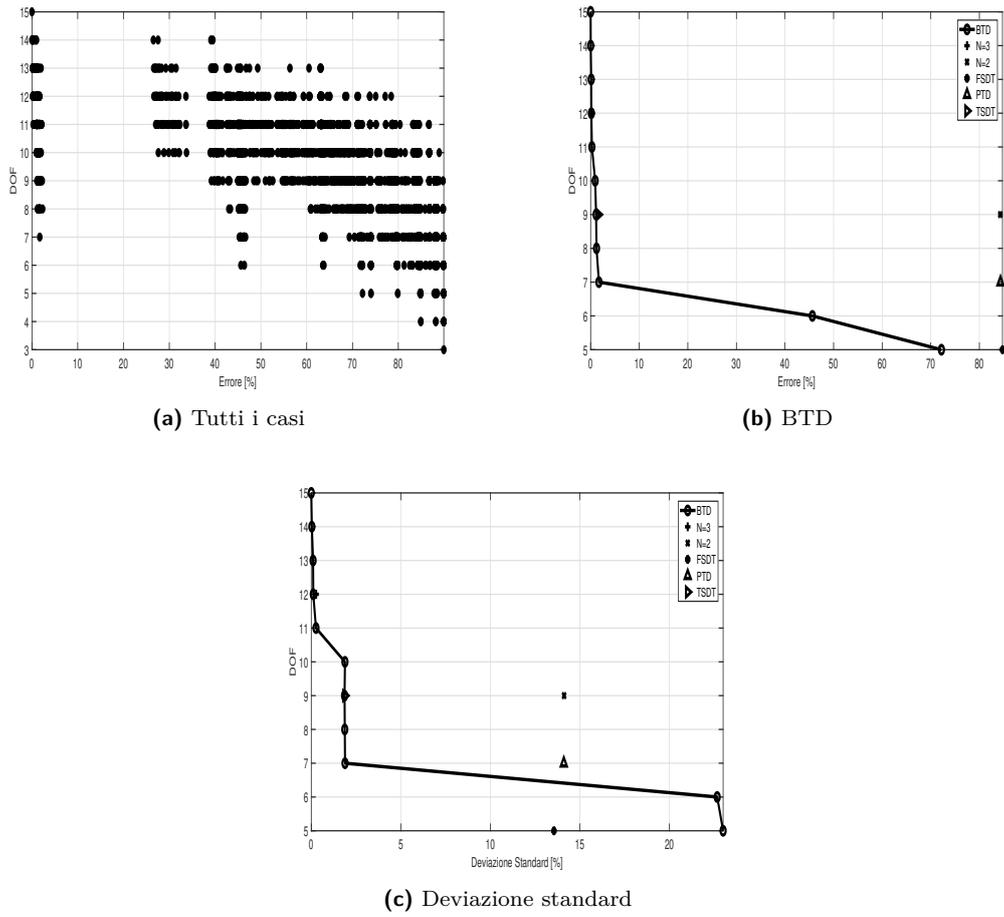


Figura 6.4: Grafici per  $h/a = 0.1$  e  $R/a = 1$

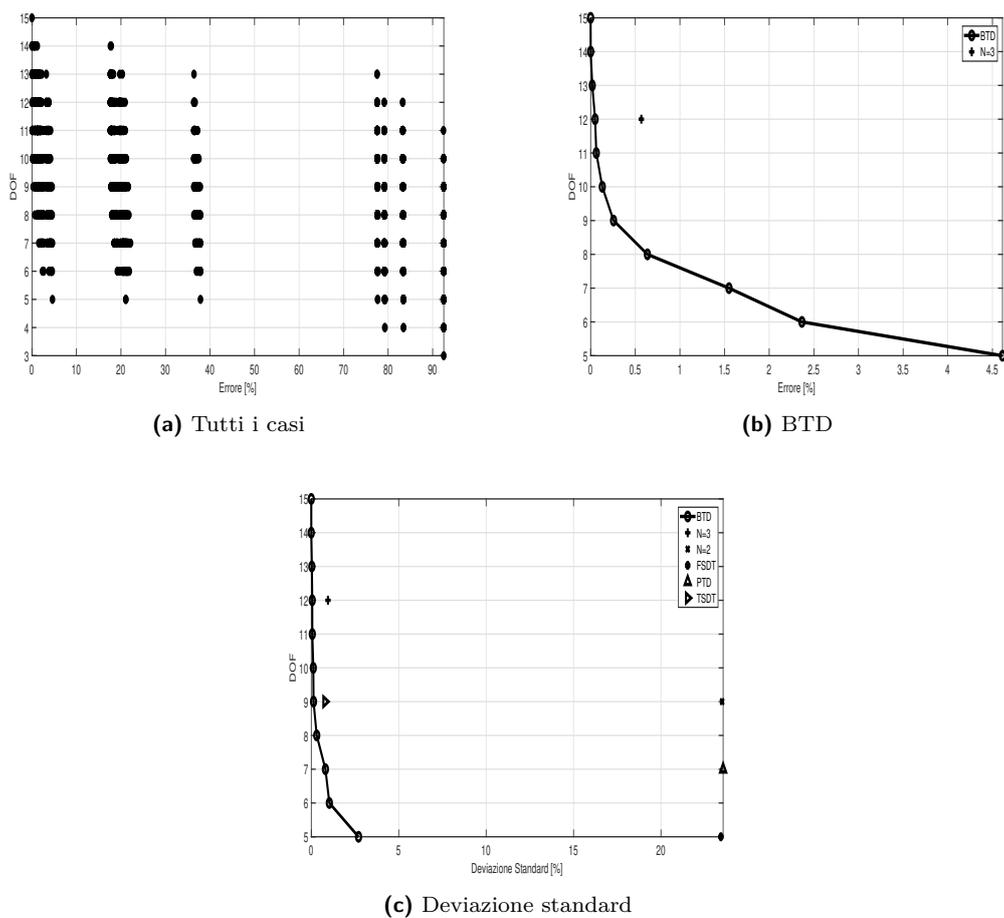


Figura 6.5: Grafici per  $h/a = 0.01$  e  $R/a = 1$

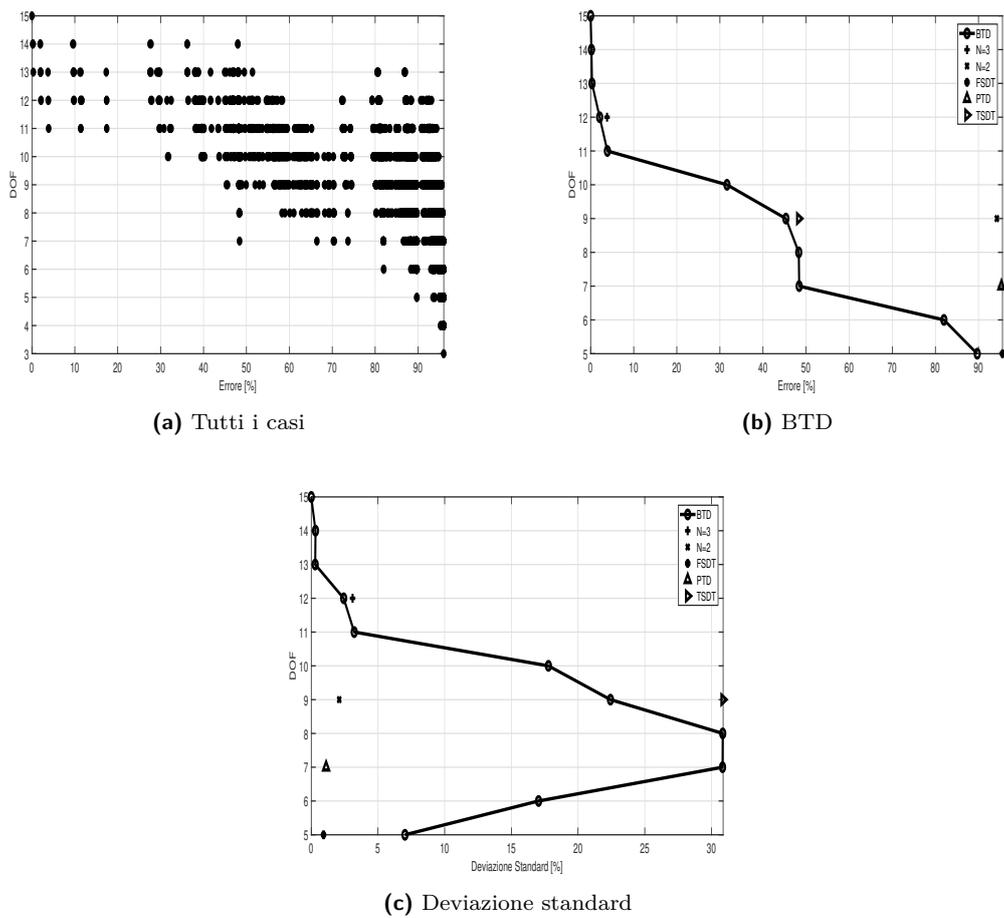


Figura 6.6: Grafici per  $h/a = 0.25$  e  $R/a = 20$

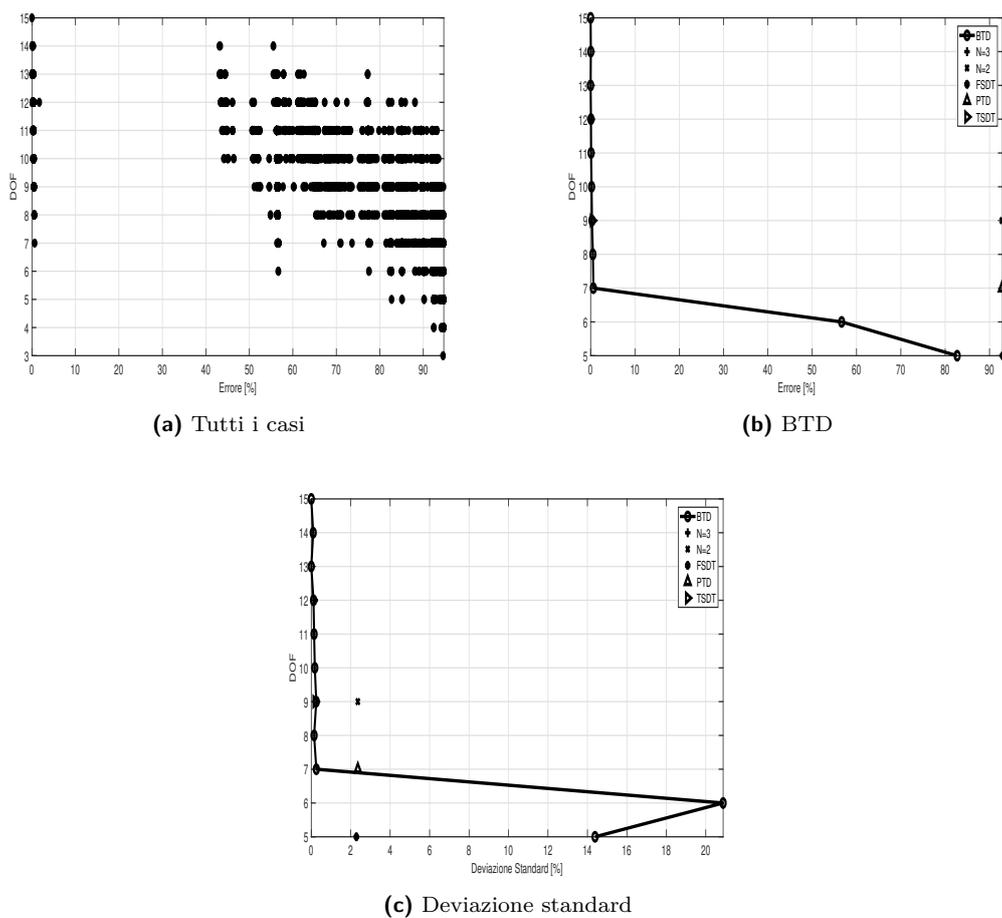


Figura 6.7: Grafici per  $h/a = 0.1$  e  $R/a = 20$

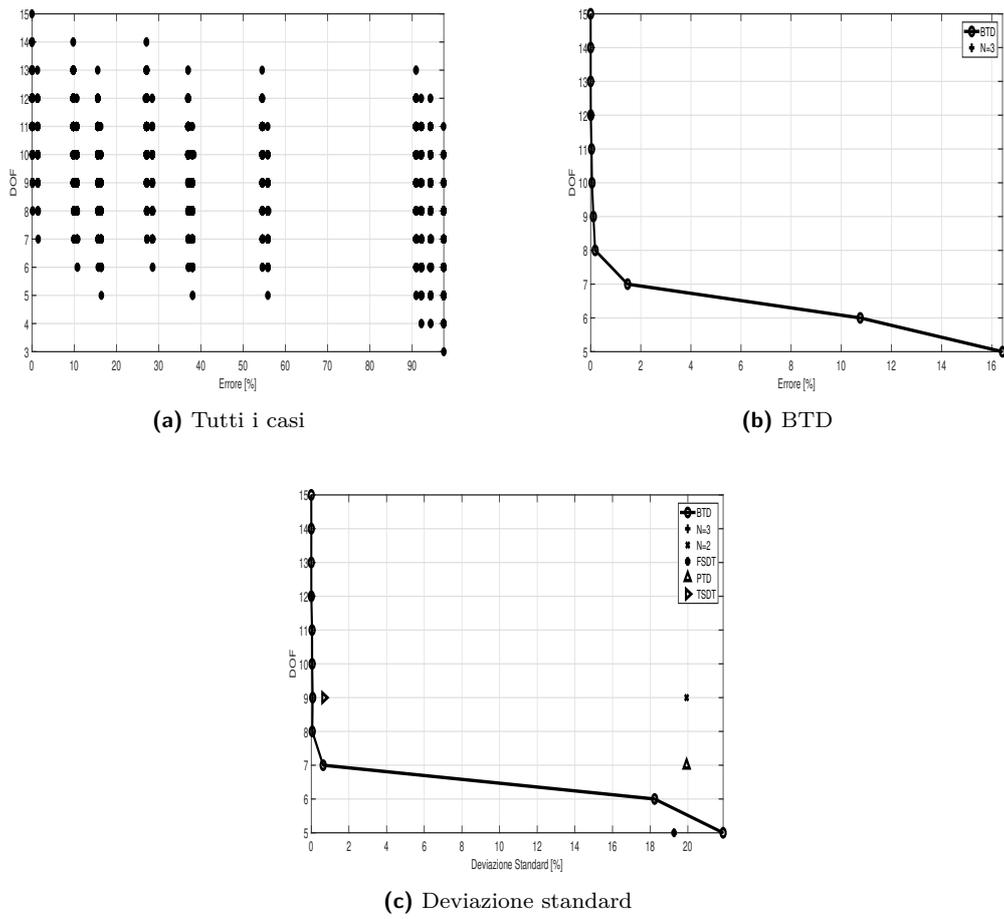


Figura 6.8: Grafici per  $h/a = 0.01$  e  $R/a = 20$



**Parte II**  
**Risultati**



# Capitolo 7

## Analisi rete neurale

### 7.1 Variazione del Modello di riferimento

Nella sezione 2.3.1 si è parlato di come viene impostata la rete neurale per la determinazione di una curva Best Theory Diagram. I primi risultati che si vogliono ottenere sono quelli relativi ad un cambio di modello di riferimento utilizzato per il calcolo dell'errore dei primi dieci modi di vibrare. In questo modo si va a testare il comportamento della rete neurale ad una prima modifica dei dati utilizzati. Come avevamo si era visto, l'errore viene calcolato utilizzando la seguente relazione:

$$\epsilon = \frac{f_i^{N=4} - f_i^*}{f_i^{N=4}}$$

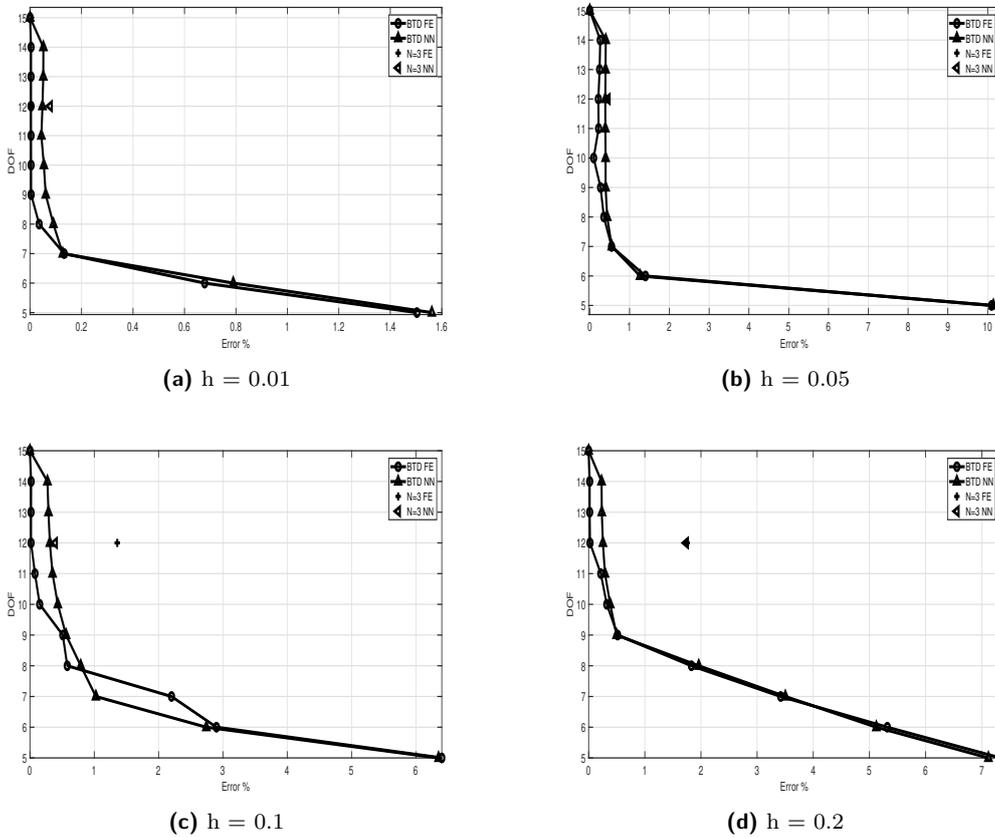
In questa fase, però, non si userà più il modello  $N = 4$  come riferimento, ma si andrà ad inserire il modello LD3. L'errore risulterà essere quindi calcolato in maniera analoga nel seguente modo:

$$\epsilon = \frac{f_i^{LD=3} - f_i^*}{f_i^{LD=3}}$$

Avendo variato il modello di riferimento, passando da un Equivalent Single Layer al quarto ordine di espansione ad un Layer Wise al terzo ordine di espansione, varieranno anche le frequenze ottenute tra un caso e l'altro. In tabella 7.1 sono riportati i modi di vibrare normalizzati dei gusci in esame così da evidenziare le differenze tra il caso ED4 e LD3 (entrambe le teorie sono state introdotte in 1.2). Osservando i valori riportati in tabella si nota che essi differiscono tra ED4 e LD3 di poco, ma sempre di più a mano a mano che si aumenta lo spessore del guscio. Per effettuare l'analisi si utilizza la stessa rete introdotta nei paragrafi precedenti ed utilizzando gli stessi parametri trovati iterativamente. In questo modo si andrà anche a testare la ripetibilità della rete anche quando si variano alcuni aspetti dei dati da analizzare. Come già detto, per la rete, non esistono degli algoritmi standard che ci dicano come impostare i parametri del training e quindi al variare dei dati in input è molto probabile che varino anche le esigenze della rete. Per questo caso non ci si attende sostanziale variazione del training poiché si è variato il vettore di target solo dal punto di vista numerico e non della forma. In figura 7.1 sono riportati tutte le curve Best theory diagram ricavate per i gusci analizzati in tabella 7.1. Quello che si nota è che la rete neurale è riuscita ad approssimare molto bene quella che è la curva ottenuta dall'analisi FEM. Tale comportamento lo si poteva attendere in quanto, come già anticipato precedentemente, non si è variata sostanzialmente la rete neurale dai casi precedenti a questo, ma soltanto è variato il modo di fornire i dati in input.

