POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Meccanica

Tesi di Laurea Magistrale

Sviluppo di una metodologia per manutenzione predittiva real-time di un componente meccanico in ambito automotive



Anno Accademico 2022/2023 Sessione di laurea: Ottobre 2023

Relatore: Prof. Massimo Sorli Tutor aziendale: Ing. Giovanni Pesare Azienda: Capgemini Italia S.p.A.

> Candidato: Lorenzo Alunni

Sommario

Il seguente lavoro di tesi mira allo sviluppo di una metodologia per effettuare manutenzione predittiva in tempo reale di componenti comunemente impiegati in ambito automotive. Durante il funzionamento, gli elementi di un sistema meccanico sono sollecitati da carichi che differiscono per tipologia e intensità. Pertanto, in fase progettuale vengono opportunamente dimensionati affinché possano resistere agli sforzi attesi. In generale, i componenti di un assieme meccanico non possiedono una vita illimitata ed è opportuno eseguire delle ispezioni di controllo per verificare la corretta funzionalità delle varie parti e prevenire eventuali guasti. Le vetture odierne monitorano contemporaneamente lo stato di più gruppi meccanici, come ad esempio motore, impianto frenante, sospensioni e trasmissione. Infatti, le auto sono dotate di numerosi sensori per acquisire i dati provenienti da ciascun sistema, che una volta elaborati dai sistemi di controllo software embedded consentono di produrre una risposta adeguata alle richieste di potenza del conducente. Le diverse ECU (Electronic Control Unit) sono collegate ai trasmettitori e tra loro mediante il CAN-bus (Controller Area Network), ossia la rete informatica interna al veicolo (standard seriale per bus di campo). La crescente connettività dei dispositivi installati a bordo di un'automobile, le tecnologie IoV (Internet of Vehicles) e IoT (Internet of Things) rivoluzioneranno l'esperienza di guida. Evidentemente, l'idea di applicare tecniche di manutenzione predittiva ai componenti degli autoveicoli nasce dalla constatazione di una progressiva e inarrestabile digitalizzazione del settore automotive. In questo contesto, i dati raccolti da sensori IoT possono divenire gli input di modelli analitici o di algoritmi di intelligenza artificiale, che a partire dalle informazioni ricevute permettono di valutare lo stato di salute di un elemento meccanico. Di seguito si riassume brevemente il contenuto dell'elaborato di tesi. Nel capitolo iniziale si introduce il tema della manutenzione predittiva e si mostra la procedura seguita per la scelta del componente utilizzato come use-case per il progetto (la molla della sospensione di un veicolo di prova). Si prosegue descrivendo le caratteristiche del sensore laser impiegato per l'acquisizione dati. In aggiunta, si riporta uno schema dei cablaggi effettuati. In seguito, si presentano i modelli agli elementi finiti realizzati per simulare il componente staticamente e a fatica. Dopo aver elencato le principali tecniche di intelligenza artificiale, si illustra il procedimento adottato per costruire una rete neurale profonda capace di prevedere il danno subito dalla molla durante il funzionamento. Il modello surrogato viene addestrato attraverso un database ricavato dalla correlazione delle grandezze di input con quelle di output delle analisi agli elementi finiti. Infine, si riporta un esempio numerico in cui viene stimato il danneggiamento del componente a partire da uno spettro di carico irregolare acquisito mediante il trasmettitore laser. Si precisa che il disegno tridimensionale della sospensione è realizzato con NX Siemens, mentre i software utilizzati per le simulazioni agli elementi finiti sono Altair HyperMesh, Altair Optistruct e FEMFAT. D'altra parte, i codici per l'automatizzazione dei processi e per il training delle reti neurali sono scritti nel linguaggio di programmazione Python.

Abstract

The aim of this master thesis is the development of a new methodology to implement real-time predictive maintenance of widely used components in the automotive industry. During normal operating conditions, the elements of a mechanical system are stressed by various types of loads. Hence, they are appropriately designed to withstand the expected strain. Usually, the components of a mechanical assembly do not have an unlimited lifespan and therefore, control inspections should be carried out to verify the correct functionality of each part and prevent possible failures. Nowadays, technology in cars can supervise the conditions of several mechanical groups simultaneously, such as the engine, the braking system, the suspension and the gearbox. In fact, vehicles are equipped with numerous sensors to collect data from each system so that once they are processed by the appropriate embedded software control systems or ECUs (Electronic Control Units), a suitable response to the driver's power demands is given. The various ECUs and transmitters are connected to each other through the Controller Area Network (CAN) protocol, that is a broadcast digital serial communication bus which represents the vehicle's computer network. The driving experience will be deeply transformed by the increasing connectivity of cars' devices, Internet of Vehicles (IoV) technology and Internet of Things (IoT) technology. Accordingly, the idea of applying predictive maintenance techniques to vehicle components stems from the observation of a progressive and relentless digitalization in the automotive sector. In this context, the data collected by IoT sensors can be used as input parameters for analytical models or artificial intelligence algorithms which, based on the information received, are capable of assessing the state of health of a mechanical element. The content of this master thesis is briefly summarized below. In the first chapter, the subject of predictive maintenance is introduced and the procedure followed to select the use-case component of the project (the suspension spring of a car) is shown. Thereafter the characteristics of the laser sensor used for data acquisition are described. In addition, the electrical connection between the sensor's wires and the vehicle's battery is reported. In the following chapters, the construction of the finite element models is presented and the spring's internal stress is evaluated by means of linear static and fatigue analysis. After listing the main artificial intelligence techniques, the next chapter illustrates how to build a deep neural network capable of predicting the damage accumulated by the spring during operating conditions. The surrogate model is trained through a database obtained by using input and output quantities of the previous finite element method (FEM) simulations. Finally, a numerical example clarifies the procedure used to compute the component's damage starting from an irregular load spectrum acquired by means of a laser transmitter. The three-dimensional design of the suspension was realized using Siemens NX, while the software employed for the finite element analysis were Altair HyperMesh, Altair Optistruct and FEMFAT. Instead, the codes for process automation and neural networks training were written using Python as the programming language.

Indice

	Ele	nco delle figure 4
	Ele	nco delle tabelle 8
1	Int	roduzione al progetto di ricerca e sviluppo10
	1.1	Tipologie di manutenzione10
	1.2	La manutenzione predittiva in ambito automotive12
	1.3	Analisi dei danni subiti da alcuni componenti dell'autoveicolo 13
	1.4	Selezione dello use-case del progetto16
	1.5	Procedimento per la valutazione del danno in tempo reale19
2	. Tra	smettitori21
	2.1	Caratteristiche statiche degli strumenti di misura
	2.2	Scelta del sensore24
	2.3	Schema di cablaggio del sensore laser ILD1220-20029
3	Me	todo degli elementi finiti31
	3.1	Introduzione alle analisi agli elementi finiti
	3.2	Il problema elastico tridimensionale33
	3.2.	1 Equazioni differenziali di equilibrio
	3.2. 3.2.	2 Equazioni differenzian di congruenza
	3.3	Dalla meccanica del continuo agli elementi finiti
	3.3.	1 Il principio dei lavori virtuali
	3.3.	2 Il problema elastico in forma matriciale
	3.4	Convergenza nei metodi FEM
	3.4.	1 I pologie di elementi in HyperMesh44
4	. Mo	lla della sospensione anteriore: analisi FEM statica
	4.1	Modello geometrico46
	4.2	Proprietà del materiale del componente48
	4.3 4.3.	Modello agli elementi finiti in HyperMesh49 1 Costruzione mesh 2D e 3D51

4.3.2	Definizione dei carichi e dei vincoli	53
4.4 A	Analisi di convergenza del modello	55
4.4.1	Fase 1	55
4.4.2	Fase 2	60
4.5 N	Modello ottimizzato: definizione del carico	63
5 Moll	a della sospensione anteriore: analisi FEM a fatica	. 64
5.1 I	ntroduzione al fenomeno	64
5.2 C	Criteri di calcolo	65
5.3 N	Metodo stress-life	66
5.3.1	Effetto della tensione media	68
5.3.2	Fattori che influenzano la vita a fatica	69
5.4 C	Dbiettivi delle analisi e metodologia di calcolo	71
5.4.1	Calcolo a fatica con storie di carico irregolari	74
5.5 N	Modello agli elementi finiti in FEMFAT	77
5.5.1	Definizione dei parametri di input-output	77
5.5.2	Set-up delle simulazioni	79
5.5.3	Punto di lavoro nel diagramma di Haigh	83
5 5 <i>1</i>	Pisultati dollo simulazioni	0-
5.5.4	Risultati delle siliulazioni	05
6 Intel	ligenza artificiale	89
6 Intel 6.1 T	ligenza artificiale Secnologie alla base dell'IA	85 89 89
6 Intel 6.1 T 6.2 M	Iigenza artificiale Secnologie alla base dell'IA Machine Learning	85 89 89 90
6 Intel 6.1 T 6.2 M 6.3 I	Iigenza artificiale Secnologie alla base dell'IA Machine Learning ntroduzione alle reti neurali	85 89 89 90 92
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1	Iigenza artificiale Cecnologie alla base dell'IA Machine Learning ntroduzione alle reti neurali Struttura	89 89 90 92 92
6 Intel 6.1 T 6.2 M 6.3 F 6.3.1 6.3.2	Iigenza artificiale Cecnologie alla base dell'IA Machine Learning ntroduzione alle reti neurali Struttura Addestramento	89 89 90 92 92 92
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3	Iigenza artificiale Secnologie alla base dell'IA Machine Learning ntroduzione alle reti neurali Struttura Addestramento Parametri	85 89 90 92 92 92 94 96
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3 6	Iigenza artificiale Secnologie alla base dell'IA Machine Learning ntroduzione alle reti neurali Struttura Addestramento Parametri .3.3.1 Activation functions	89 89 90 92 92 92 94 96 97
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3 6 6 6	Iigenza artificiale Cecnologie alla base dell'IA Machine Learning ntroduzione alle reti neurali Struttura Addestramento Parametri .3.3.1 Activation functions	89 90 92 92 94 96 97 101
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3 6 6 6 6	Iigenza artificiale Cecnologie alla base dell'IA Machine Learning ntroduzione alle reti neurali Struttura Addestramento Parametri .3.3.1 Activation functions .3.3.2 Kernel e bias initializers .3.3.3 Tecniche di regolarizzazione	89 90 92 92 92 94 96 97 101 102
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3 6 6 6 6 6 6 6 6 6 6 6 6 6	Risultatione estimutation ligenza artificiale Cecnologie alla base dell'IA	89 90 92 92 92 94 96 97 101 102 103
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3 6 6 6 6 6 6 6 6 6 6 6 6 6	Risultal delle simulazioni ligenza artificiale Cecnologie alla base dell'IA. Machine Learning ntroduzione alle reti neurali Struttura Addestramento Parametri .3.3.1 Activation functions .3.3.2 Kernel e bias initializers .3.3.3 Tecniche di regolarizzazione .3.3.5 Algoritmi di ottimizzazione	89 90 92 92 92 94 96 97 101 102 103 108
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3 6 6 6 6 6 6 6 6 6 6 6 6 6	Risultatione sinulazioni ligenza artificiale Cecnologie alla base dell'IA. Machine Learning ntroduzione alle reti neurali Struttura Addestramento Parametri .3.3.1 Activation functions .3.3.2 Kernel e bias initializers .3.3.3 Tecniche di regolarizzazione .3.3.4 Funzioni di costo .3.3.5 Algoritmi di ottimizzazione .3.3.6	85 89 90 92 92 94 96 97 101 102 103 108 112
6 Intel 6.1 T 6.2 M 6.3 F 6.3.1 6.3.2 6.3.3 6 6 6 6 6 6 6 6 6 6 6 6 6	Risultation delle simulazioni ligenza artificiale Cecnologie alla base dell'IA Machine Learning ntroduzione alle reti neurali Struttura Addestramento. Parametri .3.3.1 Activation functions .3.3.2 Kernel e bias initializers .3.3.3 Tecniche di regolarizzazione .3.3.4 Funzioni di costo .3.3.5 Algoritmi di ottimizzazione .3.3.6 Learning rate .3.3.7 Batch size ed epochs	89 90 92 92 94 94 96 97 101 102 103 108 112 114
6 Intel 6.1 T 6.2 M 6.3 F 6.3.1 6.3.2 6.3.3 6 6 6 6 6 6 6 6 6 6 6 6 6	Risultal delle silidizioni ligenza artificiale Cecnologie alla base dell'IA. Machine Learning Introduzione alle reti neurali Struttura Addestramento Parametri .3.3.1 Activation functions .3.3.2 Kernel e bias initializers .3.3.3 Tecniche di regolarizzazione .3.3.4 Funzioni di costo .3.3.5 Algoritmi di ottimizzazione .3.3.6 Learning rate .3.3.7 Batch size ed epochs Modello di danneggiamento predittivo del componente	05 89 90 92 92 92 94 96 97 101 102 103 103 108 112 114 115
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3 6 6 6 6 6 6 6 6 6 6 6 6 6	Risultal dele sinulazioni ligenza artificiale Cecnologie alla base dell'IA. Machine Learning ntroduzione alle reti neurali Struttura. Addestramento. Parametri .3.3.1 Activation functions .3.3.2 Kernel e bias initializers .3.3.3 Tecniche di regolarizzazione .3.3.4 Funzioni di costo. .3.3.5 Algoritmi di ottimizzazione .3.3.7 Batch size ed epochs .3.3.7 Batch size ed epochs Madello di danneggiamento predittivo del componente Creazione del database e pre-processing dei dati	05 89 90 92 91 92 91 92 91
6 Intel 6.1 T 6.2 M 6.3 I 6.3.1 6.3.2 6.3.3 6 6 6 6 6 6 6 6 6 6 6 6 6	Risultal delle sinulazioni ligenza artificiale	85 89 90 92 92 92 92 94 97 101 102 103 108 112 114 115 116 118

7	Esempio di previsione del danneggiamento	
8	Conclusioni	134
Apj	pendice A: Raspberry Pi	
А	A.1 Che cos'è?	
А	A.2 General purpose input-output pins	
А	A.3 Sensore HC-SR04	140
	A.3.1 Principio di funzionamento	141
	A.3.2 Cablaggio	
	A.3.3 Codice Python per l'utilizzo	
Bib	bliografia	146
Rin	ngraziamenti	148