Modo	h/a = 0.01		h/a = 0.05		h/a = 0.1		h/a = 0.2	
	ED 4	LD 3	ED 4	LD3	ED 4	LD3	ED 4	LD3
1	3.10E-02	3.10E-02	3.74E-01	3.70E-01	1.65E+00	1.60E+00	3.31E+00	3.21E+00
2	5.80E-02	5.80E-02	9.50E-01	9.45E-01	4.25E+00	4.19E+00	8.51E+00	8.38E+00
3	1.17E-01	1.17E-01	1.97E+00	1.90E+00	5.42E+00	5.30E+00	1.08E+01	1.06E+01
4	1.29E-01	1.28E-01	2.19E+00	2.13E+00	6.72E+00	6.60E+00	1.34E+01	1.32E+01
5	1.36E-01	1.35E-01	2.20E+00	2.17E+00	7.69E+00	7.59E+00	1.54E+01	1.52E+01
6	1.73E-01	1.73E-01	3.00E+00	2.93E+00	8.15E+00	8.02E+00	1.63E+01	1.60E+01
7	2.19E-01	2.19E-01	3.86E+00	3.72E+00	9.30E+00	9.15E+00	1.86E+01	1.83E+01
8	2.60E-01	2.59E-01	3.88E+00	3.84E+00	9.55E+00	9.33E+00	1.91E+01	1.87E+01
9	3.33E-01	3.29E-01	4.00E+00	3.87E+00	1.03E+01	9.82E+00	2.06E+01	1.96E+01
10	3.38E-01	3.34E-01	4.51E+00	4.34E+00	1.04E+01	1.02E+01	2.07E+01	2.03E+01

**Tabella 7.1:** Primi 10 modi di vibrare per diversi gusci venti  $R/a = 0.2$  e normalizzazione del tipo:  $\omega \cdot 10\sqrt{\rho h^2/E_T}$



**Figura 7.1:** Curve BTD ricavate utilizzando il modello LD3 come riferimento per la valutazione dell'errore

## 7.2 Influenza singola frequenza

In questa sezione, come fatto precedentemente (paragrafo 5), si va ad analizzare come la singola frequenza (il singolo modo di vibrare) contribuisce all'errore sulla curva globale. Sono di seguito

riportati quindi i grafici relativi ad ogni modo di vibrare confrontati con quelli ottenuti per lo stesso modo di vibrare dalla FEM. Questa analisi viene fatta utilizzando scegliendo le coppie di dati input/output a passi regolari e non random così da avere sempre lo stesso risultato e poterlo confrontare meglio. I risultati sono riportati nelle figure 7.2 e 7.3. In generale si può osservare che il modello strutturale a 5 gradi di libertà è il più difficile da cogliere rispetto agli altri ed infatti il risultato di tutti i modi di vibrare comporta errori maggiori rispetto agli altri. infatti, esclusi i modi 4,5,9 e 10, gli altri hanno un errore percentuale molto superiore a quello previsto dalla FEM, nonostante gli altri modelli a più alto numero di gradi di libertà vengano colti in modo accurato. Sempre in generale fino tutti i modelli strutturali fino a 7 gradi di libertà hanno errori molto prossimi allo zero e questo si rispecchia nell'andamento globale in cui le curve derivanti dalla FEM e dalla rete neurale sono praticamente sovrapponibili. Come succedeva in 5 gli errori commessi sul singolo modo di vibrare sono inferiori rispetto all'errore globale, questo perché nell'andamento complessivo il modello strutturale deve accomodare tutti e 10 i modi di vibrare andando così a non prendere il migliore di ognuno.

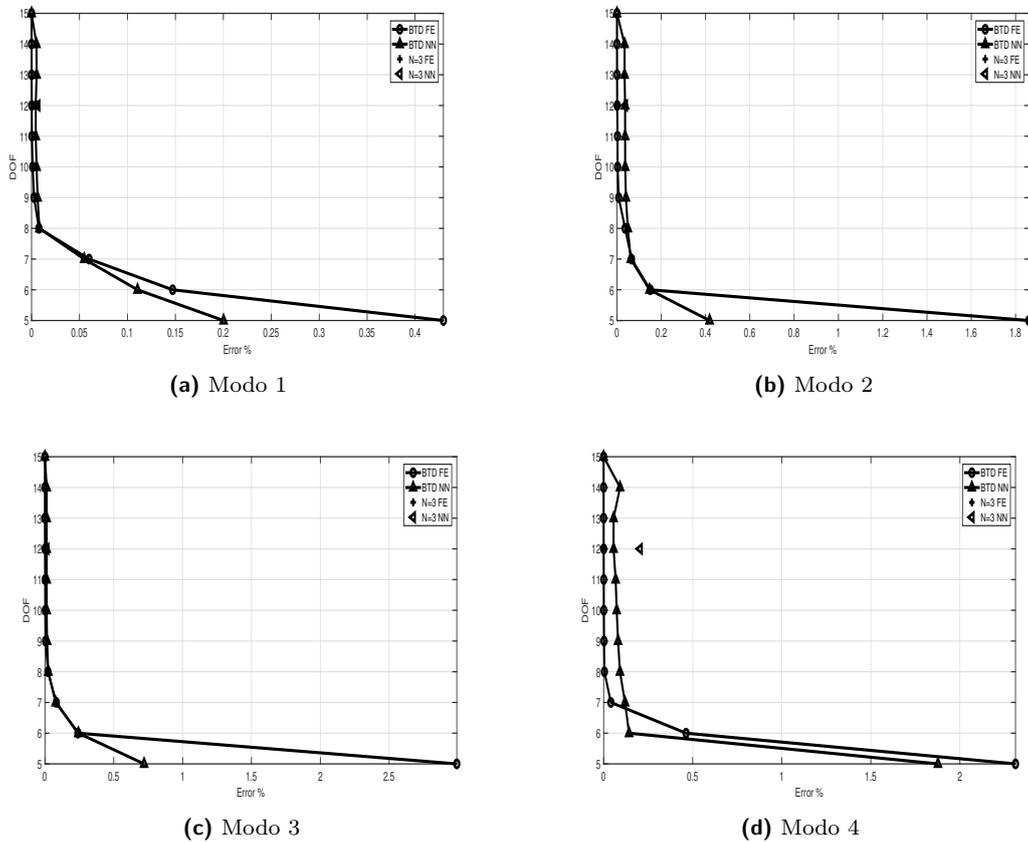
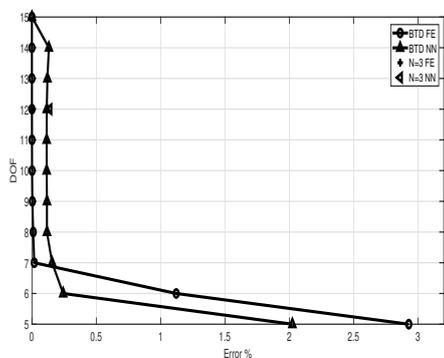
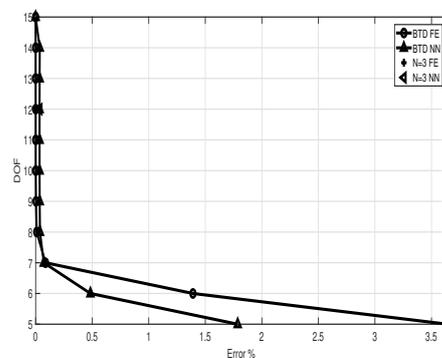


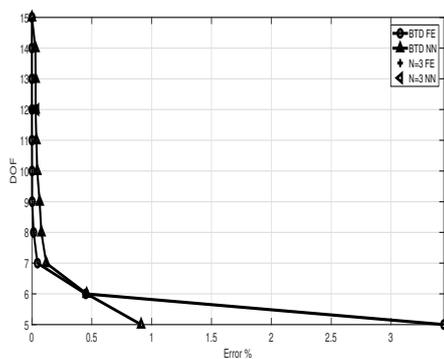
Figura 7.2: Confronto curve per i modi di vibrare dal 1 al 4



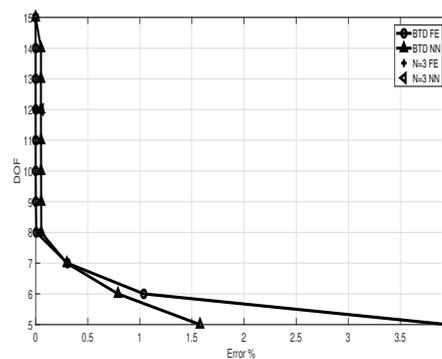
(a) Modo 5



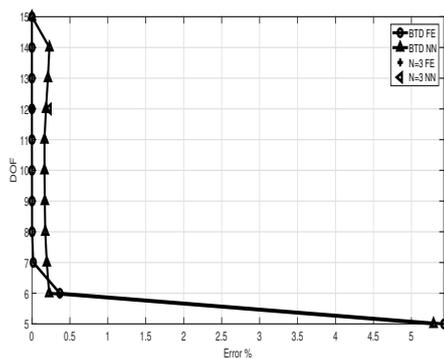
(b) Modo 6



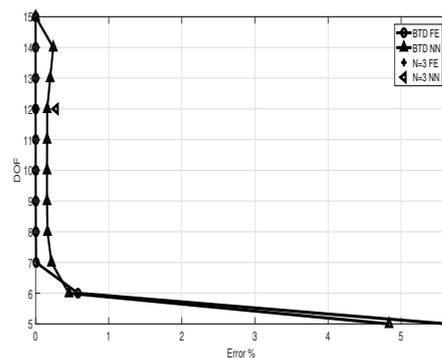
(c) Modo 7



(d) Modo 8



(e) Modo 9



(f) Modo 10

Figura 7.3: Confronto curve per i modi di vibrare dal 5 al 10

## Capitolo 8

# Influenza dello spessore

Durante l'introduzione alla Best Theory Diagram in 2.2 si è parlato dei fattori che vanno ad influenzare la curva. Tra questi vi è lo spessore della piastra. I risultati finora ottenuti sono stati ricavati andando a considerare uno spessore di piastra alla volta e ricercando quindi soluzioni per quello spessore. Il vantaggio di questa applicazione, come già detto, risiede nel poter andare a ridurre il numero di analisi agli elementi finiti per poter ottenere un risultato, ma senza però avere la possibilità di variare nessun parametro geometrico. A tal proposito si è andati a migliorare la rete in modo da poter aumentare ancora il vantaggio che questa apporta e ridurre ancora di più il tempo per ottenere una soluzione. L'idea è stata infatti quella di allenare la rete fornendo come input, oltre al modello strutturale, anche lo spessore della piastra relativo a quel modello. Il target che si otterrà sarà il risultato di 12 input dati dal modello strutturale ed un input dato dallo spessore della piastra. In questo modo è possibile, andando a fornire la rete in fase di allenamento di un numero finito di spessori di piastra, ricavare qualsiasi curva BTD per un generico spessore. Per questa applicazione si giustifica la scelta di utilizzare più strati interni, in quanto il problema è più complicato ed un solo strato non permette la sua risoluzione. Per poter utilizzare questa applicazione si è modificato il codice permettendo di inserire in input un certo numero di risultati FEM con il loro relativo spessore per poter allenare la rete e chiedere in output al problema un qualsiasi spessore di piastra compreso tra questi. Per fare questo si è modificato il processo di raccolta dati andando a creare una matrice di dimensione pari al numero di casi dati in input moltiplicato per il numero di incognite del problema (in questo caso  $2^{12}$ ) come righe e 13 colonne, ovvero 12 input del modello strutturale a cui si aggiunge l'input dello spessore. La problematica che si è riscontrata è stata il fatto che la scelta di coppie input/target in modo random portava in alcuni casi a risultati piuttosto diversi da quelli attesi poiché la scelta casuale poteva ricadere in modo più preponderante su uno spessore e meno in un altro. Se in quel caso, come output, si chiedeva la BTD per uno spessore più vicino a quello scelto in misura minore durante la fase di training la curva veniva colta molto male. Per ovviare a questo inconveniente si è impostato che le coppie input/target venissero scelte sempre in modo casuale, ma nello stesso numero per tutti gli input di training impostati.

In figura 8.1 sono riportate le curve BTD per due diversi spessori ottenuti grazie alla rete neurale allenata su piastre che presentavano rapporti  $h/a$  diversi da quelli richiesti. Infatti, nel caso specifico in figura, la rete è stata allenata avendo come input i seguenti valori di  $h/a$ : 0.01; 0.05; 0.1; 0.2. Analizzando la figura 8.1 si osserva che la curva prevista segue piuttosto fedelmente quella ricavata attraverso l'analisi agli elementi finiti. In 8.1a si nota che solo il modello strutturale a 5 gradi di libertà non viene colto in maniera precisa, mentre tutti gli altri modelli hanno errori molto ridotti. Per quanto riguarda la curva rappresentata in figura 8.1b invece tutti i punti sono stati previsti con un livello molto alto di accuratezza. Anche in questo caso, come già riscontrato nella sezione precedente, la scelta di coppie input/target in maniera casuale comporta una diversa

approssimazione della curva in diverse analisi. Anche in questo caso, però, la scelta di questa metodologia rispetto a quella a passi costanti è dettata del fatto che fornisce risultati mediamente sempre migliori.

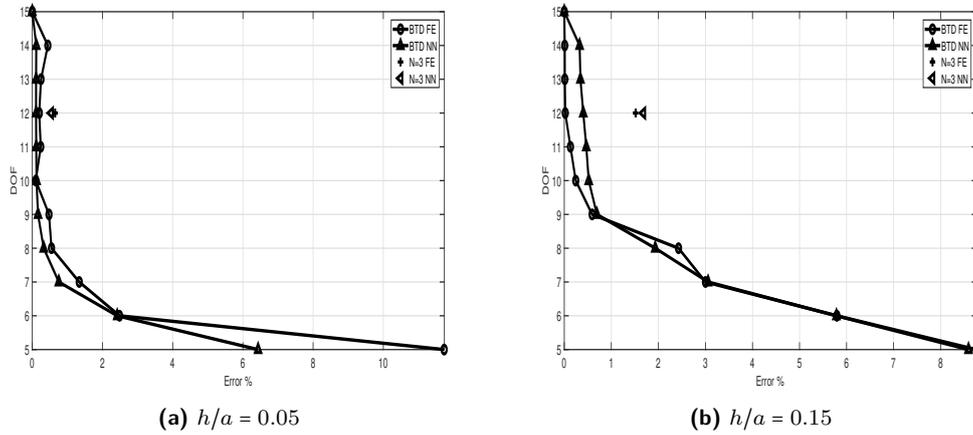


Figura 8.1: Previsioni BTD per diversi spessori

Questa prima analisi è stata fatta andando ad utilizzare uno spessore compreso nell'intervallo di quelli utilizzati per il training della rete neurale. La seconda analisi che viene eseguita è utilizzando uno spessore che sia di valore superiore a quello massimo dell'intervallo di training per verificare se è sufficiente per la rete avere un numero di spessori di guscio su cui allenarsi per poter poi prevedere tutti gli altri. Per testare la rete in questo senso si procede quindi andando a richiedere come output la Best Theory Diagram per un guscio di spessore  $h/a = 0.4$ , dando come input gli stessi rapporti riportati in precedenza. In questo modo si sta richiedendo un risultato per un guscio con spessore doppio rispetto al massimo spessore utilizzato per il training. Il risultato di tale analisi è riportato in figura 8.2.

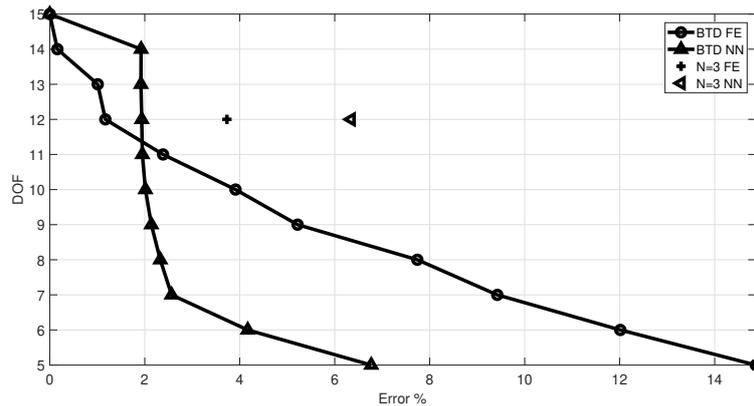


Figura 8.2: Curva BTD prevista per  $h/a = 0.4$  utilizzando un intervallo di training che non comprende tale spessore

Si osserva che la curva BTD ricavata con la rete neurale è del tutto diversa da quella ricavata con l'analisi agli elementi finiti. Inoltre, in fase di ricerca della soluzione, il software ha impiegato

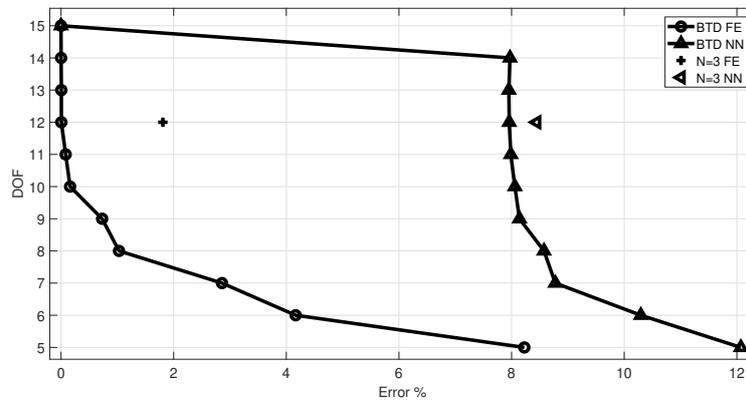
più tempo rispetto ai casi precedenti per estrapolare la curva indice del fatto che ha faticato maggiormente a trovare una soluzione. Per questi motivi si deve escludere la possibilità, in questa fase, di poter prevedere un andamento per uno spessore non compreso tra quelli imposti in fase di training. Per poter prevedere il rapporto  $h/a = 0.4$  è necessario aumentare l'intervallo di training fino ad almeno quel valore.

## 8.1 Aumento dei dati in input

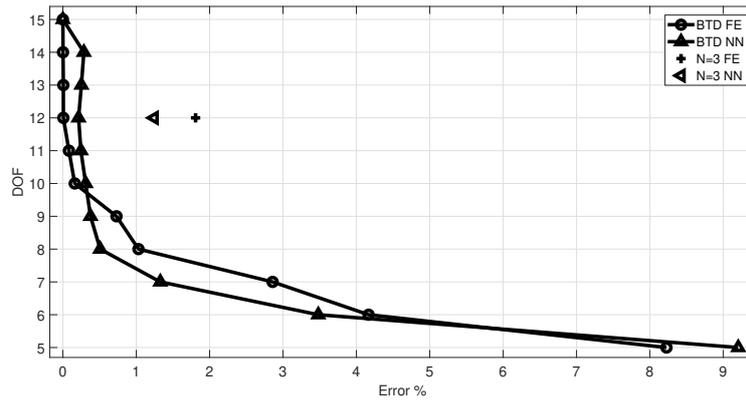
Nella sezione precedente si è osservata la potenzialità della rete neurale in fase di previsione di una curva BTD aumentando il numero di input con l'inclusione dell'influenza dello spessore. Quello che si vuole fare a questo punto è, allora, aumentare ancora i dati che si forniscono alla rete in fase di training in modo da aumentare ancora il campo di applicazione della rete allenata. Durante la simulazione FEM completa si vanno ad inserire, oltre allo spessore della piastra, anche la laminazione, il tipo di materiale e le condizioni al contorno. In questa fase si cercherà allora di andare a fare una previsione di curva BTD per generiche condizioni al contorno e generica laminazione fornendo in input un set di dati con le proprie condizioni al contorno e laminazioni. Quello che si vuole ottenere è, similmente a quanto fatto per lo spessore, costruire la curva BTD per infinite condizioni di input partendo da un set limitato di simulazioni FEM date alla rete neurale in fase di training. Ciò che si è osservato nel corso del lavoro è che la rete neurale risponde bene ai requisiti di errore massimo accettabile se in input vengono forniti valori compresi tra 0 ed 1; allo stesso modo, se in output la rete deve fornire valore compresi tra 0 e l'unità, i risultati sono migliori. A tal proposito l'influenza dello spessore era già univocamente determinato nell'intervallo di valori richiesto, nel caso in esame invece bisogna prestare più attenzione. La laminazione generalmente è un'alternanza di fibre orientate, rispetto ad un riferimento, di  $0^\circ, 90^\circ/45^\circ$ . Naturalmente possono essere presenti molte altre orientazioni, ma le più comuni sono quelle riportate. Le condizioni al contorno presentano un problema anche più complicato in quanto non vi sono dei valori che esprimano espressamente la condizione al contorno utilizzata. Risulta quindi essere necessaria una convenzione con cui riportare univocamente gli input richiesti. Siccome tali input non vengono direttamente utilizzati dalla rete per svolgere dei calcoli, ma come riferimento per l'associazione dell'input con il target, il valore da assegnare può essere arbitrario con l'unico vincolo che sia univoco.

Per identificare l'orientamento delle fibre si è scelto di dare il valore 1 se esse sono orientate a  $90^\circ$ , 0 se orientate a  $0^\circ$  e 5 se orientate a  $45^\circ$ . Questi valori devono essere riportati dopo uno 0. Ad esempio, se una piastra con tre strati ha orientamento delle fibre del tipo  $0^\circ/45^\circ/90^\circ$  allora l'input da inserire per la rete sarà 0.051. Con lo stesso principio si vanno ad identificare le condizioni al contorno. In questo caso abbiamo la possibilità di avere lato incastrato o lato appoggiato. Si assegna al lato incastrato il valore 1 e 0 a quello appoggiato. In questo modo, se un guscio avrà due lati appoggiati e due incastrati alternati il valore da fornire alla rete in input sarà 0.1010. Usando quindi queste convenzioni, insieme naturalmente a quella per lo spessore, si è andati a svolgere delle simulazioni per testare come risponde la rete ad un numero di input più grande. La rete, per le simulazioni successive, viene allenata dando quattro analisi complete provenienti dalla FEM relativi ai seguenti valori di  $h/a$  : 0.01, 0.05, 0.15 e 0.2. Alla seconda ed alla terza piastra verrà fornita una laminazione di  $0^\circ/90^\circ/0^\circ$ , mentre alle altre due  $90^\circ/0^\circ/90^\circ$ . Per il primo test le condizioni al contorno riporteranno per tutte le piastre il valore in input relativo alla condizione di quattro lati semplicemente appoggiati. In output verrà richiesta la BTD per una piastra di spessore  $h/a = 0.1$  con laminazione  $90^\circ/0^\circ/90^\circ$ . Utilizzando la rete descritta nelle sezioni precedenti, il risultato ottenuto dal software non era in grado di soddisfare i requisiti di errore massimo ammesso tra la curva BTD ottenuta con l'analisi FEM e quella ottenuta, appunto, attraverso la rete neurale. Siccome, al momento, non esistono algoritmi standard per il training di una rete, quello che si fa è andare per tentativi fino ad arrivare ad una convergenza di soluzione accettabile.

Il primo tentativo che si è portato avanti è stato quello di aumentare il numero di strati interni della rete, ma tale soluzione si è dimostrata inefficiente. Ciò che invece ha portato un ottimo risultato è stato aumentare radicalmente il numero di neuroni per ogni strato interno. Questo accorgimento, naturalmente, viene pagato in costo computazionale (conseguentemente tempo) utile all'ottenimento della curva. Questo tempo risulta però essere indispensabile in quanto non esiste una soluzione più rapida per il tipo di problema in esame. In figura 8.3 viene mostrata la curva Best theory Diagram ottenuta utilizzando la precedente rete neurale. È evidente come le curve riportate abbiano vagamente in comune solamente l'andamento globale, ma che i valori ottenuti con la rete neurale siano completamente diversi da quelli di riferimento. In figura 8.4, invece, viene riportata la curva BTD ricavata dalla rete neurale con un numero maggiore di neuroni (sempre confrontata con quella ottenuta attraverso l'analisi FEM). In questo caso si può notare un andamento molto prossimo tra le due curve, con valori che non si discostano di molto.



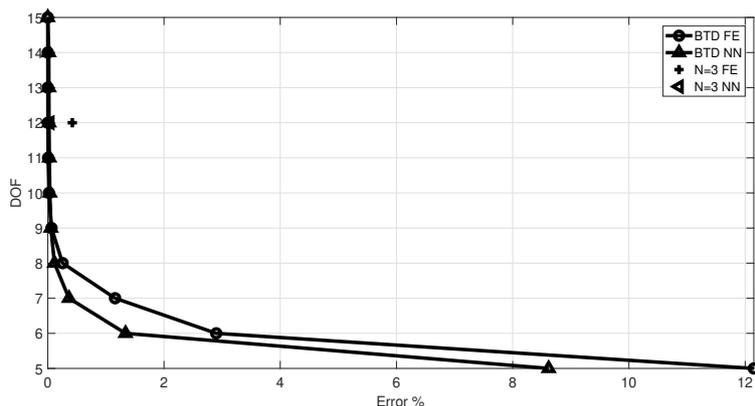
**Figura 8.3:** Curva BTD prevista per  $h/a = 0.1$ , con in input i valori di laminazione, condizioni al contorno e spessore. Tre strati interni con 15/10/10 neuroni



**Figura 8.4:** Curva BTD prevista per  $h/a = 0.1$ , con in input i valori di laminazione, condizioni al contorno e spessore. Tre strati interni con 15/25/25 neuroni

Successivamente, viene testato un altro guscio per verificare la solidità della rete. Gli input per il training rimangono gli stessi, mentre in output viene chiesta una curva BTD per un guscio

con laminazione  $90^\circ/0^\circ/90^\circ$  e spessore con rapporto  $h/a = 0.05$ . I risultati vengono riportati in figura 8.5 e si può notare che anche in questo caso le due curve ottenute con i due metodi sono molto vicine tra loro e presentano scarti ai vari gradi di libertà contenuti, eccetto per il modello strutturale a cinque gradi di libertà che contiene un errore più alto rispetto agli altri. In questo caso, avere moltissime combinazioni da prevedere e molti input da gestire porta la rete a non essere precisa come negli altri casi. È da sottolineare però che viene commesso un errore di poco sopra l'1% a fronte di un errore del 12%



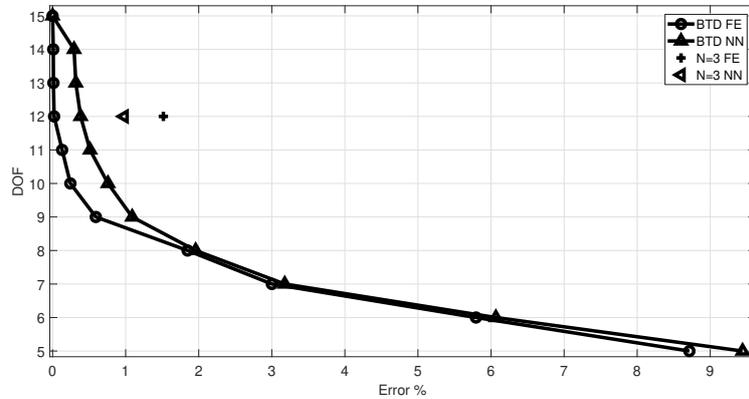
**Figura 8.5:** Curva BTD prevista per  $h/a = 0.05$ , con in input i valori di laminazione, condizioni al contorno e spessore. Tre strati interni con 15/25/25 neuroni

Questo tipo di soluzione, seppur per problemi semplici sembrerebbe funzionare, potrebbe entrare in crisi per gusci o piastre che presentino un grande numero di strati. Infatti, andando ad aumentare le cifre decimali si va a confondere l'errore di approssimazione con l'errore macchina. È sicuramente vero che l'errore macchina è dell'ordine di  $10^{-17}$ , ma una rete volta a dover effettuare delle previsioni sulla base di un numero finito di dati in input può non presentare univocità per un numero di cifre decimali molto minore. Si prenda per esempio un guscio multistrato con la seguente laminazione:  $0^\circ/90^\circ/0^\circ/90^\circ/0^\circ$  ed un altro con laminato nel seguente modo:  $0^\circ/90^\circ/0^\circ/90^\circ/90^\circ$ . Secondo la convenzione data in precedenza i due gusci avrebbero rispettivamente le seguenti codifiche: 0.01010 e 0.01011. Osservando tali valori è evidente come i due numeri siano molto prossimi nonostante la laminazione (conseguentemente quindi i risultati ad essa associati) sia completamente diversa tra i due casi. In un caso del genere la macchina, in fase di training, potrebbe confondere i due numeri come uguali e dare quindi risultati errati. Tale problema diventa tanto più importante quanto più si aumentano gli strati. La stessa problematica potrebbe essere riscontrata nel caso della determinazione delle condizioni al contorno. A differenza del numero di strati che coinvolge la laminazione, però, le condizioni al contorno (almeno per i casi più semplici), non possono presentare un numero troppo elevato di termini. Infatti, un guscio ha normalmente quattro lati nei quali possono essere presenti una condizione al contorno per ogni lato, sempre nei casi canonici. Per far fronte al tipo di problematica che si verifica nella determinazione della laminazione, si è quindi pensato di avere univocità della codifica andando a "pesare" le varie laminazioni del guscio e normalizzarle. Per fare questo si va a moltiplicare i gradi di orientamento delle fibre per il numero dello strato che si sta considerando; si sommano tutti gli strati e si divide per la somma dei vari orientamenti moltiplicato il numero di strati. In questo modo si ottiene un input che è sempre univoco e che si può identificare fino alla terza o quarta cifra decimale per piastre o gusci con numero di strati qualsiasi. Si consideri, per esempio, il guscio analizzato

precedentemente con laminazione  $0^\circ/90^\circ/45^\circ$ . In questo caso il valore da inserire come input sarà:

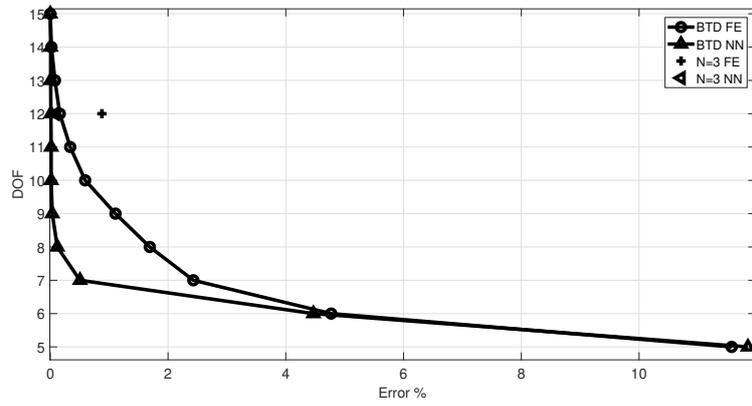
$$I = \frac{1 \cdot 0 + 2 \cdot 90 + 3 \cdot 45}{3 \cdot (0 + 90 + 45)} = 0.714$$

Per testare questa nuova tipologia di codifica si va, in prima analisi, a richiedere alla rete neurale la curva della Best Theory Diagram per un guscio le cui caratteristiche erano state utilizzate per allenare la rete neurale. Il guscio in questione ha un raggio di curvatura  $R/a = 0.2$ , spessore del guscio  $h/a = 0.15$  e laminazione  $0^\circ/90^\circ/0^\circ$  e presenta quindi come input il valore  $I = 0.667$ . Il risultato è riportato in figura 8.6. Si può osservare che la curva BTD ottenuta grazie alla rete neurale approssima molto bene la curva ottenuta tramite l'analisi agli elementi finiti. Si precisa che anche in questo caso è stato necessario aumentare il numero di neuroni per ogni singolo strato poiché il maggior numero di input comporta la necessità di avere una maggiore complessità della rete. Una seconda analisi è stata effettuata per andare a testare la solidità della rete per quanto



**Figura 8.6:** Curva BTD per un guscio con caratteristiche utilizzate in fase di training

riguarda la previsione di una curva BTD per un guscio con caratteristiche non presenti tra i gusci utilizzati per l'allenamento della rete. Questo tipo di verifica è indispensabile per poter essere confidenti del fatto che, una volta consolidata la rete, l'algoritmo sia in grado di prevedere un certo range di problematiche senza la necessità di dover cambiare input di training caso per caso. Il test viene effettuato su un guscio che presenta laminazione  $45^\circ/0^\circ/60^\circ$  ( $I = 0.714$ ), spessore  $a/h = 0.1$  e raggio di curvatura  $R/a = 0.2$ . Ciò che si è osservato in quest'ultima analisi è una crescente necessità di complessità da parte della rete. Infatti, per poter arrivare ad un risultato accettabile, come quello riportato in figura 8.7, si è dovuto aumentare il numero di neuroni per ogni singolo strato interno. Anche in questo caso la scelta di aumentare il numero stesso di strati interni si è rivelata inefficace. Altra considerazione da fare è che, come avveniva per lo spessore, la rete non è in grado di cogliere una laminazione se come codice di codifica ha un numero non compreso nell'intervallo di training. È necessario quindi utilizzare una popolazione di dati per il training che sia sufficientemente ampia da coprire tutte le combinazioni, tenendo in considerazione che per come è definita la codifica può assumere valore massimo pari ad uno.



**Figura 8.7:** Curva BTD prevista per un guscio con caratteristiche non presenti in fase di training



## Capitolo 9

# Rete multi target

### 9.1 Best Theory Diagram e modi di vibrare

Per un campo di applicazioni più esteso della rete neurale si impone la ricerca di più risultati contemporaneamente con la stessa rete neurale. Si cerca quindi di implementare quindi una rete multi target andando ad allenarla utilizzando per un set di input più dati in output. I target che verranno richiesti sono ancora una volta la curva Best theory Diagram e, contemporaneamente, un numero di modi di vibrare arbitrario. Per iniziare e per motivi di capacità computazionale, verrà richiesto il primo modo di vibrare insieme alla curva BTM. La rete neurale che deve ricercare più target contemporaneamente richiede un costo computazionale ed un tempo di calcolo decisamente superiore rispetto a quella che deve ricercare un solo input alla volta. Ad ogni nuovo target richiesto si allunga notevolmente il tempo di ricerca del risultato imposto. Per venire incontro al costo computazionale si utilizza la rete che in input considera solamente l'influenza dello spessore associata al set di dodici 0 ed 1 che vanno a definire i vari modelli strutturali. Essendo ancora una rete volta alla previsione di un risultato a partire da un set di input, si utilizza sempre l'algoritmo di training Backpropagation poiché, come riportato in letteratura, risulta essere il migliore per questo tipo di applicazioni. Il primo tentativo viene effettuato andando a ricercare, come anticipato, la curva BTM e il primo modo di vibrare di un guscio con spessore  $h/a = 0.15$ . La rete che dovrà ricercare i targets viene allenata utilizzando quattro gusci con spessori in un intervallo che comprende quello del guscio in esame, ma che non fa parte degli input del training. La rete, visto il maggior numero di output che deve fornire, necessita di un maggiore numero di neuroni per ogni strato che viene portato a 25 per gli strati interni, mentre lasciato a 13 per il layer di input. Il primo risultato che fornisce la rete neurale viene riportato in figura 9.1 dove si può notare un ottimo andamento della curva prevista dalla rete neurale confrontata con quella proveniente dalla FEM. Questo risultato è molto importante in quanto garantisce che aumentare il numero di target richiesti non va ad abbattere la qualità del singolo risultato. In figura 9.2 vengono invece riportate le previsioni della frequenza del primo modo di vibrare per ognuna delle  $2^{12}$  combinazioni date dal modello strutturale  $N = 4$  di partenza. Si può osservare come quelle provenienti dalla FEM e quelle provenienti dalla rete neurale siano per lo più sovrapposte o molto vicine, indice del fatto che la rete è stata correttamente allenata e le previsioni fornite sono accurate. Svolgendo l'analisi FEM si riscontra che il primo modo di vibrare ha una frequenza pari a 2.21 (la frequenza è stata adimensionalizzata con la seguente relazione:  $\omega \cdot 10\sqrt{\rho h^2/E_T}$ ) ed è quindi evidente, in linea con la teoria della Best Theory Diagram, vi sia una porzione di modelli strutturali che presenta una frequenza molto vicina a quella di riferimento (tali modelli presentano quindi l'errore più basso e dunque concorreranno alla generazione della curva BTM), mentre molti altri modelli strutturali presentano frequenze molto diverse da quella di partenze e ciò si traduce

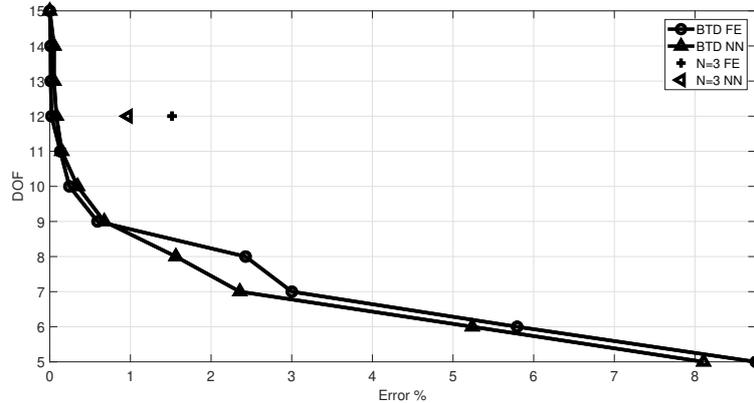


Figura 9.1: Curva BTD prevista per  $h/a = 0.15$

nel fatto che i termini trascurati per quel modello erano importanti per la determinazione del risultato finale.