Elenco delle figure

Figura 1.1: Classificazione delle tecniche di manutenzione	10
Figura 1.2: Strumenti della manutenzione predittiva	11
Figura 1.3: Flusso dei dati per il monitoraggio dello stato di salute di un componente	12
Figura 1.4: Principali componenti di un gruppo ruota automobilistico	13
Figura 1.5: Principali componenti di un impianto frenante automobilistico	14
Figura 1.6: Principali componenti di una sospensione automobilistica	15
Figura 1.7: Procedimento adottato per stimare in tempo reale il danneggiamento del	
componente	. 20
Figura 2 1. Struttura di un trasmettitore	21
Figura 2.2: Caratteristica statica reale e ideale	·· _ ·
Figura 2.3: Risoluzione di uno strumento di misura	
Figura 2.4: Isteresi di uno strumento di misura	23
Figura 2.5: Ripetibilità e accuratezza di uno strumento di misura	23
Figura 2.6: Estensimetro a lamina metallica	25
Figura 2.7: Potenziometro lineare	25
Figura 2.8: LVDT	26
Figura 2.9: Sensore laser ILD1220-200	26
Figura 2.10: Modulo di interfaccia IF2001/USB	27
Figura 2.11: Interfaccia grafica del software "sensorTOOL"	27
Figura 2.12: Ingombri del sensore laser ILD1220-200	28
Figura 2.13: Installazione del sensore laser in vettura	29
Figura 2.14: Cablaggio del sensore laser ILD1220-200	30
Figura 3.1: Esempio di visualizzazione dei risultati di un'analisi FEM	31
Figura 3.2: Validazione di modelli FEM	32
Figura 3.3: Vettore tensione in P	33
Figura 3.4: Equilibrio dei vettori tensione	33
Figura 3.5: Campo elementare di tensione	34
Figura 3.6: Deformazione di un continuo	36
Figura 3.7: Metodo degli elementi finiti	38
Figura 3.8: Stato di tensione in un infinitesimo 2D	39
Figura 3.9: Spostamenti virtuali in un infinitesimo 2D	39
Figura 3.10: Generico punto P interno ad un elemento piano a 4 nodi	41
Figura 3.11: Proprietà delle funzioni di forma	43
Figura 3.12: Elementi 2D in HyperMesh	44
Figura 3.13: Elementi piani a 3 e 4 nodi	44
Figura 3.14: Elementi 3D in HyperMesh	45

1 guia 4.1 a. Modello Chil completo
Figura 4.1-b: Dettaglio molla sospensione46
Figura 4.2: Quote molla in configurazione indeformata47
Figura 4.3: Curva $\sigma-\varepsilon$ dell'acciaio 55 SiCr 6-348
Figura 4.4: Interfaccia grafica del software Altair HyperMesh49
Figura 4.5: Costruzione del modello FEM50
Figura 4.6: Creazione della mesh 2D in HyperMesh51
Figura 4.7: Creazione della mesh 3D in HyperMesh52
Figura 4.8: Dettaglio della mesh 3D della molla52
Figura 4.9: Elementi monodimensionali RBE2 e RBE353
Figura 4.10: Nodi caricati e vincolati appartenenti alla superficie superiore della molla
Figura 4.11: Nodi vincolati appartenenti alla superficie inferiore della molla
Figura 4.12: Andamento dello spostamento verticale in funzione del numero di
elementi
Figura 4.13: Andamento della tensione equivalente in funzione del numero di elementi
Figura 4.14: Andamento dell'errore relativo allo spostamento verticale in funzione del
numero di elementi
Figura 4.15: Andamento dell'errore relativo alla tensione equivalente in funzione del numero di elementi
Figura 4.16: Andamento del tempo di simulazione in funzione del numero di elementi 58
Figura 4.17-a: Mappa degli spostamenti
Figura 4.17-b: Mappa delle tensioni equivalenti
 Figura 4.17-b: Mappa delle tensioni equivalenti
Figura 4.17-b: Mappa delle tensioni equivalenti
Figura 4.17-b: Mappa delle tensioni equivalenti.59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di convergenza60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66
Figura 4.17-b: Mappa delle tensioni equivalenti.59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di convergenza60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66Figura 5.4: Curva di Wöhler67
Figura 4.17-b: Mappa delle tensioni equivalenti.59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di convergenza60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66Figura 5.4: Curva di Wöhler67Figura 5.5: Retta di Basquin67
Figura 4.17-b: Mappa delle tensioni equivalenti.59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di convergenza60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66Figura 5.4: Curva di Wöhler67Figura 5.5: Retta di Basquin67Figura 5.6: Diagramma di Haigh68
Figura 4.17-b: Mappa delle tensioni equivalenti.59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di convergenza60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66Figura 5.4: Curva di Wöhler67Figura 5.5: Retta di Basquin67Figura 5.6: Diagramma di Haigh68Figura 5.7: Fattori che influenzano la vita a fatica69
Figura 4.17-b: Mappa delle tensioni equivalenti.59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66Figura 5.4: Curva di Wöhler67Figura 5.5: Retta di Basquin67Figura 5.6: Diagramma di Haigh68Figura 5.7: Fattori che influenzano la vita a fatica69Figura 5.8: Influenza della tipologia di carico sul limite di fatica70
Figura 4.17-b: Mappa delle tensioni equivalenti
Figura 4.17-b: Mappa delle tensioni equivalenti.59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66Figura 5.4: Curva di Wöhler67Figura 5.5: Retta di Basquin67Figura 5.6: Diagramma di Haigh68Figura 5.7: Fattori che influenzano la vita a fatica69Figura 5.8: Influenza della tipologia di carico sul limite di fatica70Figura 5.9: Modello di danneggiamento lineare di Palmgren-Miner71Figura 5.10: Cicli sinusoidali con tensione media e alternata variabili72
Figura 4.17-b: Mappa delle tensioni equivalenti
Figura 4.17-b: Mappa delle tensioni equivalenti59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66Figura 5.4: Curva di Wöhler67Figura 5.5: Retta di Basquin67Figura 5.6: Diagramma di Haigh68Figura 5.7: Fattori che influenzano la vita a fatica69Figura 5.9: Modello di danneggiamento lineare di Palmgren-Miner71Figura 5.10: Cicli sinusoidali con tensione media e alternata variabili72Figura 5.11: Valutazione della tensione limite di fatica i-esima72Figura 5.12: Costruzione del diagramma di Wöhler i-esimo72
Figura 4.17-b: Mappa delle tensioni equivalenti59Figura 4.18: Area del componente sottoposta alla massima tensione equivalente59Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di60Figura 5.1: Esempi di carichi random64Figura 5.2: Rottura per fatica65Figura 5.3: Tipologie di cicli di tensione66Figura 5.4: Curva di Wöhler67Figura 5.5: Retta di Basquin67Figura 5.6: Diagramma di Haigh68Figura 5.7: Fattori che influenzano la vita a fatica69Figura 5.9: Modello di danneggiamento lineare di Palmgren-Miner71Figura 5.10: Cicli sinusoidali con tensione media e alternata variabili72Figura 5.11: Valutazione della tensione limite di fatica i-esima72Figura 5.12: Costruzione del diagramma di Wöhler i-esimo72Figura 5.13: Valutazione dell'i-esimo danno subito dal componente73

Figura 5.15: Esempio di applicazione del metodo di conteggio rainflow a 3 punti	·75
Figura 5.16: Spostamento verticale della molla della sospensione in funzione del	
tempo	$\cdot 75$
Figura 5.17: Interfaccia grafica del software FEMFAT	•77
Figura 5.18: Identificazione dei parametri di input delle analisi a fatica	•77
Figura 5.19: Esempio di ciclo sinusoidale	.78
Figura 5.20: Curva di Wöhler del materiale	80
Figura 5.21: Diagramma di Haigh del materiale	80
Figura 5.22: Definizione dei Channels in FEMFAT	. 81
Figura 5.23: Definizione degli Influence Factors in FEMFAT	.82
Figura 5.24: Scelta del target dell'analisi in FEMFAT	.82
Figura 5.25: Definizione del punto di lavoro nel diagramma di Haigh	.83
Figura 5.26: Stato di tensione nel piano critico	.84
Figura 5.27: Mappa di danneggiamento del componente nella zona maggiormente sollecitata	.85
Figura 5.28: Andamento del danno in funzione del numero di cicli di sollecitazione	.87
Figura 5.20: Andamento del danno in funzione della componente alternata di	,0,
spostamento	.87
Figura 6.1: Classificazione di alcune tecniche di intelligenza artificiale	90
Figura 6.2: Modelli di apprendimento del machine learning	90
Figura 6.3: Tecniche di machine learning	. 91
Figura 6.4: Schema di funzionamento del reinforcement learning	.92
Figura 6.5: Struttura delle reti neurali	.93
Figura 6.6: Neurone di una rete neurale	.93
Figura 6.7: Processo di addestramento della rete neurale	.94
Figura 6.8: Algoritmo di forward-backward propagation	.95
Figura 6.9: Funzione di attivazione "Sigmoide"	.98
Figura 6.10: Funzione di attivazione "Softplus"	.98
Figura 6.11: Funzione di attivazione "Relu"	.99
Figura 6.12: Funzione di attivazione "Leaky Relu"	.99
Figura 6.13: Funzione di attivazione "Tangente iperbolica"1	100
Figura 6.14: Funzione di attivazione "Softsign"1	100
Figura 6.15: Esempi di distribuzioni normali	101
Figura 6.16: Dropout	102
Figura 6.17: Suddivisione del database	105
Figura 6.18: Esempio di curva di apprendimento	105
Figura 6.19: Esempio di learning curve con underfitting	106
Figura 6.20: Esempio di learning curve con overfitting	106
Figura 6.21: Esempio di learning curve con training set non rappresentativo	107
Figura 6.22: Esempio di learning curve con validation set non rappresentativo	107
Figura 6.23: Visualizzazione dell'algoritmo di discesa del gradiente su una funzione	
quadratica a due variabili	108

Figura 6.24: Influenza del momentum sulla velocità di convergenza del modello109 Figura 6.25: Influenza degli optimizers sull'addestramento di una rete neurale impiegata
per classificazione di immagini
Figura 6.26: Ricerca del minimo globale di una funzione a più variabili
Figura 6.27: Influenza del learning rate sulla ricerca del punto di minimo della loss function
Figura 6.28: Influenza del learning rate sull'andamento della loss function
Figura 6.29: Influenza del learning rate sull'addestramento della rete
Figura 6.30: Influenza del batch size sull'addestramento della rete114
Figura 6.31: Divisione del database di progetto117
Figura 6.32: Impostazione degli iperparametri per l'addestramento in Python 118
Figura 6.33: Utilizzo di una rete neurale addestrata in Python
Figura 6.34: Andamenti delle loss functions relative alla rete neurale ottimizzata 126
Figura 7.1: Dati sperimentali relativi alla prova 3127
Figura 7.2: Dati sperimentali relativi alla prova 4 127
Figura 7.3: Dati sperimentali relativi alla prova 5
Figura 7.4: Dati sperimentali relativi alla prova 10128
Figura 7.5: Estratto di codice per la scomposizione della storia di carico in sequenze di picchi e valli
Figura 7.6: Storia di carico prima della scomposizione in una sequenza di picchi e valli
Figura 7.7: Storia di carico dopo la scomposizione in una sequenza di picchi e valli 130
Figura 7.8: Estratto di codice per la previsione del danno complessivo accumulato nel componente
Figura 7.9: Convergenza del danno in funzione del numero di intervalli dello spostamento alternato
Figura 7.10: Convergenza del danno in funzione del numero di intervalli dello spostamento medio
Figura A.1: Raspberry Pi 3 Model B+ 135
Figura A.2: Flashing della scheda SD 137
Figura A.3: Simboli dei sistemi operativi Raspbian e NOOBS 137
Figura A.4: Raspberry Pi 4 GPIO Header
Figura A.5: Voltaggio e numerazione dei pin
Figura A.6: Breadboard per Raspberry e Arduino
Figura A.7: Sensore ad ultrasuoni HC-SR04140
Figura A.8: Andamenti dei segnali elettrici dei pin TRIG e ECHO141
Figura A.9: Circuito elettrico per la connessione del sensore HC-SR04 al Raspberry Pi 4
Figura A.10: Prima parte del codice per l'utilizzo del sensore HC-SR04
Figura A.11: Seconda parte del codice per l'utilizzo del sensore HC-SR04

Elenco delle tabelle

Tabella 1.1: Cause ed effetti dei guasti nei gruppi ruota automobilistici	.14
Tabella 1.2: Cause ed effetti dei guasti negli impianti frenanti automobilistici	.14
Tabella 1.3: Cause ed effetti dei guasti nelle sospensioni automobilistiche	.16
Tabella 1.4: Parametri di valutazione per la scelta dello use-case del progetto	. 17
Tabella 1.5: Valutazioni dettagliate dei componenti	.18
Tabella 1.6: Valutazioni complessive dei componenti analizzati per la scelta dello use-ca	se
	. 19
Tabella 2.1: Principali sensori impiegati in ambito industriale	24
Tabella 2.2: Sensori per misure di spostamento e deformazione	.25
Tabella 2.3: Caratteristiche tecniche sensore laser ILD1220-200	28
Tabella 4.1: Parametri geometrici molla	.47
Tabella 4.2: Proprietà dell'acciaio 55 SiCr 6-3	48
Tabella 4.3: Possibili trattamenti termici ed impieghi del materiale 55 SiCr 6-3	49
Tabella 4.4: Risultati dell'analisi di convergenza dopo lo step 1	.55
Tabella 4.5: Suddivisione del componente in 5 zone per la fase 2 dell'analisi di	00
convergenza	60
Tabelle 4.6-4.7-4.8: Risultati delle simulazioni di riferimento	.61
Tabelle 4.9-4.10: Risultati della prima e seconda simulazione	.61
Tabelle 4.11-4.12: Risultati della terza e quarta simulazione	62
Tabella 4.13: Modello selezionato a valle dell'analisi di convergenza	62
Tabella 5.1: Parametri per la stima del danneggiamento del componente nel caso di cicli	di
sollecitazione sinusoidali con tensione media ed alternata variabili	.73
Tabella 5.2: Risultato del conteggio con il metodo rainflow per la storia di carico in figur	ra
5.15	.75
Tabella 5.3: Parametri per la stima del danneggiamento della molla della sospension	ie
The line of the second se	.'/0
Tabella 5.4: Lunghezza della molta della sospensione in funzione delle condizioni di lavo	.78
Tabella 5.5: Intervalli di variazione delle grandezze di input per le simulazioni a fatic	ea
Taballa 5 6: Propriotà del materiale	.78
Tabella 5.0. I Toprieta dei materiale	- /9 i
Tabella 5./. Tensioni ininte di fatica dei materiale per diverse upologie di siorzi element	.70
Tabella 5.8: Danneggiamento del componente con spostamento medio costante pari a	
118,2 mm	86
Tabella 6 1. Parametri per la generazione di una rete neurale	06
Tabella 6.2. Parametri per la configurazione di una rete neurale	90
Tabella 6 3: Parametri per l'addestramento di una rete neurale	90
- as one of the analysis por the according to the found of the found of the second of	20

Tabella 6.4: Esempi di activation functions	97
Tabella 6.5: Step da seguire per l'utilizzo della rete neurale	. 115
Tabella 6.6: Input e output della rete neurale	. 116
Tabella 6.7: Parametri e risultati dell'addestramento della rete neurale 1	.120
Tabella 6.8: Parametri e risultati dell'addestramento della rete neurale 2	.120
Tabella 6.9: Parametri e risultati dell'addestramento della rete neurale 3	. 121
Tabella 6.10: Parametri e risultati dell'addestramento della rete neurale 4	. 121
Tabella 6.11: Parametri e risultati dell'addestramento della rete neurale 5	.122
Tabella 6.12: Parametri e risultati dell'addestramento della rete neurale 6	.122
Tabella 6.13: Parametri e risultati dell'addestramento della rete neurale 7	.123
Tabella 6.14: Parametri e risultati dell'addestramento della rete neurale 8	.123
Tabella 6.15: Parametri e risultati dell'addestramento della rete neurale 9	.124
Tabella 6.16: Parametri della rete neurale ottimizzata	.124
Tabella 6.17: Training set, validation set e test set della rete neurale ottimizzata	.125
Tabella 6.18: Confronto tra il danno atteso e quello predetto dalla rete neurale ottimizz	zata
	. 125

Tabella 7.1: Previsione della distanza rimanente prima della rottura del componente....133

1 Introduzione al progetto di ricerca e sviluppo

1.1 Tipologie di manutenzione

Le moderne tecnologie digitali stanno mutando profondamente lo stile di vita della maggioranza della popolazione mondiale, sia in ambito privato che professionale. Allo stesso modo, il settore industriale sta vivendo un notevole processo di innovazione grazie agli incentivi per lo sviluppo dell'Industria 4.0 e delle smart factories. In questo contesto, si stanno evolvendo anche le tecniche di controllo dei macchinari e dei componenti; oltre alle tradizionali manutenzioni correttive e periodiche, in diversi settori si sta diffondendo la cosiddetta "predictive maintenance". Per comprenderne le differenze, in figura 1.1 si riporta una classificazione delle diverse tipologie di manutenzione.