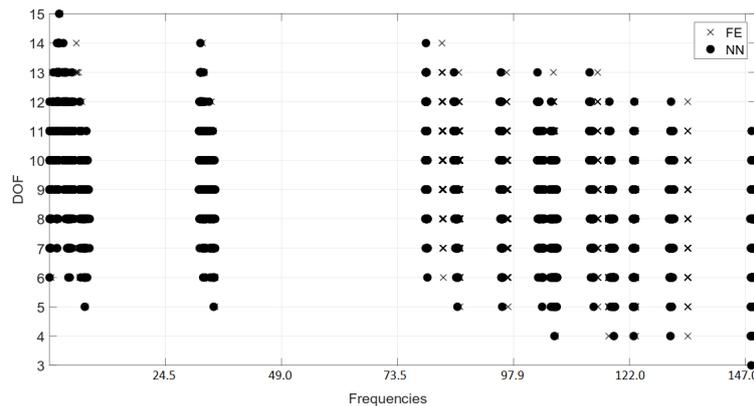


Figura 9.2: Frequenze previste per  $h/a = 0.15$

Per testare la solidità della rete si procede effettuando la stessa analisi, ma utilizzando un guscio con spessore diverso. In questo caso lo spessore è pari a  $h/a = 0.1$ . La rete rimane inalterata vista l'analogia del problema e si cerca di dimostrare che essa è ripetibile. In figura 9.3 viene riportata la curva BTD per il guscio in esame ottenuta come primo di due targets. Si nota che anche in questo caso la curva prevista dalla rete neurale è molto prossima a quella ottenuta con l'analisi agli elementi finiti. Anche in questo caso, quindi, non vi è alterazione di un risultato per via della richiesta di un secondo di diverso tipo. In figura 9.4 sono invece riportate le frequenze previste per il primo modo di vibrare per il guscio in esame. Il grafico è da intendersi come quello riportato in precedenza. Anche in questo caso le osservazioni da fare sono simili a quelle già fatte per il guscio con spessore  $h/a = 0.15$  poiché vi è una buona sovrapposizione tra frequenze ottenute con la rete neurale e frequenze ottenute attraverso la FEM. In questo caso la frequenza ottenuta dal modello di riferimento, per il primo modo di vibrare, è pari a 1.20. Si nota quanto già detto in precedenza, ovvero una porzione di modelli strutturali che forniscono risultati molto prossimi a quel valore e altri che invece non riescono a coglierlo. La differenza con 9.2 è che in questo caso

le frequenze vengono colte perfettamente dalla rete neurale, questo comportamento è dato dal fatto che lo spessore del guscio in esame è proprio uno di quelli utilizzati per il training della rete neurale. Questo risultato evidenzia come la rete sia suscettibile all'intervallo di training e ribadisce il concetto visto nei capitoli precedenti che più l'intervallo di training è folto e contenente il caso da analizzare, migliori saranno i risultati ottenuti. Naturalmente, per poter avere un intervallo sufficientemente accurato per tutti i casi sono necessarie molte analisi agli elementi finiti a priori e quindi maggior costo computazionale. Naturalmente, a seconda dell'errore massimo accettabile nel problema in esame, si sceglierà l'intervallo di training da utilizzare. A questo punto, avendo

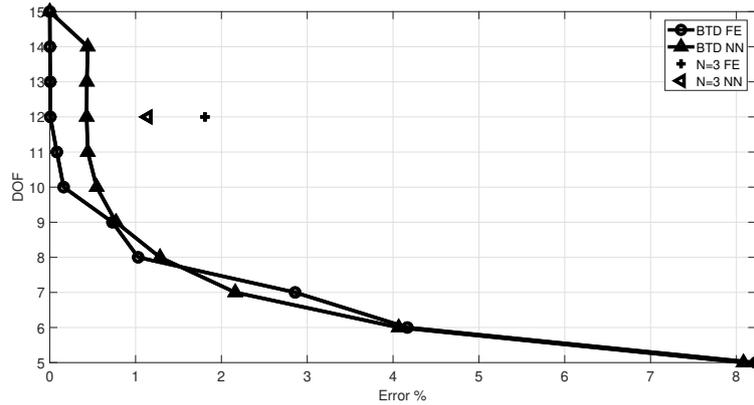


Figura 9.3: Curva BTD prevista per  $h/a = 0.1$

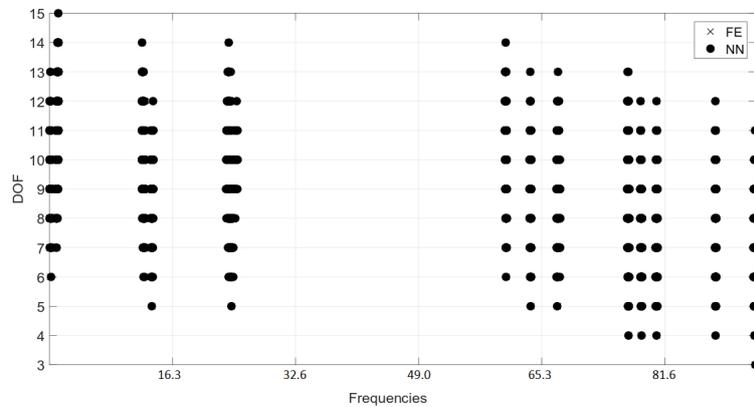
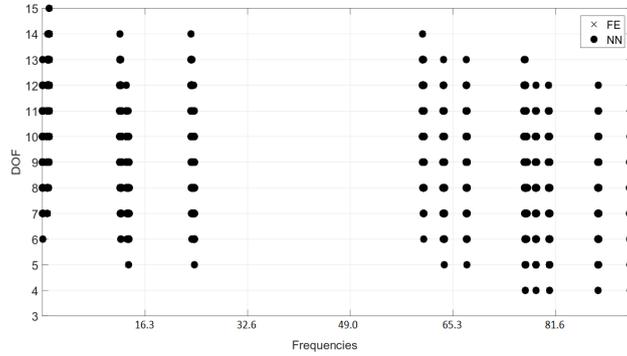


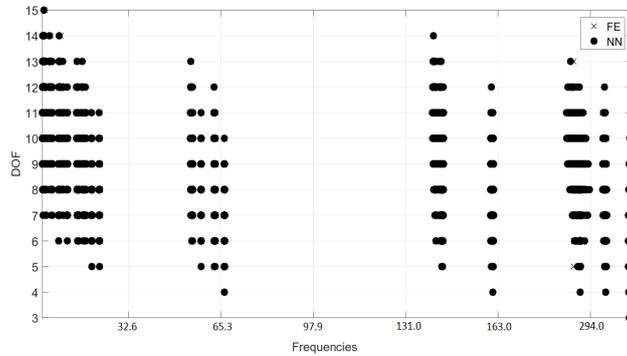
Figura 9.4: Frequenze previste per  $h/a = 0.1$

appurato che la rete è in grado di poter prevedere più risultati senza degradare la qualità del singolo risultato, è possibile andare a variare il multi target ed impostare come output della rete neurale un numero arbitrario di modi di vibrare. Naturalmente, andando ad aumentare molto il numero di target, aumenta molto il costo computazionale della rete e anche la necessità di avere una rete più complessa. Per questo motivo si riportano i casi in cui si richiede il primo ed il secondo modo di vibrare per diversi gusci. Il primo guscio ha uno spessore di  $h/a = 0.1$  e raggio di curvatura  $R/a = 0.2$ . Tale guscio è tra quelli utilizzati per effettuare il training della rete e si nota come infatti essa riesca a cogliere quasi perfettamente tutte le  $2^{tau}$  frequenze dei primi due modi

di vibrare. I risultati sono riportati in figura 9.5. A questo punto, ancora una volta, si va a testare



(a) Modo 1



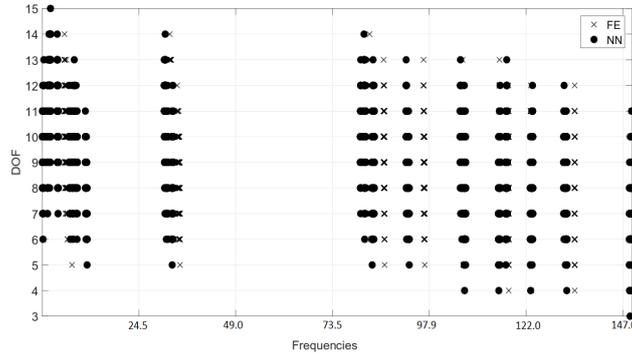
(b) Modo 2

Figura 9.5: Previsione dei primi due modi di vibrare

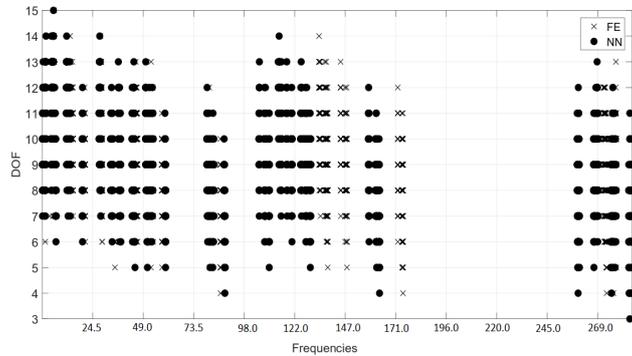
la capacità della rete di prevedere i targets imposti dall'operatore per un guscio con caratteristiche non utilizzate per il training della rete stessa. Per fare questo si utilizza nuovamente il guscio con spessore  $h/a = 0.15$  e si studia il comportamento della rete. Naturalmente, si poteva utilizzare una rete con in input anche la laminazione, ma ciò unito al multi target si sarebbe tradotto in un grande aumento di costo computazionale. I risultati vengono riportati in figura 9.6. Si può osservare che il modo 1 viene colto ancora una volta molto bene (naturalmente con accuratezza inferiore rispetto al caso precedente) ed infatti la maggior parte dei casi sono sovrapposti o molto prossimi a quelli ricavati dalla formulazione agli elementi finiti che sono il riferimento del problema. Per quanto riguarda il secondo modo di vibrare si nota che vi è, in alcuni punti del grafico, una maggiore dislocazione tra la frequenza prevista e quella di riferimento. Anche in questo caso la differenza è piuttosto contenuta, ma si prova ridurre aumentando i neuroni per ogni singolo strato interno. In figura 9.7 è riportato il risultato che si è ottenuto a fronte di un aumento del numero di neuroni per ogni singolo strato interno.

## 9.2 Analisi dinamica e statica

Un altro problema interessante che si vuole andare a risolvere sfruttando la tecnologia del Machine Learning è quello della previsione del Best Theory Diagram contemporaneamente per un



(a) Modulo 1

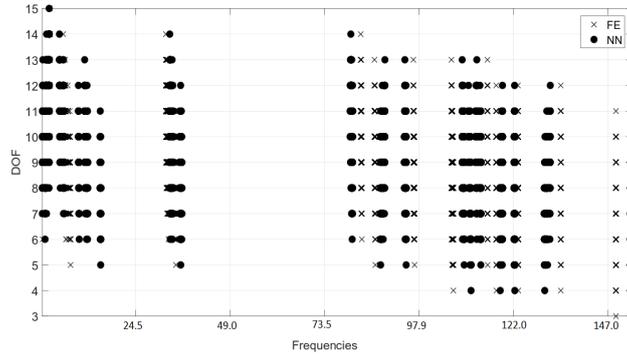


(b) Modulo 2

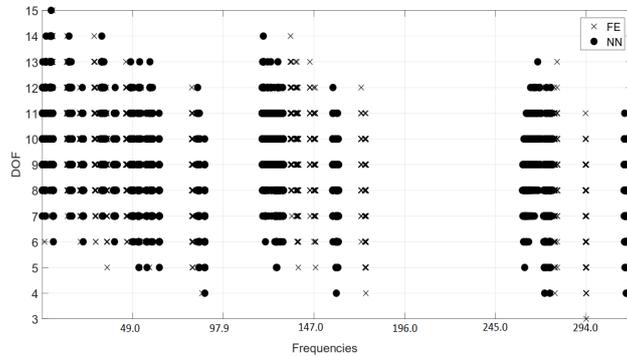
**Figura 9.6:** Previsione dei primi due modi di vibrare

problema statico e per un problema dinamico. In questo modo il risparmio in termini di costo computazionale sarebbe ancora più evidente in quanto si andrebbe ad evitare di effettuare due analisi agli elementi finiti con  $2^{tau}$  righe di risultato. Il vantaggio del Machine Learning con una rete neurale allenata in questo senso risulterebbe essere elevato. Per effettuare il training di una rete neurale con tale scopo si lasciano inalterati i dati in input, che continuano ad essere le  $2^{tau}$  combinazioni di modelli strutturali ridotti e lo spessore del guscio relativo (si hanno quindi, in totale, 13 input per il problema) e come target si imposta, oltre alla media delle frequenze dei primi 10 modi di vibrare, lo spostamento del guscio nella sua mezzeria. Si osserva che bisogna prestare attenzione a porre che i target siano relativi allo stesso modello strutturale ridotto. Inoltre, si precisa che per questo tipo di analisi, la complessità del problema è maggiore ed è quindi richiesta una maggiore capacità da parte della rete di effettuare calcoli difficili. Per questo motivo il numero di strati interni è stato portato a 5, mentre il numero di neuroni per ogni strato interno a 20. In questo modo il tempo richiesto per l'ottenimento del risultato è maggiore, ma è maggiore anche la qualità dello stesso. Inoltre, tale tempo aggiuntivo è giustificato anche dal tipo di risultato che si sta andando a ricercare.

Il training viene effettuato utilizzando tre gusci che differiscono nello spessore:  $h/a = 0.01$ ,  $h/a = 0.1$  e  $h/a = 0.2$  ed uguale raggio di curvatura  $r/a = 0.2$ . Per ottenere i dati da inserire nella rete neurale per effettuare il training si vanno ad eseguire analisi agli elementi finiti di tipo dinamico per ottenere le frequenze (è la stessa analisi effettuata finora per l'ottenimento del target) e analisi di



(a) Modulo 1



(b) Modulo 2

Figura 9.7: Previsione dei primi due modi di vibrare

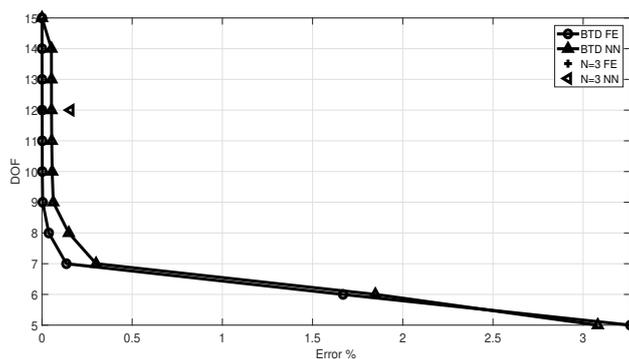
tipo statico per ottenere lo spostamento. Lo spostamento viene richiesto per un carico bisinusoidale agente nella mezzeria del guscio. Avendo premura di andare ad escludere dall'analisi i primi 3 termini si otterranno in totale 6 file contenenti l'analisi dinamica e l'analisi statica proveniente dagli elementi finiti.

Alla rete neurale si richiede quindi, dopo aver effettuato il training con i dati visti in precedenza, di prevedere la frequenza media dei primi 10 modi di vibrare e lo spostamento per ogni modello ridotto. Utilizzando questi risultati la rete sarà in grado di ottenere i grafici BTD per entrambi i problemi.

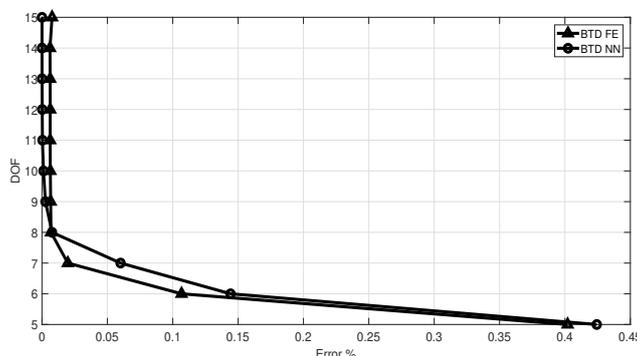
La prima analisi che si effettua è quella per un guscio di spessore  $h = 0.01$ , ovvero uno dei modelli utilizzato per il training. I risultati ottenuti sono riportati in figura 9.8. Ciò che si può osservare è un'ottima approssimazione di entrambe le curve, con scostamenti dal valore di riferimento ottenuto dall'analisi agli elementi finiti di pochi punti decimali percentuali. Un'altro grafico che è utile ad osservare la solidità di questa rete neurale è quello di figura 9.9 dove si può osservare l'andamento complessivo di tutti i  $2^{tau}$  risultati ottenuti dalla rete confrontati con quelli ottenuti dall'analisi agli elementi finiti. Anche in questo caso si osserva una complessiva sovrapposizione dei risultati ottenuti con l'analisi FEM e quelli ottenuti tramite la previsione di una rete neurale. Tali risultati così precisi sono anche dovuti al fatto che il guscio richiesto alla rete neurale è uno di quelli utilizzati per allenare la rete stessa.

Per testare l'effettiva solidità della rete si riporta nel seguito una seconda analisi per un guscio di spessore  $h/a = 0.1$  (quindi ancora uno utilizzato per il training della rete). Con i risultati di questa

seconda analisi si potrà verificare se la rete è effettivamente allenata in modo da dare risultati sempre accettabili e proseguire analizzando un terzo guscio il cui spessore (quindi una caratteristica degli input) non è stata utilizzata per allenare la rete. I risultati di questa seconda analisi sono riportati in figura 9.10. Come si può osservare la curva relativa all'analisi statica prevista dalla rete neurale, riportata in figura 9.10b, è pressoché sovrapposta a quella di riferimento ottenuta con l'analisi agli elementi finiti. Per quanto riguarda la curva prevista per l'analisi dinamica, riportata in figura 9.10a, vi è un leggero scostamento con quella di riferimento, ma è di un valore tale da rendere il risultato comunque molto buono. La riprova di questi due comportamenti è riportata in figura 9.11, in cui si nota come la differenza tra i risultati statici previsti e di riferimento sia pressoché impossibile da percepire, mentre vi è un po' di differenza per l'analisi dinamica.



(a) Analisi dinamica

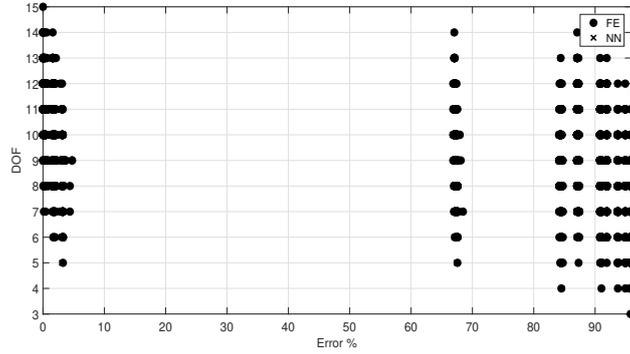


(b) Analisi statica

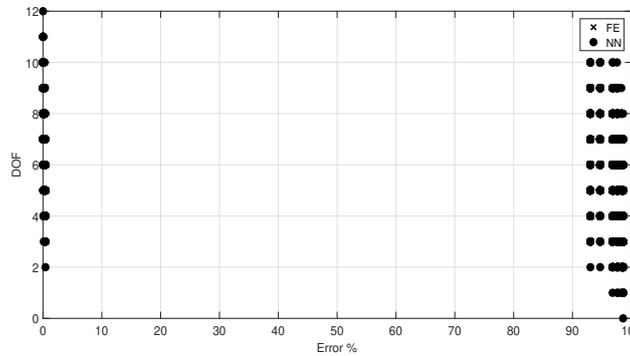
**Figura 9.8:** Previsione curve BTD per guscio con  $h/a = 0.01$ 

A questo punto si va a verificare la capacità della rete neurale di prevedere risultati accettabili anche per gusci per i quali non sono state utilizzate le stesse caratteristiche in fase di training. Come già detto, questo è il caso reale di impiego della rete neurale, in quanto ci permette di ottenere, attraverso un numero finito di simulazioni, un potenziale numero infinito di risultati. A tal proposito si richiede alla rete neurale la curva BTD (sia statica che dinamica) per un guscio di spessore  $h/a = 0.15$ .

Quello che si osserva andando ad analizzare la figura 9.12 e 9.13 è un risultato globalmente meno preciso di quelli ottenuti con i due gusci precedenti, ma un buon risultato per quanto riguarda le due curve BTD. Tale andamento dei risultati era da attendersi in quanto l'input dello spessore non era tra quelli utilizzati per il training della rete e quindi la sua influenza viene prevista sulla



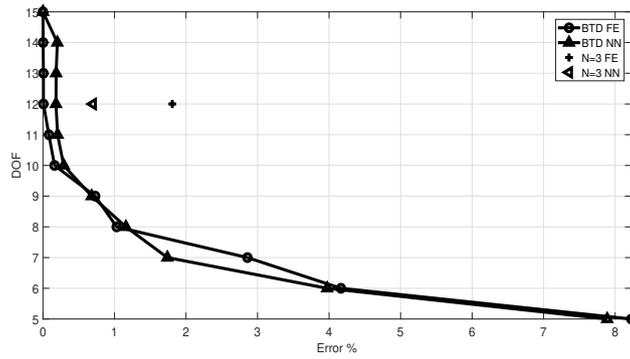
(a) Analisi dinamica



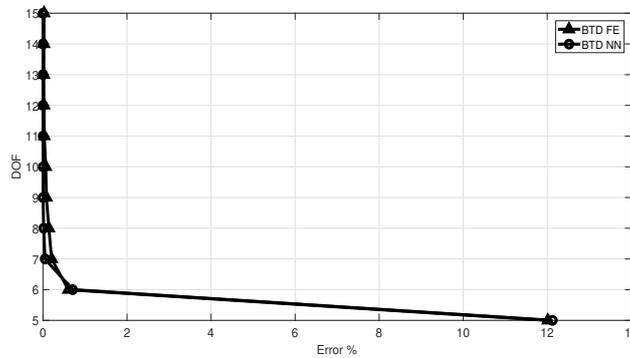
(b) Analisi statica

Figura 9.9: Risultati per guscio con  $h/a = 0.01$

base degli altri spessori sulla quale si è allenata. È da sottolineare che, anche in questo caso, lo spessore di previsione utilizzato è compreso nell'intervallo di training della rete neurale, in quanto, come già dimostrato in precedenza, se così non fosse la rete non sarebbe in grado di prevedere con sufficiente precisione l'influenza dell'input. Analizzando nel dettaglio i risultati riportati nelle figure 9.12 e 9.13, possiamo affermare che l'analisi dinamica produce lo stesso risultato di quella effettuata in precedenza quando essa era l'unico target della rete neurale. Facendo infatti riferimento alla figura 8.1b, si può affermare che le curve previste nei due casi sono pressoché identiche (si ricordi che le leggere differenze tra le varie curve sono imputabili al metodologia di scelta dei dati di input della rete neurale che è di tipo casuale). Questo primo risultato è molto importante perché dimostra che avere aumentato il numero di target della rete non ha intaccato la qualità dei risultati che la stessa rete era in grado di prevedere, permettendo quindi di essere confidenti nell'aumentare il numero di target (prestando sempre attenzione al tempo di calcolo aggiuntivo che ogni nuovo output richiede). Per quanto riguarda l'analisi statica, si può osservare come la curva BTD ottenuta per previsione dalla rete neurale sia molto prossima a quella ottenuta per l'analisi agli elementi finiti. È infatti vero che il discostarsi delle due curve è per uno scarto inferiore al mezzo punto percentuale. Il risultato a più bassi gradi di libertà è invece colto perfettamente. Per quanto riguardano tutti i  $2^{tau}$  output colti dalla rete stessa, questi sono in gran parte sovrapposti. Vi sono degli scostamenti nelle zone ad alta percentuale di errore, ma comunque piuttosto contenute.



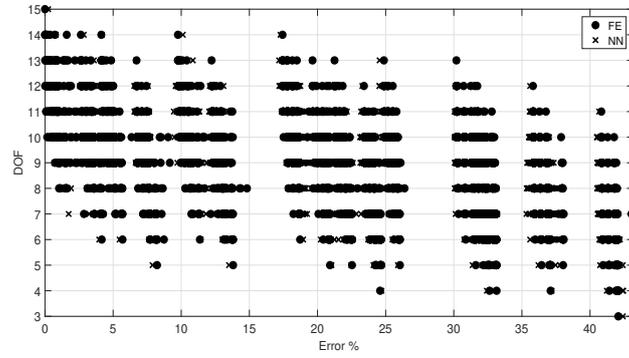
(a) Analisi dinamica



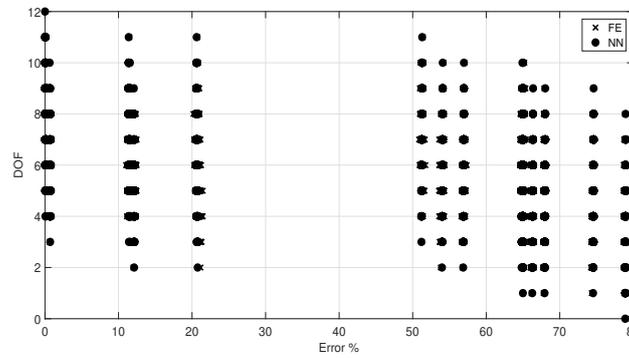
(b) Analisi statica

Figura 9.10: Previsione curve BTD per guscio con  $h/a = 0.1$

Infine, nelle tabelle 9.1 e 9.2 sono riportati i modelli strutturali ridotti per i gradi di libertà crescenti, a partire dalla teoria FSDT fino ad arrivare al modello strutturale completo e di riferimento con 15 gradi di libertà. Osservando le due tabelle si nota come, sia nel caso dinamico che nel caso statico, i termini di primo e terzo grado di espansione siano i più importanti per l'ottenimento di un risultato con il minimo errore rispetto al modello completo. Naturalmente i termini di ordine zero sono sempre considerati nella loro interezza per via dell'impostazione del problema. Vi è una differenza di importanza tra il caso dinamico ed il caso statico nei termini al secondo ed al quarto ordine. Infatti, per il problema dinamico la maggior importanza dei termini deriva da quelli al quarto ordine, mentre per il caso statico da quelli al secondo ordine.



(a) Analisi dinamica

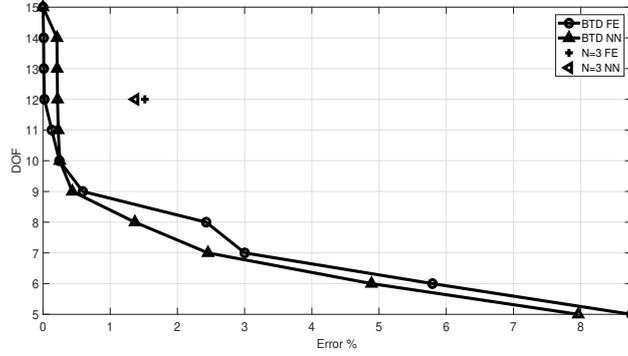


(b) Analisi statica

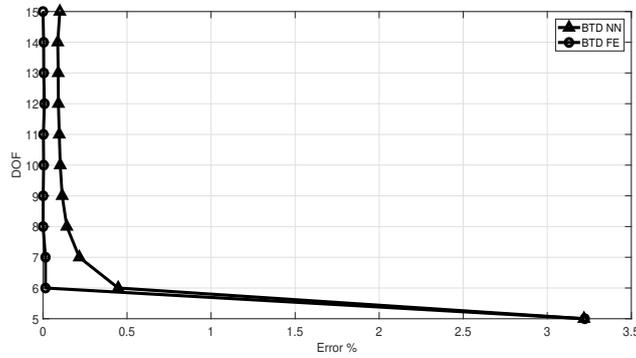
Figura 9.11: Risultati per guscio con  $h/a = 0.1$

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
5	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△
7	▲	▲	▲	▲	▲	△	△	△	△	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	△	△
9	▲	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	▲	△
10	▲	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	▲	▲
11	▲	▲	▲	▲	▲	△	△	▲	△	▲	▲	△	△	▲	▲
12	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲	△	▲	▲
13	▲	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
	RF <sub>0</sub> = 1.00			RF <sub>1</sub> = 0.67			RF <sub>2</sub> = 0.37			RF <sub>3</sub> = 0.70			RF <sub>4</sub> = 0.43		

Tabella 9.1: Modello BTd  $h = 0.15$ , analisi dinamica, multi target



(a) Analisi dinamica

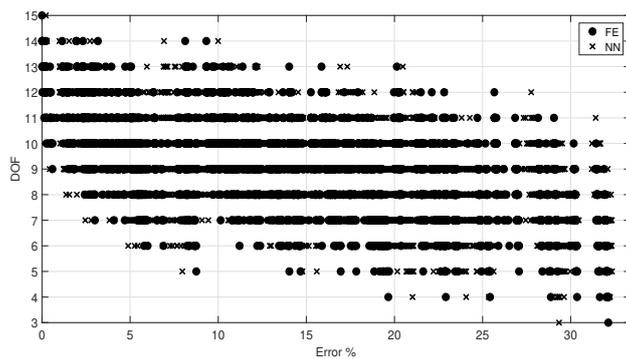


(b) Analisi statica

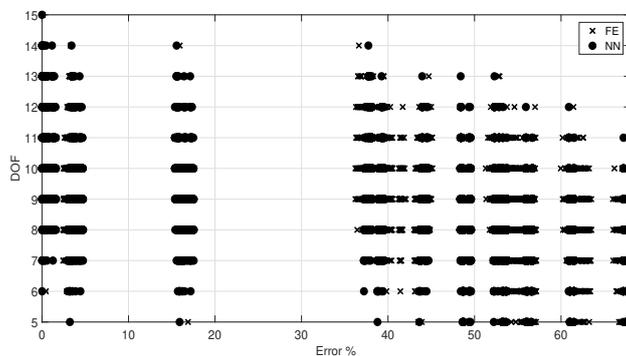
Figura 9.12: Previsione curve BTD per guscio con  $h/a = 0.15$

DOF	$u_{x1}$	$u_{y1}$	$u_{z1}$	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
5	▲	▲	▲	△	▲	△	△	△	△	▲	△	△	△	△	△
6	▲	▲	▲	▲	▲	△	△	△	△	▲	△	△	△	△	△
7	▲	▲	▲	▲	△	△	△	△	△	▲	▲	△	△	△	△
8	▲	▲	▲	▲	▲	△	△	▲	▲	▲	△	△	△	△	△
9	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	△	△	△	△
10	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	△	△	△	△
11	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	△	▲	▲
12	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	▲	△	▲	▲
13	▲	▲	▲	▲	▲	△	△	▲	▲	▲	▲	▲	▲	▲	▲
14	▲	▲	▲	▲	▲	▲	△	▲	▲	▲	▲	▲	▲	▲	▲
15	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
	$RF_0 = 1.00$			$RF_1 = 0.67$			$RF_2 = 0.50$			$RF_3 = 0.67$			$RF_4 = 0.33$		

Tabella 9.2: Modello BTD  $h = 0.15$ , analisi statica, multi target



(a) Analisi dinamica



(b) Analisi statica

Figura 9.13: Risultati per guscio con  $h/a = 0.15$

# Capitolo 10

## Problema inverso

### 10.1 Singolo input

Un'interessante applicazione di questa rete neurale potrebbe essere quella del problema inverso, ovvero andare ad invertire gli inputs ed i targets utilizzati nel corso della tesi fino a questo momento. Nel seguito quindi si andrà a costruire una rete neurale la quale, dato come input la frequenza di un modo di vibrare, è in grado di restituire come output il modello strutturale che permette di ottenere quel valore di frequenza. Naturalmente per questo tipo di applicazione ci sono diverse problematiche date dal fatto che si passa dall'avere  $\tau$  inputs che vanno a definire un solo target all'avere un solo input per definire ben  $\tau$  (o più) targets. la prima problematica che ci si aspetta è però che, aumentando molto il numero di targets, la rete neurale avrà un costo computazionale molto alto e quindi tempi di calcolo decisamente elevati. Tenendo a mente che la rete neurale viene costruita ed utilizzata per andare a ridurre i tempi di calcolo è necessaria una valutazione in termini di convenienza. Un'altra problematica è che nelle  $2^{\tau}$  combinazioni il singolo termine del modello strutturale si trova spento ed acceso più volte con, come risultato, diverse frequenze. Questo tipo di problema rende impossibile allenare  $\tau + 1$  reti neurali che diano come output uno solo dei termini del modello strutturale (soluzione che avrebbe abbassato notevolmente il tempo di calcolo). Infatti, per quello che è stato appena detto, la rete si troverebbe a gestire input uguali ed output diversi non convergendo adeguatamente. Per avere l'univocità dell'input ci possono essere diverse soluzioni, accomunate dal fatto che, in questa prima analisi, non verranno considerate tutte le  $2^{\tau}$  combinazioni che danno vita al modello ridotto a partire da quello completo riportato di seguito.

$$\begin{aligned}u_x &= u_{x_1} + z u_{x_2} + z^2 u_{x_3} + z^3 u_{x_4} + z^4 u_{x_5} \\u_y &= u_{y_1} + z u_{y_2} + z^2 u_{y_3} + z^3 u_{y_4} + z^4 u_{y_5} \\u_z &= u_{z_1} + z u_{z_2} + z^2 u_{z_3} + z^3 u_{z_4} + z^5 u_{z_5}\end{aligned}\tag{10.1}$$

In questa prima fase si vanno a considerare solamente quei modelli strutturali che presentano, alternativamente, tutti i termini in x, y e z o accesi o spenti. In questo modo vi sono 8 diversi modelli strutturali ai quali è legata una frequenza. I target e la loro nomenclatura vengono riportati in tabella 10.1. In questo modo ho la possibilità di avere target diversi per input diversi. Naturalmente, quello che si può subito notare è che la rete avrà a disposizione, per il singolo guscio un set molto ridotto di input/target per effettuare il training.

Data la complessità del problema sono necessari un numero crescente di strati interni affinché il risultato prodotto possa essere accettabile. La prima prova che si effettua è su un guscio di spessore  $h/a = 0.01$  e raggio di curvatura  $h/a = 0.2$ . In questo primo caso sia il training che il risultato verranno ricercati sullo stesso guscio per poter capire la fattibilità della risoluzione del

problema e la qualità a cui si può aspirare.

Come detto, quindi, i target sono quelli nella tabella 10.1. Il primo set di input/output è riportato in tabella 10.2 e si può notare come si presenti una nuova difficoltà nel problema dovuto alla similitudine tra i vari errori che sono dati dai modelli strutturali in input. Addirittura, il modello 001 e 000 hanno lo stesso input ed infatti l'output è identico per entrambi. Questo tipo di problematica si verifica poiché l'influenza dei termini in questo set di input è praticamente la stessa. Infatti, scansionando il file contenente le  $2^{tau}$  combinazioni di modelli strutturali ridotti, con i relativi errori per ognuno dei primi dieci modi di vibrare, si nota che l'errore del primo modo di vibrare è lo stesso per decine di modelli ridotti. Tale comportamento si verifica poiché l'influenza di un set di termini mancanti è simile se non identica ad un'altra. A tal proposito non è possibile andare a migliorare il risultato in quanto, se l'input è lo stesso la rete neurale non è in grado di prevedere due output differenti. Tuttavia, esclusi gli ultimi due output, i primi si riescono a differenziare anche se non di molto. Infatti, il modello strutturale codificato con 111 è stato previsto dalla rete con un 110.0026, che potrebbe confondersi con il modello 110. Quest'ultimo modello strutturale è tuttavia previsto dalla rete con un 109.9998. Si può osservare che la codifica del modello strutturale viene sempre prevista a coppie. Ovvero, come visto i modelli 111 e 110 sono previsti da due numeri vicini a 110, ma il primo leggermente superiore (si sta riferendo a 111) e il secondo leggermente inferiore (si sta riferendo a 110). Infatti, tale comportamento è anche visibile per i modelli 101, 100, 011 e 010.

A questo punto, quello che si prova a fare, è vedere se esiste un modo per separare maggiormente i vari risultati dei modelli strutturali. Per far ciò, si torna ad usare la media degli errori dei primi dieci modi di vibrare (ovvero lo stesso che si è utilizzato fino a questo momento per ricavare le curve della Best Theory Diagram). Andando a combinare tutte le dieci frequenze si auspica che vi sia maggiore differenza tra un modello strutturale e l'altro, di modo che la rete neurale possa riuscire a distinguere la provenienza dei vari input. Il guscio rimane lo stesso e gli input e gli output sono riportati in tabella 10.3. Osservando gli input di questo secondo problema si osserva come questi siano ancora molto simili a coppie, ma vi è un miglioramento sugli ultimi due che in questo caso sono diversi seppur di poco. Si osserva che l'andamento dei risultati previsti segue la stessa legge di quelli ricavati in precedenza.

A questo punto si eseguono gli stessi passaggi per un guscio di spessore diverso, in modo da poter testare la rete neurale con input diversi. Si procede quindi ad analizzare un guscio con rapporto  $h/a = 0.4$ . Anche qui si effettua dapprima l'analisi utilizzando in input il primo modo di vibrare. I risultati sono riportati in tabella 10.4. Le osservazioni da farsi sono pressoché le stesse fatte per il caso precedente, poiché anche in questo caso gli input sono molto simili tra di loro (gli ultimi sono uguali) e quindi anche gli output sono molto vicini tra di loro. Quello che si può osservare in questo caso, diversamente dal caso precedente, è che per i modelli strutturali ridotti studiati gli errori in gioco sono molto più bassi. Infatti, precedentemente erano quasi tutti oltre il 90%, mentre in questo caso sono tutti al di sotto del 40%.

Si procede, anche in questo caso, ad analizzare l'errore medio dei primi dieci modi di vibrare. Quello che si osserva in questo caso, analizzando la tabella 10.5, è che vi sono coppie di modelli strutturali che presentano errori molto simili (ad esempio 100 e 001). Come accadeva nei casi precedenti, la rete neurale non è in grado di distinguerli ed offre in output pressoché lo stesso risultato. Si cerca quindi di superare tale problematica andando a moltiplicare ognuno degli input per un coefficiente moltiplicativo che lo individui unicamente. Come prima prova si va a moltiplicare ogni input per la propria posizione e dividendo per il numero di input totale, ovvero:

$$I = \frac{Err/100 \cdot pos}{I_{tot}}$$

I risultati sono riportati nella tabella 10.6 e ora l'input non è più l'errore percentuale medio, ma un valore arbitrario calcolato con la formula vista in precedenza. Si può notare che in questo caso non vi sono input troppo simili ed infatti i modelli strutturali sono colti tutti abbastanza bene.

Andando a confrontare 10.6 e 10.5 si può osservare un miglioramento dei risultati piuttosto sensibile. Per testare la solidità di questo metodo si va ad analizzare un terzo guscio, concentrandosi sull'ultimo metodo e quindi effettuando una sola analisi. Il guscio in esame ha rapporto  $h/a = 0.2$  e mantiene invariate le altre caratteristiche. Gli input e relativi output di questo problema sono riportati nella tabella 10.7. In questo caso i risultati ottenuti dalla rete neurale sono molto buoni in quanto riesce a prevedere in maniera esatta 6 modelli strutturali su 8. Utilizzando la codifica introdotta si è potuto evitare di avere errori percentuali identici tra i vari modelli strutturali e quindi poter andare a prevederli con maggiore accuratezza.

A questo punto si prova a modificare leggermente il principio di funzionamento della rete neurale per andare però a risolvere il medesimo problema. Quello che si vuole andare a cambiare è il tipo di target che si richiede alla rete, andandolo a suddividere in tre parti da fornire ad altrettante sotto reti. Facendo sempre riferimento alla tabella 10.1 (la cui macrostruttura dei target è sempre la stessa), si costruisce un nuovo set di target da fornire. Per il nuovo problema si allenano tre diverse reti per ognuno dei termini  $x$ ,  $y$  e  $z$  dell'equazione di partenza 10.1. Quello che si ottiene per ognuna delle singole sotto-reti sono due possibili target: 0 ed 1. A tale valore corrisponderà il termine acceso o spento che la rete rappresenta. L'input è sempre, per tutte le sotto-reti, l'errore (che può essere quello medio o quello di un singolo modo di vibrare) codificato come visto in precedenza. Andando poi a combinare tutti gli output delle sotto-reti si otterranno i modelli strutturali relativi all'input. Quello che ci si aspetta da questo tipo di soluzione è una maggiore precisione del risultato in quanto il singolo sotto elemento della rete neurale deve andare a prevedere semplicemente uno 0 oppure un 1, differentemente da quanto visto prima in cui invece doveva prevedere un numero di tre cifre. Una prima analisi la si effettua con un guscio identico a quello analizzato in tabella 10.7 in modo da poterne confrontare i risultati. Gli output della nuova rete, relativi a questo primo problema, sono riportati in tabella 10.8. I tre output si riferiscono rispettivamente all'elemento  $x$ ,  $y$  e  $z$ . Come si può osservare, la qualità dei risultati è molto elevata, in quanto sia lo 0 che l'1 sono colti molto bene (nel primo caso con numeri molto bassi, nel secondo con esattamente il dato corretto). Nella tabella 10.9 si riportano i risultati per un guscio con spessore avente rapporto  $h/a = 0.4$ , sempre con input i primi dieci modi di vibrare codificati. I risultati, in questo caso, sono leggermente peggiorati rispetto al caso visto in precedenza. Detto questo il modello strutturale è sempre riconoscibile in maniera piuttosto univoca, salvo per i casi 110 e 001 in cui vi è l'incertezza su entrambi i terzi elementi, ovvero il termine in  $z$ . Per completezza si riportano in tabella 10.10 i risultati ottenuti per un guscio di spessore  $h/a = 0.01$  e il resto dei dati invariati. Si nota come anche in questo caso la difficoltà maggiore del codice risiede nel prevedere con precisione il termine in  $z$ . Tuttavia, anche per questo problema i modelli strutturali sono identificabili piuttosto bene grazie ai risultati forniti dalla rete neurale.