Figura 1.1: Classificazione delle tecniche di manutenzione

La manutenzione correttiva interviene nella riparazione delle macchine solo in presenza di avarie o guasti, riducendo pesantemente l'efficienza dell'intero sistema produttivo. La manutenzione su condizione, particolare tipologia di manutenzione preventiva, mira all'ispezione di un componente nel momento in cui un suo parametro caratteristico raggiunge un valore di soglia prestabilito che rischia di comprometterne il funzionamento. Il monitoraggio di un sistema si effettua mediante strumenti tradizionali di acquisizione dati, come contatori o semplici sensori. Il controllo del livello di olio motore in un'auto o della pressione di un fluido in un condotto sono alcuni esempi di manutenzione su condizione. In una strategia di manutenzione periodica, tecnici specializzati revisionano e puliscono i macchinari di un impianto ad intervalli di tempo regolari, procedendo alla sostituzione di eventuali componenti danneggiati per evitare cedimenti improvvisi. Questa tecnica permette di ridurre la probabilità di malfunzionamenti e non richiede una fase di training del personale particolarmente complessa. Tuttavia, una pianificazione rigida delle ispezioni di controllo rischia di interrompere inutilmente la produzione e di cambiare delle parti prima che siano effettivamente giunte al termine del ciclo vita: il limite di questo metodo deriva dal fatto che i fermi macchina sono eventi stocastici e non deterministici. Contrariamente alle precedenti tecniche, la manutenzione predittiva non si limita a prevenire un guasto ma a prevedere la probabilità con cui potrebbe accadere nel futuro. Gli strumenti che consentono di attuarla sono visibili a pagina seguente.



Figura 1.2: Strumenti della manutenzione predittiva

Gli algoritmi di intelligenza artificiale hanno un ruolo centrale poiché sono in grado di rilevare i pattern nascosti nei dati forniti dall'utente, permettendo di effettuare delle previsioni sullo stato di salute di un sistema e di stimare il tempo residuo prima di una rottura. Definendo con razionalità i piani di manutenzione si migliorano le performance complessive del processo produttivo e la qualità dei prodotti finiti. I sensori IoT, collegati alla rete, sono impiegati per acquisire e monitorare in tempo reale un'elevata quantità di dati provenienti da differenti parti di un impianto. L'utilizzo del cloud computing è idoneo per archiviare le informazioni in un database remoto, da cui possono essere prelevate ed elaborate tramite appositi algoritmi. La combinazione di tecniche di deep learning, come ad esempio le reti neurali, di sensori IoT e di sistemi per la gestione dei Big Data costituisce il punto di partenza per l'implementazione di strategie di manutenzione predittiva. Inoltre, la prototipazione virtuale, le applicazioni software di ingegneria assistita dal computer (Computer Aided Engineering, CAE) e lo sviluppo di modelli matematico-statistici rappresentano ulteriori strumenti per la valutazione dei danneggiamenti subiti dai componenti di un assieme meccanico. Gli svantaggi di questo metodo risiedono nei maggiori costi di investimento iniziali rispetto alle classiche strategie di manutenzione, nel complesso processo di formazione del personale e nel lungo periodo di adattamento alle nuove dinamiche produttive aziendali. Si sottolinea che i giovamenti di tale approccio sono tangibili in molti altri contesti produttivi diversi da quello industriale, come ad esempio nei settori energetico, edilizio, agricolo o automotive.

Per questo motivo, si assiste ad un crescente e costante incremento del numero di aziende che investono notevoli risorse nella realizzazione di progetti di manutenzione predittiva.

1.2 La manutenzione predittiva in ambito automotive

Il concetto di manutenzione predittiva nasce all'interno degli impianti industriali per massimizzare la produttività dei macchinari e ridurre i costi di gestione. Tuttavia, anche il mondo delle auto può trarre beneficio dall'utilizzo di questa nuova modalità di controllo dello stato. dei componenti. In particolare, i vantaggi introdotti nel settore automotive sono:

- Riduzione dei costi;
- Aumento della sicurezza e dell'affidabilità del veicolo;
- Estensione del ciclo di vita del prodotto;
- Pianificazione razionale degli interventi di controllo e riparazione;
- Incremento delle prestazioni della vettura;
- Aumento della qualità dell'esperienza di guida;
- Miglioramento dell'assistenza tecnica al cliente nella fase post-vendita.

Si precisa che la manutenzione predittiva può essere adottata per conoscere in tempo reale le condizioni di diversi elementi meccanici presenti in un'autovettura, come ad esempio i dischi freno, gli pneumatici, gli ammortizzatori delle sospensioni o i componenti del motore.



Figura 1.3: Flusso dei dati per il monitoraggio dello stato di salute di un componente

Lo schema raffigurato in figura 1.3 agevola la comprensione del metodo sviluppato in Capgemini Engineering per prevedere il danneggiamento dei componenti durante la marcia del veicolo. Indipendentemente dal gruppo meccanico considerato, si comincia sempre dall'acquisizione delle variabili caratteristiche del sistema mediante sensori IoT: in questo modo è possibile conoscerne lo stato di funzionamento in ogni istante temporale. Una volta archiviate le informazioni all'interno di dispositivi elettronici dedicati, si sceglie se monitorare semplicemente l'andamento dei parametri rilevati o se pre-processare i dati. Nel primo caso si visualizzano a schermo i segnali ricevuti dai sensori, azione utile per verificare il corretto funzionamento dell'intero apparato di acquisizione. Nel secondo caso i dati divengono i parametri di input di una rete neurale, ossia un modello computazionale matematico-informatico in grado di prevedere i danni subiti dal sistema con un determinato livello di affidabilità. A questo punto, le grandezze di output del modello vengono adeguatamente salvate e l'intero processo ricomincia da capo.

La costruzione di un database virtuale, eseguita attraverso simulazioni agli elementi finiti, è necessaria per effettuare offline il cosiddetto addestramento della rete neurale, un'operazione che

permette al modello surrogato di imparare le relazioni che legano le grandezze di input e di output; senza una preliminare fase di training, la rete perderebbe ogni capacità predittiva. Appare evidente come le analisi FEM risultino parimenti cruciali ai fini dell'applicabilità del metodo, in quanto consentono di conoscere la risposta del sistema quando viene sottoposto a sollecitazioni analoghe a quelle incontrate durante il moto dell'automobile. Il set up delle simulazioni si basa sugli andamenti delle variabili monitorate mediante i trasmettitori installati a bordo della vettura. L'efficacia e l'accuratezza della metodologia descritta sono tanto migliori quanto maggiore è la sinergia tra gli strumenti indispensabili alla manutenzione predittiva, ovvero sensori, prototipazione virtuale e intelligenza artificiale.

1.3 Analisi dei danni subiti da alcuni componenti dell'autoveicolo

La fase iniziale del progetto è incentrata sullo studio delle possibili cause di danneggiamento di componenti appartenenti a diversi assiemi, nonché sull'impatto di eventuali avarie in termini di handling e comfort della vettura. I sistemi meccanici analizzati sono l'impianto frenante, la sospensione anteriore e il gruppo ruota, in quanto rappresentano il giusto compromesso tra complessità costruttiva, utilità della manutenzione real-time e accessibilità agli elementi costituenti. Il gruppo ruota è formato principalmente dai seguenti componenti:

- Pneumatico: trasferisce a terra la potenza generata dal motopropulsore, consentendo l'avanzamento del veicolo;
- Mozzo: sede di collegamento di semiasse, ruota, sospensione ed elementi dell'impianto frenante;
- Cerchione: parte della ruota che sostiene lo pneumatico, realizzati in lega o in acciaio.

In figura 1.4 si riporta un esempio di vista in sezione.



Figura 1.4: Principali componenti di un gruppo ruota automobilistico

L'usura o addirittura la rottura di uno dei componenti di questo assieme meccanico condurrebbe alla nascita di vibrazioni al volante, al consumo anomalo degli pneumatici e ad una ridotta tenuta di strada, pregiudicando la sicurezza dei passeggeri. In tabella 1.1 sono indicate le cause di danneggiamento degli elementi del gruppo ruota e le corrispondenti conseguenze in termini di maneggevolezza e comfort.

Principali malfunzionamenti nei gruppi ruota automobilistici					
Componente	Funzione	Causa rottura/usura	Effetto del danno su Handling & Comfort vettura		
Pneumatico	Collega il veicolo con la strada permettendo la trazione, sostiene il peso della vettura	Errata pressione di gonfiaggio, componenti sospensione in cattivo stato, perdita della convergenza, stile di guida del conducente, condizioni della strada	Rumorosità, vibrazioni, aumento dei consumi, minore tenuta di strada, rischio di foratura o scoppio, surriscaldamento battistrada		
Mozzo	Collega il semiasse, la sospensione, il freno e il cerchio alla ruota	Usura dei cuscinetti, urti, condizioni della strada	Rumorosità, vibrazioni, scollegamento della ruota, eccessiva usura pneumatici, instabilità di guida		
Cerchione	Collega lo pneumatico al veicolo	Condizioni della strada, errori di montaggio, urti	Deformazioni, pneumatico a terra, vibrazioni al volante, ridotta tenuta di strada, disallineamento ruota		

Tabella 1.1: Cause ed effetti dei guasti nei gruppi ruota automobilistici

Il secondo sistema esaminato è l'impianto frenante, costituito da:

- Pinza: monoblocco o in due pezzi, fissa o flottante. Contiene i cilindretti idraulici che, mossi dal fluido in pressione proveniente dalla pompa, spingono le pastiglie contro il disco, generando una coppia frenante che riduce la velocità angolare della ruota;
- Disco: pieno, forato, ventilato o baffato. È solidale alla ruota e in fase di frenata entra in contatto con le pastiglie;
- Pastiglia: ottenuta per sinterizzazione o con materiali organici; è situata all'interno della pinza.



Figura 1.5: Principali componenti di un impianto frenante automobilistico

In tabella 1.2 sono riportate le cause di danneggiamento dei componenti dell'impianto frenante e i corrispondenti effetti in termini di maneggevolezza e comfort del veicolo.

Principali malfunzionamenti negli impianti frenanti automobilistici					
Componente	Funzione	Causa rottura/usura	Effetto del danno su Handling & Comfort vettura		
Pinza	Distribuisce il fluido freni proveniente dalla pompa alle pastiglie	Deterioramento, rottura accidentale, condizioni ambientali avverse, errori di montaggio, utilizzo di un fluido freni inadatto	Instabilità in frenata, riduzione dell'efficienza di frenata, perdite di liquido freni, rumorosità eccessiva		
Pastiglia	Situata all'interno della pinza; entrando a contatto con il disco permette di generare la forza frenante	Attrito meccanico, errori di montaggio, malfunzionamenti della pinza	Rumorosità durante la frenata, vibrazioni, presenza di righe o cricche sulla fascia frenante, segni di surriscaldamento, riduzione liquido freni, usura eccessiva dischi		
Disco	Solidale alla ruota; entrando a contatto con le pastiglie permette di generare la forza frenante	Attrito meccanico, elevate quantità di calore, procedura di rodaggio scorretta, errori di montaggio	Rumorosità durante la frenata, vibrazioni, presenza di righe o cricche sulla fascia frenante, segni di surriscaldamento, riduzione liquido freni, usura eccessiva pastiglie		

Tabella 1.2: Cause ed effetti dei guasti negli impianti frenanti automobilistici

Dischi e pastiglie si usurano a causa dell'attrito generato dall'adesione delle superfici di contatto. Per evitare surriscaldamenti eccessivi, i dischi freno sono realizzati con numerosi fori e canali interni per consentire ai flussi d'aria che li attraversano di smaltire opportunamente il calore prodotto in fase di frenata. D'altra parte, le pinze si rompono solitamente in modo accidentale. Il malfunzionamento di un qualsiasi componente del sistema comprometterebbe fortemente la sicurezza del conducente e dei passeggeri.

Il terzo e ultimo assieme meccanico analizzato è relativo ad un tipico schema di sospensione anteriore. Con riferimento alla figura 1.6, gli elementi principali che la compongono sono:

- Fuso a snodo dello sterzo (1): contiene parte del mozzo ruota, dell'albero del semiasse, del montante dell'ammortizzatore e del disco freno. Risulta essenziale per un corretto allineamento degli pneumatici;
- Barra stabilizzatrice (2): è connessa ad entrambe le sospensioni anteriori e lavorando a torsione limita i movimenti di rollio della vettura, soprattutto in curva;
- Collegamento stabilizzatore (3): unisce il montante dell'ammortizzatore con la barra antirollio;
- Molla-ammortizzatore (4): rappresentano l'elemento elastico e smorzante della sospensione, rispettivamente. Evitano che le vibrazioni dovute alle irregolarità del manto stradale si trasmettano all'interno dell'abitacolo e sostengono il carico verticale dovuto alle masse sospese del veicolo;
- Braccio di controllo (5): congiunge il telaio dell'automobile con la sospensione, permettendo il controllo del movimento verticale e laterale delle ruote.



Figura 1.6: Principali componenti di una sospensione automobilistica

Analogamente ai casi precedenti, a pagina seguente si evidenziano le relazioni di causa-effetto tra i danni subiti dalle varie parti della sospensione e le performance del veicolo. Nella tabella 1.3 sono indicati alcuni componenti aggiuntivi rispetto a quelli descritti poc'anzi. Le boccole stabilizzatrici e i giunti sferici sono organi di collegamento, mentre i silent block sono dei particolari cuscinetti antivibranti dotati di inserti in gomma, facilmente degradabili se esposti ad agenti chimici aggressivi o a temperature eccessivamente elevate.

Principali malfunzionamenti nelle sospensioni automobilistiche				
Componente	Funzione	Causa rottura/usura	Effetto del danno su Handling & Comfort vettura	
Molla	Elemento elastico	Rottura statica o a fatica, usura, sovraccarico o urto, corrosione, usura dei supporti di sostegno	Variazione altezza da terra del veicolo, incremento delle vibrazioni, usura anomala pneumatici	
Ammortizzatore	Elemento smorzante	Corrosione, corpi estranei contaminanti, urti, usura guarnizioni, esposizione ad agenti aggressivi, sovraccarichi	Perdita di contatto ruota-strada, incremento spazio di frenata, perdite d'olio, segni di abrasione, ampie oscillazioni del veicolo	
Barra stabilizzatrice	Anti-rollio	Usura dei silent block o delle biellette stabilizzatrici, urti	Peggioramento performance dinamiche del veicolo	
Fuso a snodo	Elemento di collegamento	Lubrificazione insufficiente, urti, condizioni ambientali aggressive, corrosione	Peggioramento reattività dello sterzo, battiti provenienti dalla sospensione, disallineamento ruote, usura pneumatici	
Collegamenti (biellette) stabilizzatori	Elemento di collegamento	Urti, usura, accoppiamenti imprecisi, stress meccanico	Vibrazioni, guida instabile, rumorosità	
Boccole stabilizzatrici	Supporto e collegamento	Fatica, usura meccanica	Rumore o cigolii provenienti dall'asse interessato	
Bracci di controllo	Elemento di collegamento; controllo movimento pneumatico	Usura meccanica, urti, corrosione	Rumorosità durante la marcia, singoli pneumatici non guidati correttamente, consumo anomalo gomme, problemi di convergenza	
Giunti sferici	Elemento di collegamento	Usura, condizioni ambientali aggressive, urti, sovraccarichi	Incremento vibrazioni, peggioramento maneggevolezza veicolo, possibile scollegamento di una ruota, rumorosità	
Silent block	Cuscinetti antivibranti	Usura meccanica, termica o chimica dell'inserto elastomerico, carichi dinamici, urti	Consumo pneumatici non uniforme, perdita della tenuta di strada, aumento rumorosità, problemi di convergenza	

Tabella 1.3: Cause ed effetti dei guasti nelle sospensioni automobilistiche

1.4 Selezione dello use-case del progetto

Una volta approfondite le modalità di guasto relative agli elementi della sospensione anteriore, del gruppo ruota e dell'impianto frenante, è necessario scegliere un componente che permetta non solo di verificare la fattibilità del progetto, ma anche di implementare la metodologia ideata per effettuare manutenzione predittiva in tempo reale. Per la selezione dello use-case, si costruisce un criterio di valutazione personalizzato attraverso i seguenti passaggi:

- Si definiscono venti parametri di valutazione suddivisi in cinque categorie differenti, ovvero Valore aggiunto per Capgemini, Stato dell'arte, Componente, Sensore e Simulazione;
- Si assegna un peso variabile nell'intervallo [1,5] ad ogni parametro di valutazione in funzione della sua rilevanza ai fini della manutenzione predittiva;
- Per ogni parametro, si conferisce ai componenti dei sistemi meccanici analizzati una valutazione variabile nell'intervallo [1,5];
- Si calcolano le valutazioni complessive di tutti i componenti mediante una media ponderata e si seleziona quello con la valutazione complessiva maggiore.

Per chiarezza, si descrive brevemente il significato di alcuni parametri:

- Expertise (1): disponibilità di personale competente e professionale in un ambito tecnico di interesse;
- Sviluppi futuri (3): volontà dell'azienda di investire in progetti correlati nel medio-lungo periodo;
- Know-how del sistema (4): attuale conoscenza e padronanza della tecnologia di un sistema meccanico;
- Complessità (7): si riferisce alla geometria del componente e alla sua posizione nel sistema meccanico di appartenenza;

- Tipologie (8): si considerano tutte le classi esistenti di un componente, anche quelle non impiegate nei sistemi meccanici presentati;
- Costo (9): si riferisce al solo componente, non alla manutenzione o alla manodopera;
- Conseguenze della rottura (10): si riferisce al beneficio ottenuto con la manutenzione predittiva in termini di risparmio sui costi e tempi di sostituzione del componente danneggiato;
- Utilità manutenzione real-time (12): si valuta in funzione del rischio per la sicurezza conseguente alla rottura del componente;
- Accessibilità al componente (13): semplicità di accesso al componente installato nel veicolo;
- Numero di sensori necessari al test in vettura (14): la geometria e il tipo di componente determinano la quantità di sensori da installare;
- Costo (16): si riferisce al solo sensore, non al sistema di condizionamento o al circuito di acquisizione;
- Frequenza variazione dati per integrazione con AI (18): la rapidità con cui si modificano i valori delle grandezze fisiche monitorate influenza la dimensione del database utilizzato per gli addestramenti delle reti neurali, che necessitano di elevate quantità di dati per eseguire previsioni affidabili;
- Semplicità modello di danneggiamento (19): la tipologia di analisi da sviluppare (termica, strutturale, chimica, elettromagnetica, fluidodinamica...) varia a seconda del componente scelto.

Nelle tabelle 1.4 e 1.5 sono visibili i parametri, i pesi e le valutazioni assegnate ad ogni elemento meccanico.

CATEGORIE	PARAMETRI DI VALUTAZIONE	PESO PARAMETRI [1-5]	DESCRIZIONE VALUTAZIONE [1-5]		
	1) Expertise	2	[1=nessuna expertise]		
Valore aggiunto per Capgemini	2) Multidisciplinarità	2	[1=molto scarsa]		
	3) Sviluppi futuri	3	[1=basso interesse]		
	4) Know-how del sistema	1	[1=scarsa conoscenza]		
Stato dell'arte	5) Investimenti-ricerca	2	[1=investimenti esigui]		
	6) Tecniche di manutenzione predittiva esistenti	4	[1=manutenzione predittiva già esistente ed efficace]		
	7) Complessità	3	[1=molto elevata]		
	8) Tipologie	1	[1=poche tipologie]		
Componento	9) Costo	4	[1=elevato]		
Componente	10) Conseguenze della rottura	5	[1=vantaggio esiguo]*		
	11) Influenza sulla sicurezza	5	[1=poco influente]		
	12) Utilità manutenzione real-time	5	[1=scarsa utilità]		
	13) Accessibilità al componente	4	[1=molto ridotta]		
	14) Numero di sensori necessari al test in vettura	3	[1=molto elevato]		
Sensore	15) Facilità di installazione	4	[1=difficile]		
	16) Costo	3	[1=molto elevato]		
	17) Disponibilità sul mercato	4	[1=tempo di approvvigionamento elevato]		
	18) Frequenza variazione dati per integrazione con Al	5	[1=molto lenta]		
Simulazione	19) Semplicità modello di danneggiamento	4	[1=complesso]		
	20) Semplicità modello FEM	2	2 [1=complesso]		
*Si riferisce al benefecio o	ttenuto con la manutenzione predittiva in termini di rispa	rmio sui costi e temp	i di sostituzione del componente danneggiato		

Tabella 1.4: Parametri di valutazione per la scelta dello use-case del progetto

SOSPENSIONE ANTERIORE									11	MPIANTO	FRENANT	ſE				GRUPPC	RUOTA				
Ма	olla	Ammort	tizzatore	Braco cont	tio di Tollo	Collega stabiliz	mento zatore	Barra ar	nti-rollio	Pir	za	Past	iglia	Dis	со	Pneun	natico	Мо	zzo	Cerch	nione
Р	V	Р	V	Р	V	Р	V	Р	V	Р	V	Р	V	Р	V	Р	V	Р	V	Р	V
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
2	2	2	5	2	3	2	2	2	3	2	3	2	3	2	4	2	5	2	2	2	2
3	4	3	4	3	4	3	4	3	4	3	4	3	4	3	4	3	4	3	4	3	4
4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	4	4	4	4	4
5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	5	5	5	5	5
6	5	6	3	6	5	6	5	6	3	6	4	6	2	6	4	6	1	6	3	6	4
7	5	7	2	7	3	7	4	7	4	7	2	7	3	7	3	7	2	7	3	7	3
8	4	8	2	8	5	8	1	8	1	8	2	8	3	8	3	8	4	8	2	8	4
9	4	9	2	9	3	9	4	9	3	9	3	9	3	9	2	9	4	9	3	9	2
10	3	10	5	10	4	10	2	10	3	10	3	10	2	10	3	10	2	10	5	10	4
11	4	11	4	11	4	11	3	11	4	11	5	11	5	11	5	11	5	11	4	11	5
12	3	12	4	12	4	12	3	12	3	12	4	12	2	12	4	12	4	12	4	12	4
13	3	13	4	13	4	13	3	13	4	13	4	13	3	13	4	13	5	13	4	13	5
14	5	14	2	14	3	14	3	14	3	14	2	14	3	14	2	14	3	14	3	14	4
15	3	15	2	15	3	15	3	15	3	15	2	15	2	15	2	15	4	15	3	15	3
16	3	16	3	16	3	16	3	16	2	16	3	16	2	16	3	16	3	16	4	16	4
17	3	17	3	17	2	17	2	17	2	17	4	17	3	17	3	17	4	17	3	17	3
18	4	18	4	18	4	18	4	18	4	18	4	18	2	18	4	18	3	18	3	18	3
19	4	19	1	19	3	19	3	19	3	19	1	19	2	19	2	19	2	19	3	19	3
20	5	20	2	20	3	20	3	20	4	20	2	20	5	20	4	20	3	20	3	20	3
	P = Parametro di valutazione ; V = Valutazione																				

Tabella 1.5: Valutazioni dettagliate dei componenti

Per comprendere la logica di attribuzione di alcune valutazioni, si riportano sinteticamente alcune considerazioni:

- Alla voce "*Tecniche di manutenzione predittiva esistenti (6)*", allo pneumatico è assegnato un voto pari ad 1. Infatti, l'azienda Goodyear ha già sviluppato un sistema di manutenzione predittiva real-time per il controllo della temperatura e della pressione degli pneumatici;
- Alla voce "*Tecniche di manutenzione predittiva esistenti (6)*", all'ammortizzatore è assegnato un voto pari a 3. L'introduzione delle sospensioni attive nel mondo automotive consente una regolazione in tempo reale delle variabili del sistema, ma solo in ottica di modifica dell'assetto del veicolo e non di manutenzione;
- Alla voce "*Tipologie (8)*", al braccio di controllo è assegnato un voto pari a 5 poiché esistono molteplici soluzioni costruttive a seconda della tipologia di sospensione considerata (Multilink, a quadrilatero, McPherson, a ponte...);
- Alla voce "*Utilità manutenzione real-time (12)*", alla pastiglia è assegnato un voto pari a 2 in quanto si usura molto lentamente prima di compromettere la sicurezza dei passeggeri;
- Alla voce "*Numero di sensori necessari al test in vettura (14)*", alla molla è assegnato un voto pari a 5 poiché è sufficiente acquisire solamente lo spostamento verticale del componente per valutarne lo stato di salute;
- Alla voce "Semplicità modello di danneggiamento (19)", alla pinza freno è assegnato un voto pari ad 1 a causa delle numerose conoscenze multidisciplinari necessarie alla costruzione di un modello di danneggiamento affidabile;
- Alla voce "Semplicità modello FEM (20)", all'ammortizzatore è assegnato un voto pari a 2 perché la geometria del componente e i collegamenti con gli altri elementi della sospensione anteriore risultano complessi.

Il voto complessivo V_c di ogni componente è ricavato mediante i pesi P_i e le singole valutazioni V_i :

$$V_c = \sum_{i=1}^{20} \frac{V_i \cdot P_i}{P_i}$$

Nella tabella 1.6 sono indicate le valutazioni complessive di ogni componente; a valle dello studio, si può asserire che applicare strategie di manutenzione predittiva alla pastiglia dei freni risulta difficile. Viceversa, si seleziona la molla della sospensione anteriore come use-case per il progetto.

Sistema meccanico	Componente	Valutazione complessiva
	Molla	3,70
	Ammortizzatore	3,20
Sospensione anteriore	Braccio di controllo	3,53
	Collegamento stabilizzatore	3,18
	Barra anti-rollio	3,23
	Pinza	3,24
Impianto frenante	Pastiglia	2,83
	Disco	3,33
	Pneumatico	3,39
Gruppo ruota	Mozzo	3,45
	Cerchione	3,59

Tabella 1.6: Valutazioni complessive dei componenti analizzati per la scelta dello use-case

1.5 Procedimento per la valutazione del danno in tempo reale

Una volta selezionato il componente da manutenere, si definisce una procedura che consente di stimarne il danneggiamento in tempo reale, articolata nei seguenti passaggi:

- Acquisizione della variabile che definisce la condizione di funzionamento della molla, ovvero lo spostamento verticale. Infatti, a partire da questo si possono ottenere gli stati di tensione e deformazione del pezzo. Si precisa che la storia di carico è irregolare, in quanto dipende dalle sollecitazioni casuali trasmesse dalla strada al veicolo attraverso gli pneumatici;
- Pre-processing dei dati e scomposizione della storia di carico in cicli elementari. Il segnale ricevuto dal sensore è elaborato in Python implementando il metodo di conteggio rainflow a tre punti. Le forze agenti sulla molla impongono di effettuare delle analisi a fatica sul componente, dalle quali si ricavano i danni prodotti dai singoli cicli di sollecitazione. I modelli agli elementi finiti e i metodi di trasformazione delle storie di carico irregolari saranno descritti nel dettaglio più avanti;
- Valutazione dei danneggiamenti causati dai cicli di fatica elementari mediante tecniche di intelligenza artificiale. Nello specifico, si utilizza una rete neurale addestrata con un database virtuale contenente input e output di analisi agli elementi finiti. Il modello surrogato sostituisce le simulazioni FEM e permette di conoscere in tempo reale lo stato di salute dell'elemento elastico della sospensione anteriore;
- Post-processing dei dati;
- Applicazione della legge di Miner per calcolare il danno complessivo subito dal componente in un definito intervallo temporale;
- Revisione dei risultati.