TARGET	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
000	△	△	△	△	△	△	△	△	△	△	△	△
001	△	△	▲	△	△	▲	△	△	▲	△	△	▲
010	△	▲	△	△	▲	△	△	▲	△	△	▲	△
100	▲	△	△	▲	△	△	▲	△	△	▲	△	△
011	△	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲
101	▲	△	▲	▲	△	▲	▲	△	▲	▲	△	▲
110	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
111	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲

**Tabella 10.1:** Tabella target

INPUT	6.0285E-12	0.368511	96.8105	96.8106	97.5777	97.5778	98.6039	98.6039
OUTPUT	110.0026	109.9998	100.0026	99.9999	10.5023	10.4984	0.4995	0.4995
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.2:** Risultati per guscio  $h/a = 0.01$ , usando come input l'errore sul primo modo di vibrare

INPUT	0.00	1.5630	90.8148	90.8234	84.3712	84.4747	95.7717	95.7784
OUTPUT	110.6488	110.6426	100.5995	100.5204	10.6161	10.5464	0.8292	0.8286
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.3:** Risultati per guscio  $h/a = 0.01$ , usando come input la media degli errori dei primi dieci modi di vibrare

### 10.1.1 Problema inverso con base FSDT

In questa sezione si vuole andare a variare la tipologia di scelta degli errori dati in input e, conseguentemente, il target richiesto alla rete neurale. Per far questo si va a considerare una serie di modelli strutturali ridotti la cui base di partenza è quella del modello First-Order Shear Deformation Theory (FSDT) che è la seguente:

$$\begin{aligned}
 u_x &= u_{x_1} + z u_{x_2} \\
 u_y &= u_{y_1} + z u_{y_2} \\
 u_z &= u_{z_1}
 \end{aligned}
 \tag{10.2}$$

In questo caso quindi il modello 000 si riferirà al modello FSDT con termini 2 attivi su x e y ed un termine attivo su z. Nello specifico, nella tabella 10.11 sono riportati tutti i target di questo problema. In questa tabella è da notare che i termini  $u_{x_2}$  e  $u_{y_2}$  sono sempre attivi proprio per via della natura del problema che stiamo andando a risolvere. Il tipo di procedura che si va a svolgere è lo stesso effettuato precedentemente, andando a verificare il comportamento della rete neurale sul problema del riconoscimento di un numero limitato di modelli strutturali. In questo caso non è necessario andare a codificare l'errore percentuale medio dei primi dieci modi di vibrare in quanto i modelli così scelti presentano una differenza di errore che, seppure piccola, consente alla rete neurale di distinguere i vari modelli strutturali. Anche in questo caso gli errori sono stati divisi per 100 in modo da avere come input un numero compreso tra 0 ed 1 così da consentire alla rete neurale di lavorare meglio.

Le analisi verranno svolte inizialmente sull'errore medio dei primi dieci modi di vibrare per gusci di spessore  $h/a = 0.01, 0.1, 0.2e0.4$ . Nelle tabelle 10.12, 10.13, 10.14 e 10.15 sono riportati i risultati. La nomenclatura è la stessa delle tabelle viste in precedenza. Nei primi due casi si può notare un'ottima corrispondenza tra il target e l'output prodotto dalla rete neurale. Invece, nel caso di guscio con spessore  $h/a = 0.2$  (tabella 10.14) i risultati sono leggermente peggiori in quanto i vari input che vengono dati alla rete sono molto più simili tra di loro. In questo caso sarebbe utile andarli a codificare nonostante nei primi due casi invece i risultati sono molto buoni. Nell'ultima analisi (tabella 10.15), eccetto per il modello strutturale 010, i risultati tornano ad essere molto buoni con modelli strutturali previsti in modo perfetto.

A questo punto si va a ripetere l'analisi effettuata sulla media degli errori dei primi dieci modi di vibrare, andando ad utilizzare come input per l'analisi l'errore sul singolo modo di vibrare. Quello che ci si aspetta dopo le problematiche riscontrate nella sezione 10.1 è che lo scarto tra gli errori dati dai vari modelli strutturali ridotti sul singolo modo sia più bassa rispetto a quella data dalla media degli errori. Per via di questa caratteristica del problema ci si potrebbero aspettare risultati di più bassa qualità rispetto a quelli appena analizzati.

Nelle tabelle 10.16, 10.17, 10.18 e 10.19 sono riportati i risultati relativi al problema per gusci di spessore  $h/a = 0.01, 0.1, 0.2e0.4$ . Andando ad analizzare la tabella 10.16 si riscontra proprio quanto

INPUT	2.2826E-12	0.5966	27.0399	27.1825	19.5434	19.9991	39.3663	39.3663
OUTPUT	110.9263	109.9894	100.9228	100.6253	10.9933	9.9974	0.9997	0.9997
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.4:** Risultati per guscio  $h/a = 0.4$ , usando come input l'errore sul primo modo di vibrare

INPUT	0.00	6.5869	23.7712	31.4343	5.2521	10.5090	28.6862	35.5287
OUTPUT	110.9117	109.8135	100.3750	1.0397	11.2575	10.0583	1.0022	1.0671
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.5:** Risultati per guscio  $h/a = 0.4$ , usando come input la media degli errori dei primi dieci modi di vibrare

detto, ovvero input molto simili tra di loro che non permettono alla rete neurale di distinguere i vari casi. Se si osservano le coppie di modelli strutturali ridotti 010, 000 e 110, 100 si può notare come presentano a due a due gli stessi input. Gli altri quattro casi, invece, seppur hanno input diversi questi sono molto piccoli e quindi sono anch'essi confondibili. Dato questo fatto gli output della rete neurale non sono precisi ed in molti casi i modelli strutturali previsti non sono facilmente distinguibili. Per quanto riguarda il guscio con spessore  $h/a = 0.1$  (tabella 10.17) le osservazioni sono ancora le stesse e si può andare a vedere che sia in questo caso che nel caso precedente la rete che lavora sul termine in  $y$  è quella che fatica maggiormente nel risolvere il problema. Analizzando invece la tabella 10.19 si nota come la rete neurale riesca a prevedere in maniera piuttosto chiara i vari modelli strutturali ridotti. Il miglioramento è nettamente visibile proprio nella rete che si occupa del termine in  $y$  in quanto, in questo caso, i termini accesi e spenti sono perfettamente distinguibili. Questo tipo di risultato è sicuramente dovuto al fatto che a mano a mano che si è andati ad aumentare lo spessore del guscio in esame lo scarto tra i vari errori è sempre di più aumentato. A differenza dei casi in cui si utilizzava la media degli errori sui primi dieci modi di vibrare, in questo caso la codifica risulta essere necessaria per poter distinguere i vari input in tutti i casi.

### 10.1.2 Errori arbitrari in input

Per proseguire l'analisi sulla rete in grado di prevedere il modello strutturale partendo dall'errore, è interessante andare a studiarne il comportamento quando l'errore imposto come input non è esattamente tra quelli usati per il training, ma è arbitrario. Si vuole andare a verificare se la rete è in grado di riconoscere il modello che produce un errore sulla frequenza che è il più possibile vicino a quello imposto come input del problema. Per questo tipo di analisi sicuramente ci si aspetta un comportamento della rete neurale decisamente meno preciso in fase di previsione in quanto, come visto precedentemente, i modelli strutturali ridotti che è in grado di prevedere la rete sono molto vicini a coppie. Infatti, in questo caso non è più possibile andare a codificare l'errore percentuale prodotto dal modello strutturale ridotto in quanto l'input immesso dall'utente nella rete neurale è proprio un errore percentuale. La possibile somiglianza dei modelli strutturali che verranno previsti dalla rete neurale potrebbe non essere un problema grave se la previsione avviene a cavallo tra due modelli ridotti con errori simili e vicini all'input scelto. Inoltre, come visto nei primi capitoli di questa tesi (ad esempio in figura 4.11), per gusci diversi esistono errori massimi diversi. Per questo motivo non è possibile andare ad indagare lo stesso set di errori percentuali arbitrari per tutti i gusci.

Nelle tabelle 10.20, 10.21, 10.22 e 10.23 sono riportati i risultati di questa analisi (riferiti a dei gusci di spessore  $h/a = 0.1, 0.2, 0.25e0.4$ ). Nella riga INPUT vengono riportati gli errori scelti arbitrariamente, le righe degli OUTPUT hanno lo stesso significato riportato nelle analisi precedenti. Nella riga INPUT TRAINING vengono riportati gli errori percentuali (sia in questo

INPUT	0.00	0.0165	0.0891	0.1572	0.0328	0.0788	0.2510	0.3553
OUTPUT	110.8244	108.7630	100.9778	100.0220	11.1923	10.1863	1.0558	0.2293
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.6:** Risultati per guscio  $h/a = 0.4$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati

INPUT	0.00	0.0016	0.0769	0.1042	0.0632	0.0807	0.2405	0.2783
OUTPUT	110.0488	110.0000	101.0000	100.0000	11.0000	10.0000	2.5005E-06	6.9847E-07
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.7:** Risultati per guscio  $h/a = 0.2$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati

caso che nel caso precedente sono divisi per 100 in modo da avere sempre valori sotto l'unità) usati in fase di allenamento della rete a cui è associato il target del training stesso. Gli errori percentuali su cui si allena la rete neurale sono dati dalla media degli errori sui primi dieci modi di vibrare. Si può notare anche che l'ordine degli input forniti alla rete neurale è del tutto casuale in modo da non rendere l'analisi meccanica (nel caso 10.23 che è quello più generico utilizzato per la spiegazione dei vari risultati).

Prendendo come riferimento la tabella 10.23 (i commenti sono analoghi per gli altri casi) si può notare come, globalmente ed in modo meno preciso rispetto alle analisi fatte fino a questo momento (questo tipo di problematica era stata prevista), i modelli strutturali ridotti associati all'errore fornito sono proprio quelli che danno come errore effettivo quello più simile possibile.

- Il primo input si riferisce ad un errore del 90%; tuttavia un tale errore è superiore a qualsiasi prodotto dai modelli ridotti e per questo motivo fornisce in output il modello ridotto relativo all'errore più grande, ovvero 000;
- Il secondo input si riferisce ad un errore del 5%; la rete neurale riesce a fornire in maniera univoca il modello strutturale 011, ovvero quello che produce un errore del 5.25%;
- Il terzo input si riferisce ad un errore del 10%; la rete neurale riesce a fornire in maniera univoca il modello strutturale 010, ovvero quello che produce un errore del 10.51%;
- Il quarto input si riferisce ad un errore del 25%; la rete neurale riesce a fornire in maniera univoca il modello strutturale 101, ovvero quello che produce un errore del 23.77%;
- Il quinto input si riferisce ad un errore del 35%; la rete neurale riesce a fornire in maniera univoca il modello strutturale 000, ovvero quello che produce un errore del 35.53%;
- Il sesto input si riferisce ad un errore del 20%; la rete neurale riesce a fornire in maniera univoca il modello strutturale 101, ovvero quello che produce un errore del 23.77%. È il secondo caso che viene fornito tale modello strutturale ridotto poiché in entrambi i casi era quello che aveva l'errore che si avvicinava di più a quello imposto;
- Il settimo input si riferisce ad un errore del 1%; la rete neurale riesce a fornire in maniera univoca il modello strutturale 000, ovvero quello che produce un errore del 0%;
- L'ottavo input si riferisce ad un errore del 1%; la rete neurale riesce a fornire in maniera univoca il modello strutturale 000, ovvero quello che produce un errore del 0%.

È interessante notare come ad errore uguale, per guscio diverso, corrisponde modello strutturale ridotto diverso. Se, per esempio, si va a considerare l'errore del 10% si osserva che per il guscio

INPUT	0.00	0.0016	0.0769	0.1042	0.0632	0.0807	0.2405	0.2783
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	1.3197E-05	1.4398E-05	1.6408E-07	1.3009E-08
OUTPUT 2	1.0000	1.0000	1.7479E-05	2.8873E-07	1.0000	0.9999	1.2979E-06	8.2556E-09
OUTPUT 3	1.0000	5.6499E-05	1.0000	5.6916E-06	1.0000	4.0887E-07	1.0000	2.3964E-07
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.8:** Risultati per guscio  $h/a = 0.2$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati. Problema risolto con rete multipla

INPUT	0.00	0.0165	0.0891	0.1572	0.0328	0.0788	0.2510	0.3553
OUTPUT 1	1.0000	0.9963	0.9999	1.0000	6.6268E-06	3.1976E-05	1.9000E-05	2.3250E-04
OUTPUT 2	1.0000	1.0000	1.3555E-05	5.8729E-06	1.0000	0.9999	3.8026E-06	1.4496E-06
OUTPUT 3	0.7801	0.4116	0.4537	0.1151	0.6540	0.2869	0.1312	6.0842E-05
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.9:** Risultati per guscio  $h/a = 0.4$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati. Problema risolto con rete multipla

con spessore  $h/a = 0.2$  corrisponde al modello strutturale 011, mentre per il guscio con spessore  $h/a = 0.4$  corrisponde al modello 010.

Quanto fatto finora, utilizzando la media dei primi dieci modi di vibrare, è possibile farlo andando ad utilizzare l'errore percentuale solo sul primo modo di vibrare. Quello che ci si aspetta è che il comportamento della rete non muti radicalmente dai casi precedenti, ma per quanto visto nelle analisi precedenti, ci potrebbe essere un errore di non univocità per via della troppa vicinanza degli errori prodotti dai singoli modelli strutturali ridotti.

La problematica principale che si riscontra in questo tipo di applicazione è proprio il range di errori tollerati dal singolo guscio. Analizzando le varie tabelle si nota come, da un certo valore in su, la rete neurale offre come output sempre lo stesso modello strutturale. Questo è inevitabile in quanto se si supera l'errore massimo di training, per la rete non esiste un diverso modello strutturale oltre a quello che produce il suo errore massimo. Fatte tali premesse, questa applicazione risulta utile se si conosce un range di errori di massima prodotto da un guscio specifico. Inoltre, dal punto di vista ingegneristico, ha senso parlare di errori contenuti (entro il 20%) ed è per questo che nelle tabelle 10.20 10.21 e 10.22 non vi sono errori che superano lo 0.2 ovvero, appunto, il 20% di errore sulla media dei primi dieci modi di vibrare.

A questo punto si va a risolvere lo stesso problema utilizzando il metodo di raggruppamento dei modelli strutturali ridotti visto nella sezione 10.1.1. Per questo caso si andranno ad utilizzare solo errori in input che abbiano senso dal punto di vista ingegneristico e si valuterà ancora una volta la capacità della rete neurale di fornire un output che sia in grado di soddisfare la richiesta della percentuale d'errore. Il significato delle righe delle tabelle è il medesimo visto in precedenza. I risultati di quest'ultima analisi sono riportati nelle tabelle 10.24, 10.25, 10.26 e 10.27. Anche in questo caso i risultati sono molto buoni per alcuni gusci, mentre per altri sono meno distinguibili. Il guscio con spessore  $h/a = 0.2$  (tabella 10.26) presenta tutti i modelli strutturali molto ben distinguibili gli uni dagli altri. In questo caso specifico gli input 12%, 15% e 20% presentano lo stesso modello strutturale in output in quanto il massimo valore di errore utilizzato per il training è 11.60%. Se si osserva, per esempio, l'input 4% si nota che il modello strutturale in output è quello codificato con 101 (ovvero il modello FSDT a cui vengono aggiunti i termini in  $x$  e  $z$ ). Tale modello strutturale produce un errore pari al 4.54% e quindi molto vicino al ciò che è stato richiesto. Naturalmente, per quei valori percentuali di errore che sono lontani dai valori utilizzati nel training, i modelli strutturali previsti non sono univoci in quanto derivano dal conflitto di diversi possibili output.

INPUT	0.00	0.0039	0.3406	0.4541	0.5273	0.6336	0.8380	0.9578
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	2.9683E-06	8.0695E-06	1.5649E-05	5.7859E-06
OUTPUT 2	1.0000	1.0000	2.6491E-05	0.0670	0.9865	0.9999	1.1086E-05	1.9320E-05
OUTPUT 3	0.7282	0.6450	0.3187	0.1362	0.8188	0.0448	0.9377	0.4113
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.10:** Risultati per guscio  $h/a = 0.01$ , usando come input la media degli errori dei primi dieci modi di vibrare e codificati. Problema risolto con rete multipla

TARGET	$u_{x2}$	$u_{y2}$	$u_{z2}$	$u_{x3}$	$u_{y3}$	$u_{z3}$	$u_{x4}$	$u_{y4}$	$u_{z4}$	$u_{x5}$	$u_{y5}$	$u_{z5}$
000	▲	▲	△	△	△	△	△	△	△	△	△	△
001	▲	▲	▲	△	△	▲	△	△	▲	△	△	▲
010	▲	▲	△	△	▲	△	△	▲	△	△	▲	△
100	▲	▲	△	▲	△	△	▲	△	△	▲	△	△
011	▲	▲	▲	△	△	▲	△	▲	▲	△	▲	▲
101	▲	▲	▲	▲	△	▲	▲	△	▲	▲	▲	▲
110	▲	▲	△	▲	▲	△	▲	▲	△	▲	▲	△
111	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲

**Tabella 10.11:** Tabella target con base FSDT

## 10.2 Multi input

### 10.2.1 Errore sui modi di vibrare

In questa sezione si vuole provare a fornire alla rete neurale un serie di input formati da più errori sui modi di vibrare. Fino a questo momento, infatti, i due codici che si sono analizzati lavoravano con una stringa di dati in input che si riferivano o all'errore percentuale sul primo modo di vibrare o alla media degli errori dei primi dieci modi di vibrare(codificati o non). Per poter ampliare il range di utilizzo della rete neurale si prova a fornire come input non più un singolo errore, ma una serie di errori. Questi possono essere, ad esempio, gli errori percentuali di una numero arbitrario di modi di vibrare. Quello che si cerca di ottenere è quindi una serie di modelli strutturali che si riferiscono alla serie di errori forniti come input. Naturalmente la serie di modelli strutturali (quindi di target) è sempre la stessa e si ripete per tutti gli errori forniti. Quello che dovrà essere in grado di fare la rete neurale è proprio associare, ad esempio, ai primi due modi di vibrare ognuna delle combinazioni dei target riportati in tabella 10.1.

La rete neurale utilizzata è quella commentata nella sezione 10.1 con una struttura a sottolivelli in cui ognuno di questi è in grado di prevedere quale termine è acceso o spento. Le analisi si effettueranno sugli stessi gusci visti nella sezione precedente. 10.28, 10.29 e 10.30 sono riportati i risultati delle analisi, dove INPUT 1 si riferisce al primo modo di vibrare e INPUT 2 al secondo (entrambi i modi sono forniti alla rete codificati in quanto si è riscontrato che generalmente vi è una maggiore distinzione degli input). Analizzando i risultati si nota come vi sia un leggero peggioramento nella qualità dei modelli strutturali ottenuti. Naturalmente questo tipo di comportamento da parte della rete neurale era da attendersi poiché aumentando gli input in ingresso aumenta la difficoltà da parte della rete neurale di trovare una soluzione soddisfacente. Nella tabella 10.28 si nota come non vi sia una chiara distinzione tra i modelli 111 e 110; questo tipo di comportamento è sicuramente imputabile alla piccola differenza che esiste tra i due modelli strutturali in termini di errore percentuale fornito in input. infatti, nonostante il modello completo abbia un errore decisamente più piccolo rispetto a quello dove manca il termine in z, per la macchina appaiono entrambi come numeri molto prossimi allo zero. Il modello ridotto rispetto a quello completo ha, seppur comunque lontano dal risultato ottimale, un numero identificativo del modello più vicino allo zero rispetto al modello completo. Lo stesso comportamento si riscontra

INPUT	0.00	9.7840E-04	0.0159	0.0167	0.0153	0.0163	0.0312	0.0323
OUTPUT 1	1.0000	1.0000	4.4174E-05	1.2176E-06	1.0000	1.0000	3.9388E-11	2.8321E-11
OUTPUT 2	0.9999	1.0603E-05	0.9998	3.8930E-05	1.0000	4.8332E-04	1.0000	1.1124E-05
OUTPUT 3	1.0000	1.0000	1.0000	1.0000	2.0964E-05	1.7026E-05	6.1817E-10	4.0133E-10
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.12:** Risultati per guscio  $h/a = 0.01$ , usando come input la media degli errori dei primi dieci modi di vibrare con base FSDT. Problema risolto con rete multipla

INPUT	0.00	0.0300	0.0976	0.1256	0.0065	0.0358	0.1052	0.1329
OUTPUT 1	1.0000	1.0000	1.5231E-07	2.2549E-08	1.0000	1.0000	5.2347E-08	5.4034E-08
OUTPUT 2	1.0000	2.9852E-05	1.0000	6.1607E-07	1.0000	2.1770E-05	0.9999	7.5917E-07
OUTPUT 3	0.9999	1.0000	1.0000	0.9999	0.9995	2.1730E-05	2.4811E-05	1.7019E-05
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.13:** Risultati per guscio  $h/a = 0.1$ , usando come input la media degli errori dei primi dieci modi di vibrare con base FSDT. Problema risolto con rete multipla

nei casi riportati nelle tabelle 10.29 e 10.30. Fatte queste considerazioni è comunque evidente che la rete neurale ha lavorato discretamente bene, andando a cogliere quasi in maniera univoca la maggior parte dei modelli strutturali.