Nella figura 1.7 è riassunto schematicamente il procedimento descritto poc'anzi.



Figura 1.7: Procedimento adottato per stimare in tempo reale il danneggiamento del componente

2 Trasmettitori

I trasmettitori sono dei dispositivi elettronici per l'acquisizione e la trasmissione di segnali. Il loro impiego consente di rilevare gli andamenti delle grandezze fisiche che caratterizzano un qualsiasi sistema meccanico, come ad esempio la pressione, la forza, la corrente, la coppia, lo spostamento, la tensione, la velocità o la deformazione. I trasmettitori sono formati dai seguenti elementi:

- Sensore: trasforma la variabile di processo V_p in una quantità misurabile per mezzo di una legge fisica. Ad esempio, per acquisire il valore di una forza si potrebbe usare un corpo deformabile, come una molla, un diaframma o una piastra, che converta la forza in uno spostamento;
- Trasduttore: trasforma la quantità fisica misurabile in un segnale elettrico o pneumatico;
- Condizionatore: converte il segnale di uscita in uno normalizzato, l'output è la variabile misurata V_m ; definisce la prestazione dinamica del trasmettitore.



Figura 2.1: Struttura di un trasmettitore

Si precisa che spesso il termine "sensore" è utilizzato con l'accezione più ampia di "trasmettitore".

2.1. Caratteristiche statiche degli strumenti di misura

Le caratteristiche statiche sono definite da numerosi parametri che legano uscita e ingressi di uno strumento di misura in condizioni statiche, ovvero quando i segnali di input sono costanti nel tempo o variano molto lentamente. In generale, i requisiti richiesti ad un trasmettitore ideale sono:

- Veridicità di misura: il segnale di output riproduce perfettamente quello di input, senza attenuazioni o distorsioni, sia a regime che durante i transitori;
- Caratteristica di input-output lineare;
- Insensibilità ai disturbi: il segnale di output è influenzato soltanto da quello di input;
- Assenza di errori dinamici;
- Dimensioni tali da garantire l'installazione dello strumento nella posizione più adatta all'acquisizione dei dati.

Il comportamento statico di uno strumento di misura è identificato mediante l'operazione di calibrazione o taratura, dalla quale è possibile ricavare i parametri statici elencati di seguito:

- Caratteristica statica guadagno statico;
- Campi di misura, di lavoro e di funzionamento;
- Ampiezza del campo di misura o span;
- Accuratezza o precisione;

- Ripetibilità;
- Linearità;
- Risoluzione;
- Isteresi;
- Bande morte o di non sensibilità.

La caratteristica statica esprime la relazione tra le grandezze di input e di output; se è lineare, il guadagno statico k rappresenta la pendenza della retta. Solitamente, le curve di calibrazione reali si discostano dalle rette che interpolano i dati sperimentali secondo il metodo dei minimi quadrati: in questi casi nasce il cosiddetto errore di linearità dello strumento di misura.



Figura 2.2: Caratteristica statica reale e ideale

Il campo di misura indica l'intervallo di variazione della grandezza acquisita in cui si garantiscono le prestazioni di accuratezza dello strumento, mentre quello di funzionamento rappresenta il massimo intervallo di variazione dell'input oltre il quale si rischia un danneggiamento permanente del trasmettitore. Il campo di lavoro esprime l'intervallo di utilizzo tipico dello strumento. Lo span è pari alla differenza tra gli estremi superiore e inferiore del campo di misura. Sovente, l'errore di linearità è espresso come percentuale riferita al fondo scala, ossia l'estremo superiore del campo di misura. La precisione di un trasmettitore indica l'errore compiuto nella rilevazione della variabile di output e di conseguenza nella stima del valore della grandezza di input. La taratura dello strumento consente di eliminare gli errori sistematici ma non quelli accidentali. La banda morta o di non sensibilità rappresenta un campo di variazione dell'ingresso per il quale il trasmettitore non genera variazioni apprezzabili dell'uscita. La risoluzione di uno strumento è pari alla minima variazione del segnale di output che è possibile apprezzare a fronte del cambiamento della grandezza di input. L'isteresi esprime la massima differenza tra i valori di uscita corrispondenti ad un medesimo segnale di ingresso, raggiunto con verso di variazione discorde, prima per valori crescenti e poi decrescenti.



Figura 2.3: Risoluzione di uno strumento di misura



Figura 2.4: Isteresi di uno strumento di misura

Infine, la ripetibilità evidenza la capacità dello strumento di riprodurre il medesimo output in seguito all'applicazione dello stesso segnale di input; fornisce un'indicazione riguardo la dispersione dei dati misurati attorno al valore medio. In alcuni casi, si potrebbe ottenere un'alta ripetibilità con una scarsa accuratezza. In figura 2.5 il simbolo "+" indica il valore stimato dal trasmettitore, mentre "x" quello esatto.



Figura 2.5: Ripetibilità e accuratezza di uno strumento di misura

2.2 Scelta del sensore

Per scegliere opportunamente il sensore adatto ad una specifica applicazione, è opportuno tenere in considerazione i seguenti parametri:

- Tipologia;
- Campo di misura e di lavoro;
- Accuratezza, ripetibilità e risoluzione desiderate;
- Caratteristica statica dinamica, linearità;
- Larghezza di banda e insensibilità al disturbo;
- Frequenza di acquisizione del segnale;
- Livello di affidabilità (Mean Time Between Failure);
- Costi di acquisto, installazione e manutenzione;
- Tempo di approvvigionamento;
- Dimensioni e ingombri;
- Condizioni dell'ambiente di lavoro;
- Montaggio e cablaggio;
- Materiale.

Le tipologie di trasmettitori disponibili sul mercato sono molteplici: resistivi, induttivi, capacitivi, ottici o elettro-meccanici. Nella tabella 2.1 si elencano i sensori tipicamente impiegati in ambito industriale per il rilevamento di dati. Inoltre, si specificano anche i settori di applicazione.

SENSORI TRADIZIONALI PER RILEVAMENTO DATI								
Nome	Tipologia	Grandezza misurata	Settore di applicazione					
Estensimetro	Resistivo	Deformazione	Industriale, automotive, civile, aerospace, biomedico					
Potenziometro lineare	Resistivo	Spostamenti e posizioni	Industriale, motorsport, automotive, robotica					
LVDT	Induttivo	Spostamenti e posizioni	Industriale, robotica, automotive, aerospace					
Pick up	Induttivo	Spostamenti e posizioni	Industriale, automotive					
Sensore ad effetto Hall	Magnetico	Spostamenti, posizioni, velocità	Industriale, automotive, elettronica					
Accelerometro	Elettro-meccanico	Accelerazione	Industriale, automotive, elettronica, aerospace, robotica, biomedico					
Piezoelettrico	Capacitivo	Forza, spostamento, pressione	Biomedico, industriale, aerospace, automotive, elettronica, acustica, ottica					
Cella di carico	Resistivo	Forza	Industriale, automotive, aerospace, alimentare, biomedico					
Torsiometro	Resistivo-induttivo	Coppia	Industriale, automotive, aerospace					
Resolver (analogico)	Induttivo	Spostamenti angolari	Industriale, automotive, motorsport, aerospace, robotica					
Encoder (digitale)	Ottico	Spostamenti angolari e lineari	Industriale, automotive, biomedico, aerospace, robotica					

Tabella 2.1: Principali sensori impiegati in ambito industriale

Tra i sensori indicati poc'anzi, solo una parte di essi rileva gli andamenti delle grandezze che, se opportunamente elaborati, permettono di ottenere il danno subito dalla molla della sospensione anteriore; i trasmettitori che misurano spostamenti, deformazioni o forze sono gli unici compatibili

con lo use-case scelto. Infatti, conoscendo il campo di deformazione si ricava lo stato tensionale del componente; analogamente, gli spostamenti o le forze acquisiti possono divenire gli input di modelli agli elementi finiti in grado di stimare il danneggiamento del componente. Dopo una lunga fase di ricerca, la lista dei sensori adatti a caratterizzare il comportamento della molla è visibile nella tabella 2.2. Le celle di carico non vengono riportate a causa dei costi elevati.

NOME SENSORE	TIPOLOGIA	GRANDEZZA MISURATA	соѕто	TEMPO DI SPEDIZIONE	INSTALLAZIONE IN VETTURA
Estensimetro	Resistivo	Deformazione	< 15 €	da 1 ad oltre 4 settimane	Difficile
Potenziometro lineare	Resistivo	Spostamenti, posizioni	> 200 €	fino a 4 settimane	Media difficoltà
Potenziometro a filo avvolto	Resistivo	Spostamenti, posizioni	> 300 €	fino a 4 settimane	Difficile
LVDT	Induttivo	Spostamenti, posizioni	> 200 €	da 2 ad oltre 4 settimane	Media difficoltà
Laser	Ottico	Spostamenti, posizioni	> 150 €	da 2 ad oltre 4 settimane	Media difficoltà
Ad ultrasuoni	Sonoro	Spostamenti, posizioni	> 100 €	da 2 ad oltre 4 settimane	Media difficoltà

Tabella 2.2: Sensori per misure di spostamento e deformazione

Gli estensimetri hanno dimensioni dell'ordine dei millimetri e dei costi contenuti; tuttavia, il montaggio, il cablaggio e l'elaborazione dei segnali acquisiti richiede personale specializzato.



Figura 2.6: Estensimetro a lamina metallica

I potenziometri lineari garantiscono una buona risoluzione e linearità, hanno dimensioni piuttosto contenute, corsa dello stelo fino a centinaia di millimetri e ampio range di temperatura di lavoro.



Figura 2.7: Potenziometro lineare

Gli LVDT (Linear Variable Differential Transformer) basano il loro funzionamento sulla variazione di mutua induttanza tra il circuito primario e i secondari. Quando il primario è alimentato con tensione alternata viene indotta una tensione anche nei due avvolgimenti secondari, proporzionale alla posizione assunta dal cilindro di materiale ferromagnetico. Il segnale in tensione di uscita viene rielaborato mediante un low pass filter e un demodulatore sensibile alla fase. Questi sensori sono estremamente robusti, possiedono una risoluzione infinita, un'ottima ripetibilità e una larghezza di banda elevata. Inoltre, sono in grado di sopportare alte temperature e condizioni ambientali estreme.



Figura 2.8: LVDT

Gli LVDT hanno dimensioni comparabili a quelli dei potenziometri lineari. L'uso di entrambi i trasmettitori è ritenuto inadeguato in quanto richiede un'installazione in vettura decisamente complessa. Al temine di questo processo di analisi, si decide di acquistare il sensore laser ILD1220-200 dell'azienda Micro-Epsilon; le motivazioni alla base di questa scelta sono:

- Costo di acquisto e manutenzione ridotti;
- Campo di misura compatibile con gli ingombri della sospensione anteriore;
- Ottima accuratezza, ripetibilità, risoluzione e linearità;
- Frequenza di acquisizione del segnale fino a 1 *kHz*;
- Dimensioni e peso contenuti;
- Facilità di installazione e cablaggio.



Figura 2.9: Sensore laser ILD1220-200

Questo trasmettitore è idoneo alla misurazione senza contatto di spostamenti e posizioni; il suo principio di funzionamento si basa sulla triangolazione ottica. La radiazione emessa dal laser a semiconduttore colpisce la superficie del bersaglio desiderato. Il fascio di luce riflesso viene raccolto da una lente e inviato ad un dispositivo elettronico chiamato CMOS (complementary metal-oxide semiconductor). A questo punto, un processore elabora i dati provenienti dal CMOS e calcola la

distanza dell'oggetto dal sensore; i segnali di output sono trasmessi mediante il protocollo di comunicazione seriale RS422. Il modulo di interfaccia IF2001/USB trasforma i segnali digitali RS422 in un pacchetto dati USB: collegando il convertitore ad un computer con un cavo USB, è possibile visualizzare l'andamento dello spostamento in funzione del tempo usufruendo del software open source "sensorTOOL" sviluppato dalla Micro-Epsilon.



Figura 2.10: Modulo di interfaccia IF2001/USB

Alcuni parametri del sensore, come la frequenza di acquisizione o i segnali di trigger, sono variabili tramite una Web Interface. Inoltre, il segnale di output può essere trasmesso in formato analogico o digitale.