Quello che si vuole provare a fare e migliorare i risultati ottenuti da questa rete fino a questo punto. Poiché questo secondo problema è stato risolto con la stessa rete utilizzata nella sezione 10.1, ma per quanto detto il problema risulta essere più complesso, una soluzione naturale per incrementare la precisione dei risultati è quella di andare a potenziare la rete neurale. Per fare questo si va ad aggiungere ad ognuna delle sotto-reti del codice uno strato interno in modo da aumentarne la potenza di calcolo (fino a questo momento si avevano 3 strati interni da 35 neuroni l'uno, ora si passa a 4 strati interni da 25 neuroni). I risultati ottenuti sono riportati nelle tabelle 10.31, 10.32 e 10.33. Analizzando la tabella 10.31 e andando a confrontarla con la tabella 10.28 si può notare un netto miglioramento nella capacità della rete neurale di prevedere i modelli strutturali ridotti. Infatti se si considerano i primi due modelli (111 e 110) si può chiaramente osservare come nel secondo caso, con la rete potenziata, questi siano nettamente distinti e ben riconoscibili, cosa che prima non era possibile fare. Allo stesso modo il modello 101, seppur anche prima fosse riconoscibile, in questo caso viene previsto in maniera esatta. Ciò che rimane ancora irrisolto è il modello 001 in quanto sia nel caso precedente che in questo caso non viene distinto dal modello 000; questo tipo di andamento è sicuramente dovuto al fatto che gli input di questi ultimi due modelli strutturali sono decisamente simili tra di loro nonostante la codifica data. Il secondo caso, quello riportato in tabella 10.32, non riporta sostanziali miglioramenti in quanto aveva già un elevato grado di precisione nei risultati, fatta eccezione per il modello 110 che continua ad essere problematico. Per quanto riguarda invece il terzo caso, riportato in tabella 10.33, si ha un netto miglioramento dei risultati previsti. Infatti, nel caso della rete precedente riportato in tabella 10.30 vi erano diversi modelli strutturali che non erano perfettamente previsti dal codice utilizzato. Con questa seconda soluzione invece si è andati a risolvere tutti i modelli ridotti, fatta eccezione per il modello 001 che viene ancora confuso con quello 000.

## 10.2.2 Input dinamico e statico

Così come nella sezione 9.2 si chiedeva alla rete neurale, dato in input il modello strutturale, la curva Best Theory Diagram sia con gli errori sui modi di vibrare che sullo spostamento, in questa sezione si vuole effettuare il problema inverso. Quello che si vuole quindi fare è, una volta allenata la rete neurale con in input l'errore sul modo di vibrare scelto e lo spostamento (a dato modello strutturale), ottenere i modelli strutturali che producono tali errori. Naturalmente, anche

INPUT	0.00	0.0454	0.0669	0.1102	0.0056	0.0487	0.0736	0.1160
OUTPUT 1	1.0000	1.0000	8.6858E-06	6.6986E-06	1.0000	0.9999	1.5182E-06	9.7008E-06
OUTPUT 2	1.0000	9.0694E-05	1.0000	3.3465E-05	0.9999	1.1750E-05	1.0000	3.3955E-05
OUTPUT 3	0.9431	0.9490	0.9287	0.3665	0.4465	0.9525	0.0677	0.2243
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.14:** Risultati per guscio  $h/a = 0.2$ , usando come input la media degli errori dei primi dieci modi di vibrare con base FSDT. Problema risolto con rete multipla

INPUT	0.00	0.0483	0.0297	0.0772	0.0439	0.0865	0.0729	0.1153
OUTPUT 1	1.0000	0.9976	2.4998E-05	1.6053E-05	1.0000	0.9999	8.5859E-05	2.4497E-05
OUTPUT 2	1.0000	8.3400E-05	1.0000	2.6922E-05	1.0000	8.5608E-04	1.0000	4.1158E-05
OUTPUT 3	1.0000	1.0000	1.0000	1.0000	7.8172E-05	5.1185E-05	1.0000	2.8973E-05
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.15:** Risultati per guscio  $h/a = 0.4$ , usando come input la media degli errori dei primi dieci modi di vibrare con base FSDT. Problema risolto con rete multipla

in questo caso, il problema della previsione del modello strutturale viene semplificato utilizzando solo i modelli riportati in tabella 10.1. Come visto in precedenza il problema diretto dava buoni risultati, mentre in questo caso ci si possono attendere delle difficoltà di convergenza da parte del codice data la diversa natura del dato in input.

Svolgendo le simulazioni si è riscontrato che la rete neurale che lavora sui termini in  $x$  raggiungeva velocemente la convergenza ed infatti necessita del minor numero di strati interni (soltanto 4). Le altre due sotto-reti, invece, hanno una difficoltà nel raggiungimento della convergenza e di un risultato accettabile superiore e, per questo motivo, necessitano di un numero maggiore di strati interni e neuroni. Nello specifico, la sotto-rete che si occupa di prevedere i termini in  $y$  presente 5 strati interni con lo stesso numero di neuroni della sotto-rete  $x$ ; l'ultimo termine, quello in  $z$ , viene previsto da una sotto-rete con 6 strati interni con un aumento di 10 neuroni su ogni strato. Anche in questo modo il termine in  $z$  è quello peggio risolto (così come accadeva nei problemi precedenti), ma comunque soddisfacente. Una precisazione va fatta sui dati in input che, anche in questo caso, sono codificati come nelle sezioni precedenti poiché anche l'analisi statica presenta risultati sugli errori molto vicini tra di loro.

Nelle tabelle 10.34, 10.35, 10.36 e 10.37 sono riportati i risultati per dei gusci di spessore  $h/a = 0.01, 0.1, 0.15e0.2$ . Una precisazione va fatta sul significato delle varie tabelle: in questo caso INPUT 1 e INPUT 2 si riferiscono rispettivamente all'errore sul primo modo di vibrare dell'analisi dinamica e l'errore sullo spostamento nell'analisi statica. Quello che si osserva è un leggero peggioramento nei risultati rispetto ai casi analizzati precedentemente, in quanto i termini accesi non sono sempre previsti con 1.0000, ma con numeri che possono anche discostarsi di un 30% dal valore di riferimento. Globalmente, esclusi alcuni casi isolati, i modelli strutturali sono ben distinguibili nonostante le leggere imprecisioni. Un'osservazione va fatta sugli input: quello che si nota è che l'errore sul primo modo di vibrare e sullo spostamento hanno valori molto prossimi o, in alcuni casi, identici tra di loro. Questo tipo di valori in input può essere causa del problema su alcuni modelli strutturali che non sono perfettamente previsti dalla rete neurale. A tal proposito si vanno a ripetere le analisi imponendo che l'input riguardante l'analisi dinamica sia dato dalla media dei primi 10 modi di vibrare in modo da, in linea teorica, discostarsi il più possibile dall'input derivante dall'analisi statica.

Nelle tabelle 10.38, 10.39, 10.40 e 10.41 sono riportati i risultati di questo secondo tipo di analisi. In questo caso l'INPUT 1 si riferisce alla media degli errori dei primi 10 modi di vibrare, mentre il secondo input è inalterato rispetto alle analisi fatte precedentemente ed è l'errore sullo spostamento al centro del guscio. Quello che si può osservare analizzando le tabelle è che negli spessori dei gusci più piccoli vi è un miglioramento minimo nei vari modelli strutturali. Infatti, il guscio

INPUT	6.0028E-14	1.6460E-05	5.2662E-04	5.4307E-04	0.0028	0.0028	0.0034	0.0034
OUTPUT 1	1.0000	1.0000	1.4228E-05	1.3861E-05	1.0000	0.9999	6.6609E-05	5.3516E-05
OUTPUT 2	0.3330	0.1948	0.1382	0.1268	0.5873	0.5072	0.5786	0.5612
OUTPUT 3	1.0000	1.0000	1.0000	1.0000	3.2966E-05	3.2713E-05	2.5128E-05	2.5260E-05
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.16:** Risultati per guscio  $h/a = 0.01$ , usando come input l'errore percentuale sul primo modo di vibrare con base FSDT. Problema risolto con rete multipla

INPUT	2.0519E-15	9.6509E-04	0.1127	0.1135	0.0088	0.0099	0.1220	0.1229
OUTPUT 1	1.0000	1.0000	3.2146E-05	2.9903E-05	1.0000	1.0000	3.1594E-05	3.2578E-05
OUTPUT 2	0.1013	0.1246	0.9890	0.9868	0.6826	0.3445	0.9242	0.2109
OUTPUT 3	1.0000	1.0000	1.0000	1.0000	3.6774E-05	3.6966E-05	4.2833E-05	1.1533E-05
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.17:** Risultati per guscio  $h/a = 0.1$ , usando come input l'errore percentuale sul primo modo di vibrare con base FSDT. Problema risolto con rete multipla

con spessore  $h/a = 0.01$  ha i modelli strutturali ridotti previsti allo stesso modo rispetto al caso precedente, con le stesse incertezze già riscontrate. Andando ad aumentare lo spessore si nota invece un netto miglioramento della rete neurale nel prevedere, dati gli input, i modelli strutturali associati. Per analizzare questo risultato si va a confrontare la tabella 10.37 con la tabella 10.41. In questi due casi si possono andare ad analizzare tutti i modelli strutturali che sono riportati. Si noter  infatti come nell'ultimo caso questi siano perfettamente previsti dalla rete neurale e, in special modo, il termine in  $z$  sia sempre ben distinguibile da quando "acceso" a quando   "spento". Questo netto miglioramento pu  essere dovuto al fatto che, come si aveva ipotizzato, i due input hanno una differenza che   maggiore nel caso in cui si usa la media degli errori rispetto al caso in cui si usa semplicemente l'errore sul primo modo di vibrare.

INPUT	2.4202E-15	0.0048	0.1722	0.1755	0.0067	0.0118	0.1804	0.1841
OUTPUT 1	1.0000	1.0000	4.5014E-05	2.5548E-05	1.0000	1.0000	9.5049E-06	1.2452E-05
OUTPUT 2	0.4232	0.4410	0.9629	0.9765	0.4366	0.2728	0.7899	0.1474
OUTPUT 3	1.0000	1.0000	1.0000	1.0000	6.7929E-06	1.0769E-06	6.9107E-07	3.7542E-07
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.18:** Risultati per guscio  $h/a = 0.2$ , usando come input l'errore percentuale sul primo modo di vibrare con base FSDT. Problema risolto con rete multipla

INPUT	2.2826E-15	0.0170	0.1427	0.1556	0.0045	0.0216	0.1493	0.1633
OUTPUT 1	1.0000	1.0000	3.8417E-07	1.6600E-07	1.0000	1.0000	2.1419E-07	1.4152E-07
OUTPUT 2	1.0000	4.5502E-05	1.0000	2.9030E-06	1.0000	4.2073E-05	1.0000	2.2143E-06
OUTPUT 3	0.7559	0.0686	0.9635	0.7005	0.7426	0.0706	0.6365	0.0582
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.19:** Risultati per guscio  $h/a = 0.4$ , usando come input l'errore percentuale sul primo modo di vibrare con base FSDT. Problema risolto con rete multipla

INPUT	0.00	0.05	0.10	0.15	0.20			
OUTPUT 1	1.0000	0.9998	0.3079	0.0704	3.1590E-05			
OUTPUT 2	1.0000	1.0000	0.9999	1.0000	1.0000			
OUTPUT 3	0.9924	1.3558E-04	0.6132	0.2500	1.0000			
INPUT TRAINING	0.0000	0.0072	0.3143	0.3160	0.2125	0.2177	0.4194	0.4207
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.20:** Risultati per guscio  $h/a = 0.1$ , usando come input errori arbitrari. Confronto con gli input usati per il training ed i relativi target.

INPUT	0.00	0.05	0.10	0.15	0.20			
OUTPUT 1	1.0000	0.9886	1.5890E-06	0.8597	0.9999			
OUTPUT 2	1.0000	1.0000	1.0000	1.6270E-05	1.3428E-05			
OUTPUT 3	0.9235	0.2116	0.6699	0.8846	0.6734			
INPUT TRAINING	0.0000	0.0065	0.2053	0.2085	0.1012	0.1076	0.2749	0.2783
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.21:** Risultati per guscio  $h/a = 0.2$ , usando come input errori arbitrari. Confronto con gli input usati per il training ed i relativi target.

INPUT	0.00	0.01	0.05	0.10	0.15	0.20		
OUTPUT 1	1.0000	0.9999	4.0742E-05	9.9054E-07	0.1471	0.9998		
OUTPUT 2	1.0000	1.0000	1.0000	0.3996	3.9172E-06	2.1282E-09		
OUTPUT 3	1.0000	5.7536E-06	0.0133	1.5274E-05	0.9556	0.9986		
INPUT TRAINING	0.0000	0.0090	0.2207	0.2253	0.0734	0.0820	0.2763	0.2810
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.22:** Risultati per guscio  $h/a = 0.25$ , usando come input errori arbitrari. Confronto con gli input usati per il training ed i relativi target.

INPUT	0.90	0.05	0.10	0.25	0.35	0.20	0.01	0.00
OUTPUT 1	5.3920E-04	3.0429E-05	1.0170E-04	0.9999	7.2967E-04	0.9716	0.9999	1.0000
OUTPUT 2	8.3710E-05	0.9999	1.0000	3.2142E-05	5.1025E-05	0.0089	0.9999	0.9999
OUTPUT 3	2.1985E-04	1.0000	1.2617E-05	1.0000	2.2749E-05	0.9071	1.0000	0.9999
INPUT TRAINING	0.0000	0.0659	0.2377	0.3143	0.0525	0.1051	0.2869	0.3553
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.23:** Risultati per guscio  $h/a = 0.4$ , usando come input errori arbitrari. Confronto con gli input usati per il training ed i relativi target.

INPUT	0.00	0.0100	0.0200	0.0300	0.0500	0.1000	0.2000	
OUTPUT 1	1.0000	1.0000	2.0857E-10	1.0631E-10	1.2224E-10	1.2224E-10	1.2224E-10	
OUTPUT 2	1.0000	0.3160	3.4245E-05	0.9999	1.2484E-06	1.2484E-06	1.2484E-06	
OUTPUT 3	0.9597	0.6310	0.4205	0.0918	0.0467	0.0467	0.0467	
INPUT TRAINING	0.00	9.7840E-04	0.0159	0.0167	0.0153	0.0163	0.0312	0.0323
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.24:** Risultati per guscio  $h/a = 0.01$ , usando come input errori arbitrari, base dei modelli FSDT. Confronto con gli input usati per il training ed i relativi target.

INPUT	0.00	0.0200	0.0400	0.0800	0.1000	0.1200	0.1500	0.2000
OUTPUT 1	1.0000	1.0000	0.9999	1.0762E-05	2.1756E-07	4.8466E-07	5.0041E-06	5.0041E-06
OUTPUT 2	1.0000	0.8128	3.1011E-05	0.9999	1.0000	0.0020	5.4675E-05	5.4675E-05
OUTPUT 3	0.6855	0.6834	0.1348	0.3670	0.3112	0.5607	0.3385	0.3385
INPUT TRAINING	0.00	0.0300	0.0976	0.1256	0.0065	0.0358	0.1052	0.1329
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.25:** Risultati per guscio  $h/a = 0.1$ , usando come input errori arbitrari, base dei modelli FSDT. Confronto con gli input usati per il training ed i relativi target.

INPUT	0.00	0.0200	0.0400	0.0800	0.1000	0.1200	0.1500	0.2000
OUTPUT 1	1.0000	1.0000	1.0000	4.1076E-05	6.7321E-05	8.3939E-06	8.3971E-06	8.3971E-06
OUTPUT 2	0.9999	0.8142	3.5132E-04	0.9666	0.0039	5.5794E-05	5.5817E-05	5.5817E-05
OUTPUT 3	0.9999	0.9999	1.0000	0.0026	0.4137	8.8376E-06	8.8282E-06	8.8282E-06
INPUT TRAINING	0.00	0.0454	0.0669	0.1102	0.0056	0.0487	0.0736	0.1160
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.26:** Risultati per guscio  $h/a = 0.2$ , usando come input errori arbitrari, base dei modelli FSDT. Confronto con gli input usati per il training ed i relativi target.

INPUT	0.00	0.0200	0.0400	0.0800	0.1000	0.1200	0.1500	0.2000
OUTPUT 1	0.9999	2.2489E-04	0.9983	2.6722E-04	0.4625	1.3121E-05	1.3127E-05	1.3127E-05
OUTPUT 2	1.0000	1.0000	1.0000	2.2846E-05	5.7225E-04	0.0043	0.0043	0.0043
OUTPUT 3	1.0000	0.5550	1.7958E-05	0.8707	1.5286E-05	9.7462E-04	9.7426E-04	9.7426E-04
INPUT TRAINING	0.00	0.0483	0.0297	0.0772	0.0439	0.0865	0.0729	0.1153
TARGET	111	101	011	001	110	100	010	000

**Tabella 10.27:** Risultati per guscio  $h/a = 0.4$ , usando come input errori arbitrari, base dei modelli FSDT. Confronto con gli input usati per il training ed i relativi target.