					sensor TC	English
Connections 📀			Se	earch Res	sults (1)	
		optoNCDT I	LD1220		Rav	v Parameter View
Sensor group optoNCDT V			Parameters		Start Data	Acquisition
Sensor type			Port number:	COM3	Jan Data	Acquisition
optoNCDT ILD1220 V			Baud rate: Serial number controller	921600	Open Web:	site
Sensor		V	Software version:	001.062	0	
					Configure l	baudrate
Scan Options						
Show results of similar sensors, too						
Search serial interfaces						
Enable logging						
	- 11					
Load sensor protocol						
Ready						
→ C @ ① 127.00.14	51112 Serial number	18110036		110%	19 \$	
C C C	51112 Serial number Option 500 Measuring rang	18110036 ps 10.00mms		opto	···· © ☆ NCDT1220	
C C C C C C C C C C C C C C C C C C C	Serial number Option 600 Measuring rang	18110036 je 10.00mm		opto	···· ☺ ☆ NCDT1220 Save settings	
C C	Serial number Option 000 Measuring rang	18110036 pr 10 00mm	istance	opto	••• © & NCDT1220 Save settings Status	
→ • • •	Serial number Option 000 Measuring rang	18110036 je 10.00mm D 3.3	istance 331 mm	opto	···· © ☆ NCDT1220 Save settings Status	
C 1270010 C 127001 C 1270010	Sill2 Serial number Option 000 Measuring rang	18110036 je 10.00mm D 3.3	istance 331 mm	opto	···· © ☆ NCDT12200 Stave settings Status	
	Sill2 Serial number Option 000 Measuring rang	18110036 je 10.00mm D 3.3	istance 331 mm	opto	···· ♡ ☆ NCDT12200 bave settings Status ©	
	61112 Berial number Option 660 Measuring rang	18110036 (r 10.00mm) D 3.2 vare information	istance 331 mm	opto	···· ☺ ☆ NCDT12200	
	51112 Berlal number Option 600 Measuring rang	18 1160.36 p 10 00mm D 3.2 vare information	istance 831 mm	opto	···· ♥ ☆ NCDT12200 ₩ save settings Status ●	
Control software Control sof	51112 Berial number Option 500 Messuring rang Hardtv Descri	1815036 p 10 00mm D 3.2 vare information lption	istance 331 mm Nutue 8.0120	opto	··· © ☆ NCDT12200 Status •	
C 1270010 C 127001 C 1270010 C 127001 C 127	51112 Berial number Option 500 Messuring rang Hardw Descri Name Serial	18110036 pe 10 00mm 3.3 vare information liption number	istance 331 mm Value RU0720 10110006	opto	CDT1220 CDT1220 Status O	
C 127001 C 12700 C 12700 C 127001 C 12700 C 127001 C 127001 C 127001 C 127	Bertal number Option 600 Measuring rang Hardw Descri Name Serial Option	18110036 p 19 50mm 3.2 vare information iption	istance 331 mm Value 8.01220 1010026 0000	opto		
Control software Control sof	S1112 Bertal number Option 600 Messuring rang HardW Descri Name Serial Option Article	18116256 (a) 18 50mm (a) 18 50mm (b) 18 50mm (c) 18 5	istance 331 mm Value ILD 120 18110036 000 4100260	opto	CDT1220 CDT1220 Status Status O Status Status O Status Status O Status Status O Status Statu	
Control sensings Control sen	Silli? Serial number Option 600 Measuring rang Hardw Descri Name Serial Option Article Cable	Isinosa Isi	istance 331 mm Value ILD1320 10110036 000 410036 000 410036 Wee	opto	Status	
Control suthings Control sut	Still2 Bertal number Option 600 Measuing rang Hardw Descri Name Serial 1 Option Option Article Cable 1 Measu	19110236 pr 10 50mm 3.3 vare Information pption number head head hing sampo	istance 331 mm Natur ILD 120 18110036 000 4120260 Wite Vite Vite	opto	Status	
	S1112 Bertal number Option 600 Messuing rang Hardv Descri Name Sierial Option Article Cable Messus Version	19110036 pr 10 00mm a 10 00mm a 3 . 2 vare information ption head non single ng r	istance 331 mm Value RU0720 101005 000 4120300 Were 10.00mm 10.00mm 10.00mm	opto	Status	
Construction	S1112 Defail number Option 600 Messuring rang Hardw Descri Name Serta Option Article Cable Messur Version Boot-w	19110036 19110056 2010	istance 331 mm Value ILD 120 1010006 00 410000 00 10.00mm 001.006	opto	Status	
C 127001 C 12700 C 127001 C 12700 C 127001 C 12700 C 127001 C 12700 C 127	S1112 Bend number Option 500 Keesung rang Hardw Hardw Descri Name Senial Option Article Cable Measur Version Bootw Buildford	18110206 pe 18 20mm 3.5 ware information petion head number head nerson p	istance 331 mm Nature IL0120 101005 000 410006 Wee 10000 001008 543 903000	opto	Status	
C 127001 C	S1112 Banat number Option 000 Messuring rang Hardw Descri Name Serial Option Article Cable I Messis Version Bootw Buildig Timess	19110034 pr 10 00mm a 10 00mm a 2 3 2 vare information ption number head non singe in head non singe in prion head amp in ption	istance 331 mm Nature RU03200 1010005 000 41202000 Were 10.00mm 001.09 001.09 001.09 643 20190211_104540	opto	Status	
Configuration Configu	S1112 Benal number Option 000 Measuring rang Hardv Hardv Deser Serial Option Article Cable Cable Deser Versio Bootv Uversio Bootv Duutint Times	Isinoosi Isinoo	istance 331 mm KD1220 19110036 000 4102080 Wee 10.00mm 001.009 001.009 543 20190211_104540	opto	Status	

Figura 2.11: Interfaccia grafica del software "sensorTOOL"





Figura 2.12: Ingombri del sensore laser ILD1220-200

Nella tabella 2.3 si riportano le principali caratteristiche tecniche del laser ILD1220-200.

SCHEDA TECNICA SENSORE LASER ILD1220-200							
Campo di misura [mm]	60 ÷ 260						
Frequenza di acquisizione [kHz]	1						
Linearità [% FSO*]	< ± 0,10						
Ripetibilità [µm]	20						
Temperatura di funzionamento [°C]	0 ÷ 50						
Stabilità di temperatura [% FSO/K]	± 0,01						
Sorgente di luce	Semiconduttore laser (colore rosso)						
Dimensioni fascio laser [mm]	0,75 x 1,1						
Tensione di alimentazione [V]	11 ÷ 30						
Consumo di potenza [W]	< 2						
Interfaccia digitale	RS422 (16 bit)						
Classe di protezione (DIN-EN 60529)	IP65						
Materiale	Alluminio						
Massa [g]	30						
*FSO = Full Scale Output, pari alla differenza tra gli estremi del campo di misura							

Tabella 2.3: Caratteristiche tecniche sensore laser ILD1220-200

2.3 Schema di cablaggio del sensore laser ILD1220-200

Il trasmettitore selezionato è in grado di misurare lo spostamento verticale dell'elemento elastico della sospensione anteriore destra dell'auto di prova. Per semplificare il montaggio, il fascio laser viene indirizzato sullo pneumatico piuttosto che sul piattello inferiore di sostegno della molla. Infatti, la geometria del sistema meccanico considerato permette di approssimare il movimento verticale della ruota con quello della molla, soprattutto per basse velocità di avanzamento del veicolo. Il sensore è fissato all'interno del passaruota, ad una distanza di 136,68 mm dallo pneumatico. Le condizioni operative con cui sono condotti i test in vettura sono descritte nel seguente elenco:

- Marcia a velocità ridotta, inferiore a 25 km/h;
- Percorso per lo più rettilineo;
- Temperatura dell'aria pari a 28°C;
- Assenza di passeggeri e di bagagli;
- Spostamento del piattello superiore della sospensione trascurabile rispetto a quello inferiore;
- Spostamento verticale della ruota all'incirca uguale a quello del piattello di base della sospensione (collegamento rigido).



Figura 2.13: Installazione del sensore laser in vettura

Il circuito di acquisizione è formato dai seguenti componenti:

- Batteria dell'auto da 12 V: fornisce la tensione di alimentazione all'intero circuito ed è collegata alla presa dell'accendisigari;
- Spinotto per la presa dell'accendisigari: consente di prelevare la tensione di 12 V (generata dalla batteria) dall'abitacolo ed è connesso con lo step up voltage converter;
- Step up voltage converter 12 V 24 V: dispositivo elettronico che incrementa la tensione di ingresso fino ad un valore di uscita pari a 24 V. L'introduzione di tale elemento è necessaria poiché il modulo di interfaccia richiede una tensione di alimentazione di 24 V;
- Sensore laser ILD1220-200: è collegato alla morsettiera del convertitore con nove cavi e rileva il movimento dell'elemento elastico della sospensione;
- Modulo di interfaccia IF2001/USB: trasforma i segnali digitali del sensore in un pacchetto dati USB, permettendone la visualizzazione direttamente sullo schermo del computer. È connesso allo step up voltage converter con tre fili elettrici e al PC mediante un cavo USB;
- Computer portatile: il software sensorTOOL della Micro-Epsilon concede all'utente la possibilità di gestire i dati e di monitorare l'andamento nel tempo dello spostamento della molla.



Figura 2.14: Cablaggio del sensore laser ILD1220-200

3 Metodo degli elementi finiti

3.1 Introduzione alle analisi agli elementi finiti

Il FEM (Finite Element Method) è una delle tecniche numeriche più utilizzate in ambito ingegneristico per simulare il comportamento di sistemi fisici complessi in tempi contenuti; sebbene sia nato per rispondere all'esigenza di ricercare soluzioni approssimate di problemi di natura strutturale, oggi le analisi agli elementi finiti sono impiegate in svariati campi della fisica: elettromagnetismo, fluidodinamica, acustica, termodinamica.



Figura 3.1: Esempio di visualizzazione dei risultati di un'analisi FEM

Il FEM è lo strumento alla base della progettazione assistita dal calcolatore, in quanto consente di ottimizzare la geometria del componente analizzato, abbatterne il costo e garantire le corrette condizioni di sicurezza ed affidabilità nell'arco della sua vita operativa. Inoltre, lo sfruttamento delle elevate potenzialità di calcolo dei moderni elaboratori elettronici permette di lanciare sul mercato un prodotto maggiormente competitivo, qualitativo e funzionale.

Gli attuali codici commerciali presentano una suddivisione del processo di analisi in 3 step:

- 1) Pre-processing;
- 2) Soluzione;
- 3) Post-processing.

Nella fase di pre-processing l'utente costruisce il modello mediante i seguenti passi:

- Importazione della geometria;
- Modifica e correzione della geometria (qualora fosse necessaria);
- Scelta della tipologia di elemento;
- Costruzione della mesh 2D;

- Correzione della mesh 2D (qualora fosse necessaria);
- Creazione della mesh 3D;
- Definizione del materiale;
- Attribuzione del materiale agli elementi della mesh;
- Applicazione di carichi e vincoli.

Una volta generato il modello, il solutore del software FEM è in grado di risolvere le equazioni matematiche scritte nei nodi degli elementi ed ottenere una soluzione in ogni punto del dominio. Il grande pregio dei codici commerciali risiede proprio nella semplicità di utilizzo: le macro-operazioni effettuate dall'utente vengono automaticamente tradotte dall'elaboratore in un elevato numero di equazioni, organizzate secondo una struttura matriciale.

La terza ed ultima fase del processo, ovvero quella di post processing, è necessaria per valutare l'adeguatezza dei risultati ottenuti. Infatti, il modello del componente è inevitabilmente affetto da errori, a causa di molteplici fattori concomitanti:

- Incertezza sui dati di input;
- Ipotesi semplificative;
- Limiti del metodo numerico o di precisione della macchina;
- Errori dell'utente.

Pertanto, la strada corretta da intraprendere per la validazione del modello numerico è quella di una campagna sperimentale sul componente, al fine di confrontare gli output delle simulazioni con quelli del mondo reale. Nello schema seguente ΔG_i indica la discrepanza tra la grandezza i-esima attesa e quella ottenuta dal modello, mentre ε_i rappresenta la tolleranza accettabile sull'i-esima grandezza considerata.



Figura 3.2: Validazione di modelli FEM
3.2 Il problema elastico tridimensionale

La teoria lineare dell'elasticità rappresenta una branca della meccanica dei continui che studia il comportamento di corpi solidi elastici soggetti a carichi esterni.

Il problema elastico 3D è stato studiato a partire dal 1800 ed il contributo di numerosi ingegneri e matematici del tempo ha permesso di costruire le nozioni teoriche che, successivamente, hanno condotto allo sviluppo di analisi strutturali sempre più complesse (come il metodo FEM).

Di seguito si riportano le equazioni ed i concetti fondamentali della meccanica del continuo, estremamente utili per comprendere le relazioni matematiche alla base della teoria degli elementi finiti.

Dato un corpo continuo in un sistema di riferimento cartesiano x, y, z, si definisce il vettore tensione in un generico punto P, che giace su una superficie ΔA di normale $\{n\}$ come:

$$\{t\} = \{t\}(\{n\}, P) = \lim_{\Delta A \to 0} \frac{\{\Delta F\}}{\Delta A}$$

La grandezza { ΔF } rappresenta la forza agente dall'esterno sul continuo per mezzo della superficie ΔA . Il vettore tensione ha 3 componenti, ovvero due tensioni tangenziali τ ed una tensione normale σ ; inoltre, esso dipende sia dall'orientazione della superficie ΔA che dalla posizione di P.



Figura 3.3: Vettore tensione in P

Lo stato di tensione nel punto P è noto qualora si conoscano le tensioni agenti su tre piani perpendicolari tra loro e passanti per il punto P. Per verificarlo è sufficiente costruire attorno a P il tetraedro elementare di Cauchy e scrivere l'equazione di equilibrio alla traslazione.



Figura 3.4: Equilibrio dei vettori tensione

Con riferimento alla figura 3.4 si ottiene:

$$\{t\} \cdot dA - \{t_x\} \cdot dA_x - \{t_y\} \cdot dA_y - \{t_z\} \cdot dA_z + \{\phi_v\} \cdot dV = 0$$
$$dA_x = dA \cdot n_x$$
$$dA_y = dA \cdot n_y$$
$$dA_z = dA \cdot n_z$$

Il termine $\{\phi_v\}$ indica le forze per unità di volume, mentre con dA_x si intende la superficie infinitesima ortogonale all'asse x (ragionamento analogo per dA_y e dA_z). Trascurando gli infinitesimi di ordine superiore si può scrivere in forma compatta il legame tra il vettore tensione $\{t\}$ e il tensore delle tensioni $[\sigma]$.

$$\{t\} = [\sigma]\{n\}$$
$$[\sigma] = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix}$$

Il problema elastico tridimensionale è descritto in ogni punto di una struttura generica mediante un sistema di equazioni differenziali alle derivate parziali e algebriche:

- Equazioni differenziali indefinite di equilibrio;
- Equazioni differenziali di congruenza;
- Leggi costitutive del materiale.

3.2.1 Equazioni differenziali di equilibrio

Per ricavare le equazioni di equilibrio si consideri una porzione infinitesima di continuo in un sistema di riferimento cartesiano x, y, z (vedi figura 3.5), di volume pari a:

$$d^3V = dx \cdot dy \cdot dz$$



Figura 3.5: Campo elementare di tensione

L'equazione di equilibrio lungo la direzione x è:

$$-\left(\sigma_{xx} - \frac{1}{2} \cdot \frac{\partial \sigma_{xx}}{\partial x} \cdot dx\right) dy \cdot dz + \left(\sigma_{xx} + \frac{1}{2} \cdot \frac{\partial \sigma_{xx}}{\partial x} \cdot dx\right) dy \cdot dz$$
$$-\left(\sigma_{yx} - \frac{1}{2} \cdot \frac{\partial \sigma_{yx}}{\partial y} \cdot dy\right) dx \cdot dz + \left(\sigma_{yx} + \frac{1}{2} \cdot \frac{\partial \sigma_{yx}}{\partial y} \cdot dy\right) dx \cdot dz$$
$$-\left(\sigma_{zx} - \frac{1}{2} \cdot \frac{\partial \sigma_{zx}}{\partial z} \cdot dz\right) dx \cdot dy + \left(\sigma_{zx} + \frac{1}{2} \cdot \frac{\partial \sigma_{zx}}{\partial z} \cdot dz\right) dx \cdot dy$$
$$+\phi_x \cdot dx \cdot dy \cdot dz = 0$$

Semplificando l'equazione si ricava:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} + \phi_x = 0$$

Lungo le direzioni y, z il procedimento è analogo. Per scrivere le equazioni differenziali indefinite di **equilibrio** in forma compatta si effettua un cambiamento nella notazione degli assi cartesiani x, y, z, che vengono rinominati come x_1, x_2, x_3 .

$$\sum_{i=1}^{3} \frac{\partial \sigma_{ij}}{\partial x_i} + \phi_j = 0 \qquad (j = 1, 2, 3)$$

3.2.2 Equazioni differenziali di congruenza

La seconda tipologia di equazioni differenziali che descrivono il problema elastico sono quelle di congruenza, che legano il campo degli spostamenti e il campo delle deformazioni. Con riferimento alla figura 3.6, si consideri un corpo solido continuo sul quale agisce un sistema di forze esterne che inducono uno stato di deformazione:

- I punti P e Q distano $\{dX\}$ nella configurazione indeformata;
- I punti p e q distano $\{dx\}$ nella configurazione deformata;
- Il punto P ha coordinate $\{X\}^T = \{X_1, X_2, X_3\}^T$ nella configurazione indeformata;
- Il punto p ha coordinate $\{x\}^T = \{x_1, x_2, x_3\}^T$ nella configurazione deformata;
- {*u*} rappresenta lo spostamento del punto P dalla configurazione indeformata a quella deformata;
- $\{u\} + \{du\}$ rappresenta lo spostamento del punto Q dalla configurazione indeformata a quella deformata.