INPUT 1	3.0253E-16	0.0020	0.1703	0.2273	0.2410	0.2911	0.5268	0.6020
INPUT 2	7.1442E-17	0.0025	0.1390	0.1864	0.0440	0.0587	0.3576	0.4102
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	4.8226E-06	6.5223E-06	1.0246E-05	1.3107E-05
OUTPUT 2	1.0000	1.0000	4.8754E-05	3.9996E-05	1.0000	1.0000	1.2572E-05	1.6665E-05
OUTPUT 3	0.2228	0.2201	0.7978	0.0920	0.9520	0.2681	0.0130	6.0627E-04
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.28:** Risultati per guscio  $h/a = 0.2$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla

INPUT 1	7.5036E-15	9.2128E-04	0.3630	0.4841	0.6099	0.7318	0.8628	0.9860
INPUT 2	8.9862E-15	0.0045	0.3475	0.4634	0.5751	0.6902	0.8654	0.9890
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	1.7459E-05	5.8581E-06	2.2233E-05	3.1092E-06
OUTPUT 2	1.0000	1.0000	6.0000E-05	1.0657E-04	1.0000	1.0000	3.5518E-05	1.8414E-05
OUTPUT 3	0.9999	0.9999	1.0000	2.8506E-05	0.9999	5.1396E-05	0.9999	5.0727E-05
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.29:** Risultati per guscio  $h/a = 0.01$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla

INPUT 1	2.8532E-16	0.0015	0.1014	0.1359	0.1221	0.1500	0.3445	0.3937
INPUT 2	1.5879E-16	0.0051	0.0644	0.0917	0.0229	0.0387	0.1798	0.2135
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	5.9891E-05	4.2329E-05	1.1684E-05	1.0984E-05
OUTPUT 2	0.9999	0.9999	5.4549E-05	5.1346E-04	1.0000	0.9999	1.9959E-05	1.4404E-05
OUTPUT 3	0.6328	0.5323	0.3268	0.2646	0.4582	0.0182	0.9602	0.0627
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.30:** Risultati per guscio  $h/a = 0.4$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla

INPUT 1	3.0253E-16	0.0020	0.1703	0.2273	0.2410	0.2911	0.5268	0.6020
INPUT 2	7.1442E-17	0.0025	0.1390	0.1864	0.0440	0.0587	0.3576	0.4102
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	3.6452E-05	2.4398E-05	3.4945E-05	2.9959E-05
OUTPUT 2	1.0000	1.0000	4.6102E-05	3.5714E-05	1.0000	1.0000	1.7855E-05	2.6363E-05
OUTPUT 3	1.0000	1.4262E-05	1.0000	8.7037E-11	1.0000	1.7499E-08	3.5295E-10	3.7408E-10
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.31:** Risultati per guscio  $h/a = 0.2$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla migliorata

INPUT 1	7.5036E-15	9.2128E-04	0.3630	0.4841	0.6099	0.7318	0.8628	0.9860
INPUT 2	8.9862E-15	0.0045	0.3475	0.4634	0.5751	0.6902	0.8654	0.9890
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	4.1314E-05	3.1507E-05	2.5141E-05	1.3218E-05
OUTPUT 2	1.0000	1.0000	7.0521E-05	1.0228E-05	1.0000	1.0000	4.6756E-05	5.8086E-05
OUTPUT 3	1.0000	1.0000	1.0000	1.6578E-05	1.0000	5.7436E-06	1.0000	2.8029E-06
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.32:** Risultati per guscio  $h/a = 0.01$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla migliorata

INPUT 1	2.8532E-16	0.0015	0.1014	0.1359	0.1221	0.1500	0.3445	0.3937
INPUT 2	1.5879E-16	0.0051	0.0644	0.0917	0.0229	0.0387	0.1798	0.2135
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	4.4061E-05	3.0296E-05	2.5952E-05	8.9534E-06
OUTPUT 2	1.0000	1.0000	3.3124E-06	1.8117E-06	1.0000	1.0000	1.1614E-06	1.4457E-05
OUTPUT 3	1.0000	4.7789E-05	1.0000	1.0988E-09	1.0000	2.9248E-08	6.6457E-07	1.2713E-05
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.33:** Risultati per guscio  $h/a = 0.4$ , usando come input i primi due modi di vibrare codificati. Problema risolto con rete multipla migliorata

INPUT 1	7.5036E-15	9.2128E-04	0.3630	0.4841	0.6099	0.7318	0.8628	0.9860
INPUT 2	1.4262E-14	9.0804E-04	0.3630	0.4841	0.6099	0.7318	0.8628	0.9860
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	1.9764E-05	1.2792E-05	1.6811E-05	2.3011E-05
OUTPUT 2	1.0000	1.0001	0.0050	0.0010	0.9089	0.9934	0.0011	1.3302E-04
OUTPUT 3	0.6068	0.6068	0.4731	0.4133	0.4557	0.2212	0.4505	0.5000
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.34:** Risultati per guscio  $h/a = 0.01$ , usando come input gli errori su analisi statica e dinamica

INPUT 1	2.5649E-16	0.0024	0.2443	0.3258	0.4070	0.4888	0.6926	0.7916
INPUT 2	1.8783E-15	0.0014	0.2434	0.3240	0.4064	0.4873	0.6909	0.7890
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	5.0411E-05	1.8645E-05	1.7766E-05	1.8139E-05
OUTPUT 2	1.0000	0.9983	1.1963E-13	-7.9714E-06	1.0000	1.0000	-0.0113	-3.7748E-15
OUTPUT 3	0.9998	0.9998	1.0000	5.4302E-05	1.0000	4.8861E-05	1.0000	2.1715E-05
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.35:** Risultati per guscio  $h/a = 0.1$ , usando come input gli errori su analisi statica e dinamica

INPUT 1	9.1433E-17	0.0021	0.2000	0.2669	0.3070	0.3695	0.5987	0.6843
INPUT 2	3.5987E-16	8.5950E-04	0.1961	0.2611	0.3025	0.3644	0.5881	0.6757
OUTPUT 1	1.0000	1.0000	1.0000	0.9999	2.7869E-05	8.1289E-06	7.0023E-06	6.8316E-06
OUTPUT 2	0.9949	0.9907	0.0153	0.2577	0.7627	0.9328	6.2865E-04	0.0222
OUTPUT 3	0.6535	0.2135	0.9921	0.9990	0.9922	0.0464	0.9810	2.2504E-04
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.36:** Risultati per guscio  $h/a = 0.15$ , usando come input gli errori su analisi statica e dinamica

INPUT 1	3.0253E-16	0.0020	0.1703	0.2273	0.2410	0.2911	0.5268	0.6020
INPUT 2	6.4041E-16	0.0018	0.1663	0.2180	0.2398	0.2839	0.5182	0.5876
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	4.8888E-05	1.8045E-06	1.7044E-06	1.6488E-06
OUTPUT 2	1.0002	0.9997	-0.2194	0.0730	0.7680	0.9875	0.0163	0.0677
OUTPUT 3	0.4847	0.4827	0.5199	0.2346	0.2052	0.1862	0.5913	0.3911
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.37:** Risultati per guscio  $h/a = 0.2$ , usando come input gli errori su analisi statica e dinamica

INPUT 1	0.0000	0.0039	0.3406	0.4541	0.5273	0.6336	0.8380	0.9578
INPUT 2	1.4262E-14	9.0804E-04	0.3630	0.4841	0.6099	0.7318	0.8628	0.9860
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	5.0951E-05	4.0149E-06	5.1469E-06	6.7507E-06
OUTPUT 2	0.9995	1.0011	-0.0408	0.0063	0.9707	0.9813	0.0098	0.0088
OUTPUT 3	0.4993	0.4791	0.8972	0.4768	0.8795	0.9160	0.4820	0.0734
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.38:** Risultati per guscio  $h/a = 0.01$ , usando come input gli errori su analisi statica e dinamica (errore medio dei primi 10 modi di vibrare)

INPUT 1	0.0000	0.0018	0.1179	0.1580	0.1328	0.1633	0.3670	0.4207
INPUT 2	1.8783E-15	0.0014	0.2434	0.3240	0.4064	0.4873	0.6909	0.7890
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	4.8831E-05	2.3009E-05	4.2985E-05	1.1297E-05
OUTPUT 2	0.9985	0.9983	4.6482E-04	8.8839E-04	0.9996	0.9999	-8.2237E-05	0.0107
OUTPUT 3	0.5182	0.4930	0.8842	0.9521	0.9570	0.1205	0.9392	0.0664
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.39:** Risultati per guscio  $h/a = 0.1$ , usando come input gli errori su analisi statica e dinamica (errore medio dei primi 10 modi di vibrare)

INPUT 1	0.0000	0.0014	0.0873	0.1175	0.0877	0.1135	0.2792	0.3213
INPUT 2	3.5987E-16	8.5950E-04	0.1961	0.2611	0.3025	0.3644	0.5881	0.6757
OUTPUT 1	1.0000	1.0000	1.0000	0.9999	4.4773E-06	1.6969E-07	8.6483E-08	6.9560E-08
OUTPUT 2	0.9997	1.0000	1.2755E-10	2.2694E-10	1.0000	1.0499	-8.3680E-12	-1.7676E-12
OUTPUT 3	1.0000	1.0000	1.0000	8.2955E-05	0.9999	6.2258E-05	1.0000	2.7173E-05
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.40:** Risultati per guscio  $h/a = 0.15$ , usando come input gli errori su analisi statica e dinamica (errore medio dei primi 10 modi di vibrare)

INPUT 1	0.0000	0.0016	0.0770	0.1043	0.0632	0.0807	0.2406	0.2783
INPUT 2	6.4041E-16	0.0018	0.1663	0.2180	0.2398	0.2839	0.5182	0.5876
OUTPUT 1	1.0000	1.0000	1.0000	1.0000	3.0480E-05	1.5508E-05	7.8275E-06	6.4354E-06
OUTPUT 2	1.0076	0.9961	0.0132	-0.1115	0.9948	1.0014	0.0054	0.0034
OUTPUT 3	1.0000	5.3982E-08	1.0000	4.2546E-10	1.0000	1.0403E-09	1.0000	3.9592E-04
TARGET	111	110	101	100	011	010	001	000

**Tabella 10.41:** Risultati per guscio  $h/a = 0.2$ , usando come input gli errori su analisi statica e dinamica (errore medio dei primi 10 modi di vibrare)

# Capitolo 11

## Conclusioni

Nel corso di questa tesi si è andati a studiare una possibile applicazione del Machine Learning (strettamente collegato alle reti neurali) in campo strutturale. Come visto nell'introduzione, l'universo Machine Learning è in continua espansione e i rami da esso toccati iniziano ad essere molti. La particolarità del lavoro svolto è stata quella di andare a coniugare un argomento innovativo qual è l'intelligenza artificiale con una teoria altrettanto innovativa che è quella della Best Theory Diagram. Grazie a questo connubio si è potuto sviluppare una prima rete neurale in grado di lavorare sui modelli strutturali ridotti e capire quali possano essere le loro potenziali applicazioni. Ciò che è stato svolto nella tesi viene di seguito sintetizzato in quattro punti fondamentali.

- Quello che è stato fatto inizialmente è stata una fase di training volto alla comprensione e familiarizzazione del codice per l'analisi agli elementi finiti del gruppo di ricerca appartenente al dipartimento di ingegneria aerospaziale MUL2. In questo contesto si è anche entrati a contatto con la teoria della Best Theory Diagram e dei concetti di modelli strutturali ridotti. Il passo successivo è stato andare a capire i principi base di una rete neurale tentando di ricavare delle curve BTM attraverso un codice scritto con il linguaggio MatLab. In questo modo ci si è addentrati maggiormente in quello che è il corpo della tesi: il Machine Learning applicato ai modelli strutturali;
- Lo scopo della tesi, a questo punto, è stato quello di andare a migliorare la rete neurale per risolvere problemi via via più complessi senza perdere di vista, specialmente all'inizio, la Best Theory Diagram. Infatti, uno dei primi problemi che si è affrontato con il Machine Learning è stato quello di andare ad analizzare l'influenza sull'errore medio dei primi dieci modi di vibrare del singolo modo. In questa maniera è risultato evidente come, nella maggior parte dei casi, il modello strutturale scelto per accomodare tutti i modi di vibrare non era quello che minimizzava il singolo errore, ma solamente quello globale. Naturalmente, tale aspetto era prevedibile, ma con questa analisi si è capito a fondo la sconvenienza in alcuni casi di utilizzare come errore di riferimento quello mediato tra i primi dieci modi di vibrare. Successivamente, sempre in ambito BTM si è andati ad aumentare il numero di dati in input per la determinazione della curva. Questo tipo di problema, sicuramente più complicato per la rete neurale, è stato utile per rendere più globale l'utilizzo di questa tecnologia. Infatti, andando ad aumentare le caratteristiche di un guscio (non più solamente i vari modelli strutturali ridotti) in input, la rete neurale è in grado di allenarsi su un numero di casistiche finite e, potenzialmente, poter prevedere un numero illimitato di curve per altrettanti gusci. Ciò che è risultato evidente è stato che, prendendo come caratteristica di riferimento lo spessore, la rete neurale produceva risultati molto buoni se lo spessore richiesto per il guscio analizzato era compreso tra quelli utilizzati per il training della rete neurale stessa. In questo

modo si è capito che era sufficiente aumentare l'intervallo di training per poter analizzare tutti i casi necessari;

- Successivamente, si è passati ad un ulteriore miglioramento della rete neurale per poter ottenere simultaneamente la doppia analisi: dinamica e statica. Anche questo miglioramento, unito a quello dell'aumento delle caratteristiche in input, permette alla tecnologia Machine Learning di essere sempre di più globale e di diminuire i tempi di calcolo. In questo caso, infatti, si va a risparmiare costo computazionale portato da una doppia analisi agli elementi finiti su migliaia di termini. È ovvio che in questo caso anche il tempo di training della rete neurale è aumentato poiché sono necessarie una serie di analisi FEM per poter allenare la rete; tali analisi comportano appunto un costo che si traduce in tempo di calcolo. Per quanto già detto, però, tale tempo di training viene sicuramente ammortizzato con la possibilità di andare a prevedere un numero illimitato di curve partendo sempre dallo stesso set di dati;
- Come ultima problematica ci si è posti quella di andare a svolgere il problema opposto, ovvero quello di ricavare il modello strutturale a partire da un errore in input. Tale problematica risulta essere più complicata rispetto a quelle analizzate fino a questo momento perché si passa da avere più di 12 input per determinare uno o al massimo due output all'opposto. Naturalmente, semplicemente invertire input con output non poteva dare nessun risultato accettabile poiché, oltre ad una rete troppo appesantita dai tanti target, un numero non è in grado di definirne univocamente 12. A questo si aggiunge che, le migliaia di errori che vengono prodotti dai modelli strutturali, sono a blocchi praticamente identici. Per questi motivi è impensabile ottenere un buon risultato senza modificare i presupposti. A tal proposito si è andati a semplificare il problema individuando due tipi di modelli strutturali che vertono sui termini in x, y e z. In questo modo, raggruppando tali termini a gruppi di tre è stato possibile ricavare due set da 8 modelli strutturali ridotti. Il secondo set si differenzia dal primo per avere, come base di partenza (banalmente il modello 000 ovvero tutti i termini spenti) il modello FSDT. Con questa semplificazione è stato possibile andare a costruire una rete neurale in grado di poter fare le previsioni sui modelli strutturali ridotti a partire da un errore in input. Successivamente si è ancora andati ad aumentare la difficoltà, testando la rete con degli errori di tipo arbitrario. In questo modo, partendo da una percentuale di errore desiderata, la rete neurale restituisce quel modello che è in grado di fornire l'errore più vicino a quello richiesto. Quest'ultima è un'applicazione molto interessante in ambito commerciale poiché, in base al problema che devo risolvere posso usare un modello più preciso e più costoso dal punto di vista computazionale oppure meno preciso, ma anche più veloce. In questo modo si va ad accomodare le varie esigenze rispetto al problema che si vuole andare a risolvere.

La tesi che è stata svolta ha portato un'iniziale impostazione del problema della scelta dei modelli strutturali con il Machine Learning. Sicuramente, sviluppando ulteriormente la rete neurale proposta si può andare a coprire una più vasta gamma di problematiche e maggiori casistiche legate a strutture diverse. In questo modo, migliorando sempre più ciò che è stato riportato in questo lavoro sarà possibile avvicinarsi alla possibilità di analizzare la totalità dei casi presenti ad oggi per le strutture. I possibili sviluppi futuri che possono essere effettuati a partire da questa tesi riguardano sicuramente il problema della determinazione di un modello strutturale a partire da un errore arbitrario. Come visto, il problema è stato impostato, ma in modo ancora semplificato. Infatti, allo stato attuale vi è la possibilità di scegliere tra 8 modelli strutturali a fronte delle svariate migliaia (comunque si è riscontrato che a blocchi di decine forniscono gli stessi errori). Il primo passo che si può fare per aumentare i modelli strutturali è quello di cambiare il modo di raggruppare e codificare i modelli strutturali, passando da un raggruppamento per direzione ad un raggruppamento per ordine. In questo modo si passa, per il modello  $N = 4$ , da 8 a 16 modelli

tra cui scegliere. Aumentando a mano a mano le possibili combinazioni si potrebbe arrivare, idealmente e con una rete molto complessa, ad ottenere  $2^{15}$  combinazioni anche nel problema inverso. Per quanto riguarda il problema analizzato nella prima parte della tesi, quello che è possibile migliorare è sicuramente il numero di caratteristiche che il programma è in grado sia di accettare in input che a prevedere in output. In questo modo, sicuramente, ci sarebbe un duplice miglioramento: sia l'abbattimento dei tempi per avere un'analisi, sia la più vasta gamma di analisi che sarebbe possibile effettuare con una rete neurale.



# Bibliografia

- [1] Rosenblatt e Frank. «The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain». In: *Psychological Review* 65 (gen. 1958), pp. 386–.
- [2] A. Berrais. «Artificial Neural Networks in Structural Engineering». In: *Journal of King Abdulaziz University : Engineering Sciences* 12 (gen. 1999). DOI: 10.4197/Eng.12-1.4.
- [3] E. Carrera. «The effects of shear deformation and curvature on buckling and vibrations of cross-ply laminated composite shells». In: *Journal of Sound and Vibration* 150.3 (1991), pp. 405–433.
- [4] E. Carrera et al. «Results on best theories for metallic and laminated shells including Layer-Wise models». In: *Composite Structures* 126 (2015), pp. 285–298.
- [5] P. Cicala. *Systematic approximation approach to linear shell theory*. Torino: Levrotto e Bella, 1965.
- [6] M. Cinefra. «Free-vibration analysis of laminated shells via refined MITC9 elements». In: *Mechanics of Advanced Materials and Structures* 23.9 (2016), pp. 937–947. DOI: 10.1080/15376494.2015.1121556.
- [7] M. Cinefra e S. Valvano. «A variable kinematic doubly-curved MITC9 shell element for the analysis of laminated composites». In: *Mechanics of Advanced Materials and Structures* 23.11 (2016), pp. 1312–1325.
- [8] M. Petrolo E. Zappino E. Carrera M. Cinefra. «Finite Element Analysis of Structures Through Unified Formulation». In: *Finite Element Analysis of Structures through Unified Formulation* (lug. 2014). DOI: 10.1002/9781118536643.
- [9] M. F. Elkordy, Kuo-Chun Chang e G. C. Lee. «Neural Networks Trained by Analytically Simulated Damage States». In: *Journal of Computing in Civil Engineering - J COMPUT CIVIL ENG* 7 (apr. 1993). DOI: 10.1061/(ASCE)0887-3801(1993)7:2(130).
- [10] A. K. Garg, R. K. Khare e T. Kant. «Higher-order Closed-form Solutions for Free Vibration of Laminated Composite and Sandwich Shells». In: *Journal of Sandwich Structures & Materials* 8.3 (2006), pp. 205–235. DOI: 10.1177/1099636206062569. eprint: <https://doi.org/10.1177/1099636206062569>. URL: <https://doi.org/10.1177/1099636206062569>.
- [11] A. L. Gol'denweizer. *Theory of thin elastic shells*. International Series of Monograph in Aeronautics and Astronautics. New York: Pergamon Press, 1961.
- [12] M.T. Hagan et al. *Neural Network Design*. Martin Hagan, 2014. ISBN: 9780971732117. URL: <https://books.google.it/books?id=4EW9oQEACAAJ>.
- [13] N N Huang. «Influence of shear correction factors in the higher-order shear deformation laminated shell theory». In: *International Journal of Solids and Structures* 31 (1994), pp. 1263–1277.
- [14] A. Khan, B. H. V. Topping e A. Bahreinineja. «Parallel training of neural networks for finite element mesh generation». In: (giu. 2019). DOI: 10.4203/ccp.16.5.1.

- [15] G. Kirchhoff. «Uber das gleichgewicht und die bewegung einer elastischen schiebe». In: *Journal fur reins und angewandte Mathematik* 40 (1850), pp. 51–88.
- [16] James L. Rogers. «Simulating Structural Analysis with Neural Network». In: *Journal of Computing in Civil Engineering - J COMPUT CIVIL ENG* 8 (apr. 1994). DOI: 10.1061/(ASCE)0887-3801(1994)8:2(252).
- [17] «Methods and guidelines for the choice of shell theories, Acta Mechanica, In Press.» In: (). A cura di E. Carrera M. Petrolo.
- [18] R. D. Mindlin. «Influence of rotatory inertia and shear in flexural motions of isotropic elastic plates». In: *Journal of Applied Mechanics* 18 (1951), pp. 1031–1036.
- [19] M. Petrolo e E. Carrera. «Best theory diagrams for multilayered structures via shell finite elements». In: *Advanced Modeling and Simulation in Engineering Sciences* 6.1 (2019), p. 4. ISSN: 2213-7467. DOI: 10.1186/s40323-019-0129-8. URL: <https://doi.org/10.1186/s40323-019-0129-8>.
- [20] J. N. Reddy. «Exact Solutions of Moderately Thick Laminated Shells». In: *Journal of Engineering Mechanics* 110.5 (1984), pp. 794–809. DOI: 10.1061/(ASCE)0733-9399(1984)110:5(794).
- [21] J.N. Reddy e C.F. Liu. «A higher-order shear deformation theory of laminated elastic shells». In: *International Journal of Engineering Science* 23.3 (1985), pp. 319 –330. DOI: [https://doi.org/10.1016/0020-7225\(85\)90051-5](https://doi.org/10.1016/0020-7225(85)90051-5).
- [22] E. Reissner. «The effect of transverse shear deformation on the bending of elastic plates». In: *Journal of Applied Mechanics* 12 (1945), pp. 69–76.
- [23] D. E. Rumelhart, James L. McClelland e CORPORATE PDP Research Group, cur. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2: Psychological and Biological Models*. Cambridge, MA, USA: MIT Press, 1986. ISBN: 0-262-13218-4.
- [24] W S. McCulloch e W A Pitts. «A Logical Calculus of Ideas Immanent in Nervous Activity» Bulletin of Mathematical Biophysics». In: (gen. 1994).
- [25] B. Widrow e Marcian E. H. *Adaptive Switching Circuits*. New York, 1960.