In generale, il segmento \overline{PQ} può trasformarsi in \overline{pq} per effetto di una roto-traslazione, una dilatazione (cambiamento di volume) ed una distorsione (cambiamento di forma).

La dilatazione non modifica la geometria del continuo, ma produce solamente variazioni di lunghezza dei segmenti considerati (deformazioni ε_x , ε_y ed ε_z). Al contrario, la distorsione provoca degli scorrimenti angolari a causa della variazione degli angoli formati tra segmenti (scorrimenti angolari γ_{xy} , $\gamma_{yz} \in \gamma_{zx}$). Con l'ipotesi di piccoli spostamenti le derivate che compaiono nelle equazioni di congruenza sono del primo ordine.

In un continuo 3D si ottengono sei equazioni di congruenza:



Figura 3.6: Deformazione in un continuo

Lo spostamento infinitesimo $\{du\}$ ha un legame differenziale con il campo degli spostamenti:

$$\{du\} = [J] \{dX\} = \begin{bmatrix} \frac{\partial u_x}{\partial X} & \frac{\partial u_x}{\partial Y} & \frac{\partial u_x}{\partial Z} \\ \frac{\partial u_y}{\partial X} & \frac{\partial u_y}{\partial Y} & \frac{\partial u_y}{\partial Z} \\ \frac{\partial u_z}{\partial X} & \frac{\partial u_z}{\partial Y} & \frac{\partial u_z}{\partial Z} \end{bmatrix} \begin{pmatrix} dX \\ dY \\ dZ \end{pmatrix}$$

La matrice Jacobiana [*J*] è denominata gradiente dello spostamento e si può scomporre in due tensori:

- Il tensore della rotazione rigida [W];
- Il tensore della deformazione [*E*].

Il tensore [W] esprime la parte dello spostamento infinitesimo $\{du\}$ data dal moto di rotazione rigida, mentre il tensore [E] rappresenta la parte dello spostamento infinitesimo $\{du\}$ data dal

moto di dilatazione e scorrimento (moto di deformazione). In termini matriciali:

$$[J] = [W] + [E]$$
$$[W] = \frac{1}{2} [J] - \frac{1}{2} [J]^{T}$$
$$[E] = \frac{1}{2} [J] + \frac{1}{2} [J]^{T}$$

Il tensore [E] contiene le dilatazioni lungo la diagonale principale, mentre i termini fuori diagonale sono pari a metà degli scorrimenti tra i vari assi cartesiani.

$$[E] = \begin{bmatrix} \varepsilon_{xx} & \frac{1}{2}\gamma_{xy} & \frac{1}{2}\gamma_{xz} \\ \frac{1}{2}\gamma_{xy} & \varepsilon_{yy} & \frac{1}{2}\gamma_{yz} \\ \frac{1}{2}\gamma_{xz} & \frac{1}{2}\gamma_{yz} & \varepsilon_{zz} \end{bmatrix}$$

3.2.3 Leggi costitutive del materiale

Le ultime relazioni matematiche necessarie alla descrizione del problema elastico 3D sono le leggi costitutive del materiale, che esprimono il legame tra le tensioni e le deformazioni presenti nel continuo.

Per un solido in campo lineare elastico vale la seguente espressione tensoriale (legge di Hook):

$$\{\sigma\} = [E] \{\varepsilon\}$$

Il simbolo [E] è noto come tensore di elasticità. La simmetria dei tensori della tensione e della deformazione permette di caratterizzare il tensore [E] mediante 36 componenti. Inoltre, con l'ipotesi di materiale omogeneo ed isotropo il tensore di elasticità è funzione di sole 2 costanti indipendenti, ossia il coefficiente di Poisson ν ed il modulo di Young E.

La legge di Hook scritta in notazione compatta e matriciale dimostra che il legame tra tensioni e deformazioni è espresso mediante 6 equazioni algebriche:

$$\begin{aligned} \varepsilon_{ii} &= \frac{1}{E} \left[\sigma_{ii} - \nu (\sigma_{jj} + \sigma_{kk}) \right] \\ \gamma_{ij} &= \frac{2(1+\nu)}{E} \tau_{ij} = \frac{k}{E} \tau_{ij} \\ \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \\ \end{pmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & k & 0 & 0 \\ 0 & 0 & 0 & 0 & k & 0 \\ 0 & 0 & 0 & 0 & 0 & k \\ 0 & 0 & 0 & 0 & 0 & k \\ \end{bmatrix} \begin{cases} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xz} \\ \end{array}$$

3.3 Dalla meccanica del continuo agli elementi finiti

Il problema elastico tridimensionale si basa su un sistema di equazioni differenziali alle derivate parziali. Nella seconda metà del 1800 Kirchhoff dimostra per assurdo che la soluzione esiste sempre ed è unica: assegnato un campo di forze e deformazioni ad un continuo ottenne due configurazioni energetiche distinte parimenti valide, condizione fisicamente non realizzabile. Tuttavia, la soluzione analitica in forma chiusa non è generalmente calcolabile; per tale ragione è indispensabile un approccio di tipo CAE (Computer-Aided Engineering).

La transizione dalla meccanica del continuo alle simulazioni numeriche è permessa dal metodo degli elementi finiti, ovvero una soluzione numerica in forma integrale su un dominio finito in cui un problema fisico-matematico è descritto da equazioni differenziali alle derivate parziali. Di seguito si riporta uno schema riassuntivo degli step da effettuare per trasformare una porzione di continuo in un insieme di elementi finiti.



Figura 3.7: Metodo degli elementi finiti

Nell'ambito del calcolo strutturale statico la scrittura del funzionale, ossia un'equazione integrale su un dominio finito, può essere ricavata con un approccio di tipo energetico mediante il principio dei lavori virtuali e con l'ausilio delle equazioni di equilibrio e congruenza.

Le shape functions definiscono una legge di discretizzazione del dominio: esse sono funzioni di interpolazione del campo di spostamenti e permettono di valutare lo spostamento di un generico punto interno all'elemento a partire da quelli nodali.

Al termine della procedura, il problema differenziale di partenza diviene equivalente ad uno matriciale risolubile mediante un elaboratore elettronico. A questo punto, un'analisi statica di un sistema generico è ridotta all'assemblaggio della matrice di rigidezza della struttura.

3.3.1 Il principio dei lavori virtuali

Dato un continuo infinitesimo di volume dx_1, dx_2, dx_3 ed un campo di forze e tensioni equilibrato applicando un campo di spostamenti e deformazioni virtuali il lavoro virtuale interno (dovuto alle tensioni ed alle deformazioni) è uguale al lavoro virtuale esterno (dovuto alle forze ed agli spostamenti).



Figura 3.8: Stato di tensione in un infinitesimo 2D



Figura 3.9: Spostamenti virtuali in un infinitesimo 2D

Per valutare il lavoro virtuale è sufficiente moltiplicare tutte le componenti reali di tensione per le rispettive componenti degli spostamenti virtuali e sommare i singoli risultati.

Il lavoro virtuale per un elemento bidimensionale infinitesimo lungo l'asse x_i perpendicolare alle facce di area dA_i è pari:

$$\delta L_{x_{i,v}} = \sum_{k=1}^{3} \left[-\sigma_{ik} \delta u_k dA_i + \left(\sigma_{ik} + \frac{\partial \sigma_{ik}}{\partial x_i} dx_i \right) \left(\delta u_k + \frac{\partial \delta u_k}{\partial x_i} dx_i \right) dA_i \right]$$

Il lavoro virtuale per un cubetto elementare infinitesimo considerando tutte le direzioni vale:

$$\delta L_{\nu} = \sum_{i=1}^{3} \sum_{k=1}^{3} \left(\frac{\partial \sigma_{ik}}{\partial x_i} \delta u_k + \sigma_{ik} \frac{\partial \delta u_k}{\partial x_i} \right) dV$$

Ricordando il legame tra deformazioni e spostamenti espresso dalle equazioni di congruenza, il secondo termine dell'espressione del lavoro virtuale può essere riscritto come prodotto scalare tra il vettore delle deformazioni virtuali e quello delle tensioni:

Analogamente, sfruttando l'equazione indefinita di equilibrio il primo termine dell'espressione del lavoro virtuale è esprimibile mediante prodotto scalare:

$$\sum_{i=1}^{3} \sum_{k=1}^{3} \frac{\partial \sigma_{ik}}{\partial x_i} \delta u_k = \sum_{k=1}^{3} \left(\frac{\partial \sigma_{1k}}{\partial x_1} + \frac{\partial \sigma_{2k}}{\partial x_2} + \frac{\partial \sigma_{3k}}{\partial x_3} \right) \delta u_k = -\sum_{k=1}^{3} \phi_k \delta u_k = -\{\phi\}^T \{\delta u\} = -\{\delta u\}^T \{\phi\}$$

Il lavoro virtuale ottenuto per il cubetto infinitesimo può essere integrato sull'interno volume del continuo:

$$\int_{V} \{\delta\varepsilon\}^{T}\{\sigma\} dV - \int_{V} \{\delta u\}^{T}\{\phi\} dV$$

In definitiva, tenendo in considerazione i contributi delle forze superficiali $\{t\}$ l'equazione dei lavori virtuali diventa:

$$\int_{V} \{\delta\varepsilon\}^{T}\{\sigma\} dV - \int_{V} \{\delta u\}^{T}\{\phi\} dV = \int_{A} \{\delta u\}^{T}\{t\} dA$$
$$\int_{V} \{\delta\varepsilon\}^{T}\{\sigma\} dV = \int_{V} \{\delta u\}^{T}\{\phi\} dV + \int_{A} \{\delta u\}^{T}\{t\} dA$$

 $\delta L_{virtuale,interno} = \delta L_{virtuale,esterno}$

Il funzionale integrale soddisfa il sistema di equazioni differenziali del problema elastico tridimensionale statico ed esprime l'energia accumulata nella struttura meccanica in una generica configurazione.

3.3.2 Il problema elastico in forma matriciale

Una volta scritto il funzionale integrale è indispensabile selezionare un'opportuna modalità di discretizzazione del continuo con elementi finiti.

Di seguito si riporta la formulazione a spostamenti assegnati, largamente impiegata nei modelli strutturali:

$$\{u\} = [n]\{s\}$$

- {*u*}: vettore degli spostamenti dei punti interni all'elemento;
- {*s*}: vettore degli spostamenti nodali;
- [*n*]: matrice delle funzioni di forma.

In un modello FEM le equazioni del problema fisico sono scritte e risolte nei nodi degli elementi, pertanto è cruciale utilizzare delle funzioni di interpolazione per estendere la soluzione a tutti i punti interni agli elementi.



Figura 3.10: Generico punto P interno ad un elemento piano a 4 nodi

Dopo aver ottenuto gli spostamenti della struttura analizzata in tutto il dominio è possibile conoscere tensioni e deformazioni attraverso le relazioni:

$$\{\varepsilon\} = [\partial]\{u\} = [\partial][n]\{s\} = [b]\{s\}$$

$$\{\sigma\} = [E](\{\varepsilon\} - \{\varepsilon_0\}) + \{\sigma_0\}$$

- [b]: matrice delle derivate delle funzioni di forma;
- $\{\sigma_0\}$: stato di tensione preesistente;
- $\{\varepsilon_0\}$: stato di deformazione preesistente.

Si sottolinea che l'assegnazione di un campo di spostamenti obbliga tutti i punti interni ad un generico elemento a seguire il movimento dei nodi, introducendo un vincolo nel problema fisico che nella realtà non è presente. La comparsa di tale errore conduce alla modifica dell'equazione di equilibrio, nella quale si considera un nuovo termine, detto residuo, che garantisce la validità dell'equazione in ogni punto della struttura.

Equazione di equilibrio modificata:

$$\sum_{i=1}^{3} \frac{\partial \sigma_{ij}}{\partial x_i} + \phi_j = \rho_j \qquad (j = 1, 2, 3)$$

La grandezza ρ_j indica la componente della forza di volume residua nella direzione j - esima. Il residuo non è mai nullo in un modello agli elementi finiti. Infatti, il termine ρ_j tenderebbe a zero solo nel caso in cui il numero dei nodi fosse infinito. Se ciò fosse realizzabile, la soluzione del modello sarebbe esatta.

L'introduzione delle shape functions consente di esprimere tensioni, deformazioni e spostamenti presenti nell'equazione dei lavori virtuali in funzione delle matrici [b] ed [n].

Di seguito si riportano il risultato finale ed il significato della simbologia utilizzata.

Equazione dei lavori virtuali:

$$\int_{V} [b]^{T}[E][b] dV\{s\} = \int_{A} [n]^{T}\{\tilde{t}\} dA - \int_{V} [n]^{T}\{\rho\} dV + \int_{A} [n]^{T}\{t_{n}\} dA + \int_{V} [n]^{T}\{\phi\} dV$$

In termini matriciali:

$$[K]{s} = {f} + {f_e}_{t_n} + {f_e}_{\phi}$$

- $\{\tilde{t}\}$, forze di superficie incognite, in termini di carichi generano delle reazioni vincolari;
- $\{t_n\}$, forze di superficie note, si traducono in carichi noti;
- $\{f\} = \int_A [n]^T \{\tilde{t}\} dA \int_V [n]^T \{\rho\} dV$, termine di forza contenente il residuo e le forze di superficie incognite;
- ${f_e}_{t_n} = \int_A [n]^T {t_n} dA$, carichi nodali equivalenti di superficie;
- ${f_e}_{\phi} = \int_V [n]^T {\phi} dV$, carichi nodali equivalenti di volume;
- [*K*], matrice di rigidezza.

Il problema differenziale di partenza si traduce in uno matriciale equivalente che può essere implementato a livello software nei programmi di calcolo commerciali agli elementi finiti.

3.4 Convergenza nei metodi FEM

Un modello agli elementi finiti conduce ad una soluzione approssimata dei problemi fisici reali a causa di errori intrinseci nel metodo. Le approssimazioni di un modello FEM sono causate da:

- Discretizzazione del continuo con conseguente nascita del residuo;
- Distorsione degli elementi, dipende dalla complessità geometrica del componente oggetto di analisi;
- Integrazione numerica della matrice di rigidezza, in quanto l'integrazione analitica è possibile solamente per elementi triangolari;
- Troncamento numerico.

Visionando l'elenco appena riportato, si deduce che la scelta delle funzioni di forma e del tipo di elemento è particolarmente influente sui risultati delle analisi.

Solitamente, le shape functions sono espresse mediante polinomi in quanto sono funzioni derivabili e di facile gestione. Il grado del polinomio interpolante dipende dal numero di nodi e dai gradi di libertà per ogni nodo impiegati per descrivere l'elemento.

Gli unici rimedi per giungere a convergenza sulla soluzione esatta sono incrementare il numero di elementi o il grado della funzione di forma. Quest'ultima viene definita conforme se soddisfa determinate condizioni che assicurano la convergenza del modello al diminuire della dimensione della mesh.



Figura 3.11: Proprietà delle funzioni di forma

Gli elementi con un grado della funzione interpolante maggiore permettono di giungere a convergenza più rapidamente rispetto a quelli con shape function di grado inferiore, a parità di numero di elementi della mesh. Inoltre, essi riescono a descrivere in modo ottimale gradienti e concentrazioni di tensione e deformazione.

3.4.1 Tipologie di elementi in HyperMesh



Figura 3.12: Elementi 2D in HyperMesh

In HyperMesh è possibile scegliere tra diverse tipologie di elementi per costruire la mesh 2D o quella 3D. Nel caso di mesh bidimensionali si utilizzano elementi "Tria" oppure "Quad".

La differenza, oltre che nel numero di nodi dell'elemento, risiede nel grado della funzione interpolante il campo di spostamenti.



Figura 3.13: Elementi piani a 3 e 4 nodi

$\int u = \alpha_1 + \alpha_2 x + \alpha_3 y$	$\int u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x y$
$v = \alpha_4 + \alpha_5 x + \alpha_6 y$	$v = \alpha_5 + \alpha_6 x + \alpha_7 y + \alpha_8 x y$
$\left\{ \varepsilon_{xx} = \alpha_2 \right\}$	$\left\{ \varepsilon_{xx} = \alpha_2 + \alpha_4 y \right.$
$\varepsilon_{yy} = \alpha_6$	$\varepsilon_{yy} = \alpha_7 + \alpha_8 x$
$(\gamma_{xy} = \alpha_3 + \alpha_5)$	$(\gamma_{xy} = \alpha_3 + \alpha_6 + \alpha_4 x + \alpha_8 y)$

Un elemento piano a 3 nodi non è adatto a rilevare concentrazioni di tensione in quanto lo stato di deformazione è costante in tutto l'elemento.

Viceversa, per un elemento a 4 nodi il campo di deformazione è una funzione lineare in x, y; un elemento piano ad 8 nodi è caratterizzato da un campo di deformazione in cui compaiono anche i termini di secondo grado x^2, y^2 .

Per tali ragioni, gli elementi Tria a 3 nodi e quelli Quad a 4 nodi sono definiti lineari, mentre gli elementi Tria a 6 nodi e Quad ad 8 nodi sono detti parabolici.



Figura 3.14: Elementi 3D in HyperMesh

Per mesh tridimensionali sono disponibili quattro tipologie diverse di elementi con campi di deformazione espressi mediante funzioni lineari o paraboliche. Analogamente al caso 2D, gli elementi con funzioni di forma di grado maggiore ed un numero superiore di nodi consentono una migliore analisi nei problemi strutturali in cui è necessario catturare variazioni considerevoli di tensione in aree circoscritte del componente.

A titolo di esempio, si riporta la funzione di forma per un elemento 3D "Linear Hex 8", che possiede 8 nodi e 3 gradi di libertà traslazionali per ogni nodo.

$$\begin{cases} u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z + \alpha_5 xy + \alpha_6 yz + \alpha_7 zx + \alpha_8 xyz \\ v = \alpha_9 + \alpha_{10} x + \alpha_{11} y + \alpha_{12} z + \alpha_{13} xy + \alpha_{14} yz + \alpha_{15} zx + \alpha_{16} xyz \\ w = \alpha_{17} + \alpha_{18} x + \alpha_{19} y + \alpha_{20} z + \alpha_{21} xy + \alpha_{22} yz + \alpha_{23} zx + \alpha_{24} xyz \end{cases}$$

4 Molla della sospensione anteriore: analisi FEM statica

4.1 Modello geometrico

Il case study selezionato per lo sviluppo di una metodologia di manutenzione predittiva riguarda il danneggiamento subito da una molla della sospensione anteriore di un'auto a causa di carichi affaticanti. Tuttavia, è necessario effettuare un'analisi statica preliminare sul componente. Quando la molla viene installata all'interno della sospensione subisce una deformazione per effetto del peso della vettura e di eventuali precarichi di regolazione.

Pertanto, è doveroso valutare lo stato tensionale nel pezzo in condizioni statiche, prima che le sollecitazioni stradali producano uno spettro di carico random affaticante.

Ovviamente, per poter creare un modello agli elementi finiti è indispensabile importare nel software di simulazione il CAD del componente. Di conseguenza, si è realizzata una geometria semplificata della sospensione anteriore della vettura messa a disposizione dall'azienda (la medesima auto è stata sensorizzata per acquisire i dati di spostamento della molla).

Nonostante i disegni della sospensione oggetto di studio fossero disponibili, si sono ricostruite in NX (software di modellazione 3D) solamente le parti di interesse dell'assieme, tralasciando i dettagli geometrici non necessari ai fini dell'analisi. Infatti, è bene rammentare che il componente oggetto di studio è la molla e non si intende caratterizzare il comportamento della sospensione completa.

Nella figura seguente è mostrata la geometria della sospensione.



Figura 4.1: a) Modello CAD completo ; b) Dettaglio molla sospensione

In tabella 4.1 si riportano le caratteristiche geometriche dell'elemento elastico nella configurazione indeformata, ossia antecedente al montaggio.

Dimensioni molla	a indeformata
Diametro filo [mm]	11,25
Diametro esterno 1 [mm]	122,8
Diametro esterno 2 [mm]	137,6
Lunghezza [mm]	365,45
Passo spire centrali [mm]	117,1
Angolo inclinazione spire [°]	7
Numero spire	4



Tabella 4.1: Parametri geometrici molla

Figura 4.2: Quote molla in configurazione indeformata

Una volta montata in vettura, la molla subisce una variazione di lunghezza dovuta al carico esercitato dalle masse sospese. Per quantificarla, si sono visionati i disegni tecnici dell'auto di prova in cui sono rappresentate le dimensioni dei componenti della sospensione anteriore dopo l'installazione. Si riportano la lunghezza della molla nella configurazione deformata e l'accorciamento subito rispetto alla condizione iniziale:

$$L_{def} = 245,4 mm$$

$$\Delta z = L_{ind} - L_{def} = 120,05 mm$$

4.2 Proprietà del materiale del componente

Le molle sono elementi meccanici molto utilizzati negli organi di macchine, nei trasduttori, nelle sospensioni dei mezzi di trasporto ed in generale in qualsiasi applicazione dove sia richiesto un accumulo di energia potenziale.

Le molle sono progettate per lavorare in campo elastico: per tale ragione, hanno delle tensioni di snervamento e rottura molto più elevate rispetto ai componenti meccanici realizzati con i classici acciai strutturali. Inoltre, devono assicurare un buon comportamento a fatica.

Per garantirne la funzionalità e le caratteristiche citate, gli acciai per molle sono sottoposti ad una vasta gamma di trattamenti termici, come ad esempio:

- Tempra;
- Rinvenimento;
- Ricottura;
- Normalizzazione;
- Pallinatura;
- Cementazione;
- Nitrurazione.

Il materiale che costituisce la molla della sospensione anteriore oggetto dell'analisi è l'acciaio 55 SiCr 6-3. Di seguito se ne riportano le proprietà, la curva $\sigma - \varepsilon$ ed alcune possibilità di impiego.

	Tensione di snervamento [MPa]	1750
Dronzietà massanishe	Tensione di rottura [MPa]	2100
Proprieta meccaniche	Allungamento a rottura [%]	≥5
	Tensione limite di fatica [Mpa]	945,1
	Densità [kg/m^3]	7800
Dronzietà fisishe	Modulo di Young [MPa]	206000
Proprieta fisicile	Modulo di elasticità tangenziale [MPa]	80000
	Coefficiente di Poisson	0,288
	C [%]	0,5÷0,6
	Si [%]	1,2÷1,6
	S [%]	≤0,025
Composizione chimica	P [%]	≤0,03
	Mn [%]	0,5÷0,9
	Cu [%]	≤0,12
	Cr [%]	0,5÷0,8

Tabella 4.2: Proprietà dell'acciaio 55 SiCr 6-3



Figura 4.3: Curva $\sigma - \varepsilon$ dell'acciaio 55 SiCr 6-3

55 SiCr 6-3 Heat Treatment Regime						
Annealing	Quenching	Tempering	Normalizing	Q&T		
\checkmark	\checkmark	\checkmark	\checkmark	\checkmark		

	55 SiCr 6-3 Range of products								
Product type	Products	Dimension	Processes	Deliver Status					
Plates / Sheets	Plates / Sheets	0.08-200mm(T)*W*L	Forging, hot rolling and cold rolling	Annealed, Solution and Aging, Q+T, ACID- WASHED, Shot Blasting					
Steel Bar	Round Bar, Flat Bar, Square Bar	Φ8-1200mm*L	Forging, hot rolling and cold rolling, Cast	Black, Rough Turning, Shot Blasting,					
Coil / Strip	Steel Coil /Steel Strip	0.03-16.0x1200mm	Cold-Rolled & Hot- Rolled	Annealed, Solution and Aging, Q+T, ACID- WASHED, Shot Blasting					
Pipes / Tubes	Seamless Pipes/Tubes, Welded Pipes/Tubes	OD:6-219mm x WT:0.5-20.0mm	Hot extrusion, Cold Drawn, Welded	Annealed, Solution and Aging, Q+T, ACID- WASHED					

Tabella 4.3: Possibili trattamenti termici ed impieghi del materiale 55 SiCr 6-3

4.3 Modello agli elementi finiti in HyperMesh

Il modello agli elementi finiti della sospensione è generato con l'ausilio del software Altair HyperMesh, la cui interfaccia grafica è visibile nelle immagini riportare di seguito.



Figura 4.4: Interfaccia grafica del software Altair HyperMesh

La geometria dell'assieme è presente nella "Graphics Area"; nella "Tab Area" laterale è disponibile una lista riassuntiva delle entità create per costruire il modello agli elementi finiti, come ad esempio la mesh, il materiale utilizzato, i carichi ed i vincoli imposti. In basso, il "Main menu" consente di effettuare una vasta gamma di operazioni necessarie alla definizione del modello. La prima fase del processo di analisi riguarda il pre-processing, in cui si genera il modello FEM del componente. Tutti gli step fondamentali di questa fase, dall'importazione della geometria del pezzo alla definizione di carichi e vincoli, sono descritti nello schema successivo.



Figura 4.5: Costruzione del modello FEM

Spesso, le geometrie che vengono importante all'interno dei software FEM presentano imperfezioni da correggere oppure particolarità geometriche che potrebbero compromettere la qualità dell'analisi. Pertanto, è sempre consigliato interrogarsi sull'esigenza o meno di compiere defeaturing o cleanup della geometria prima di generare la mesh. In particolare, con il termine defeaturing si intende la semplificazione delle forme del componente mediante la rimozione di dettagli geometrici come piccoli fori, raccordi o smussi. Da un lato, tale accorgimento evita che il tempo computazionale e la complessità dell'analisi incrementino eccessivamente. Dall'altro, permette di trascurare delle parti del pezzo che non sono considerate particolarmente influenti per l'analisi.

L'espressione cleanup si riferisce alla correzione di difetti geometrici del componente, come ad esempio superfici sovrapposte o mancanti. Nel nostro caso, il defeaturing ed il cleanup non sono necessari poiché il modello CAD è stato generato con l'accortezza di essere esente da imperfezioni ed elementi geometrici superflui.

4.3.1 Costruzione mesh 2D e 3D

Si ricorda che il componente della sospensione che si intende esaminare nello specifico è la molla. Pertanto, si riportano gli studi dettagliati riguardanti la mesh bidimensionale e tridimensionale del solo elemento elastico, tralasciando la trattazione degli altri pezzi dell'assieme in quanto non sono oggetto di analisi. Nel paragrafo dedicato alla convergenza del modello saranno approfonditi gli aspetti legati all'influenza della dimensione e del numero di elementi sui risultati delle simulazioni. In questa sezione si indicano i parametri e le modalità di costruzione della mesh. La mesh 2D, generata soltanto sulla superficie circolare superiore della molla, possiede i seguenti parametri:

- Tipologia di elemento;
- *D* [*mm*]: dimensione del lato dell'elemento;
- *N_{base}*: numero di elementi presenti sulla semicirconferenza.

Nella sezione "Automesh" di HyperMesh è possibile impostare i parametri della mesh; nel nostro modello, gli elementi impiegati per la mesh bidimensionale sono i "Quad" lineari a 4 nodi.



Figura 4.6: Creazione della mesh 2D in HyperMesh

Per la realizzazione della mesh tridimensionale si utilizza la funzione "Line drag" del comando "Solid map" di HyperMesh, che permette di estrudere una mesh 2D lungo una o più linee selezionate dall'utente. In questo caso, è indispensabile specificare anche la dimensione dell'elemento in direzione circonferenziale.

Inoltre, il software consente di definire una legge di disposizione spaziale degli elementi: in tal modo, è possibile infittire localmente la mesh per catturare dei gradienti significativi di tensione senza incrementare a dismisura il numero di elementi ed il peso computazionale del modello.

Si precisa che per la costruzione della mesh 3D vengono utilizzati gli elementi "Hex" lineari a 8 nodi.



Figura 4.7: Creazione della mesh 3D in HyperMesh



Figura 4.8: Dettaglio della mesh 3D della molla

4.3.2 Definizione dei carichi e dei vincoli

Le ipotesi adottate per la definizione di carichi e vincoli sono riportate di seguito:

- Il contatto tra l'estremità della molla ed il raccordo del piattello superiore della sospensione è considerato come una connessione rigida. In questa area del componente è applicato il carico;
- Il contatto tra la superficie inferiore della molla ed il piattello di base della sospensione è considerato come una connessione rigida. In questa area del componente sono applicati i vincoli;
- Il carico è applicato lungo la direzione dell'asse Z del sistema di riferimento, parallelo all'asse della molla;
- La componente verticale dello spostamento della molla è un ordine di grandezza superiore rispetto alle altre due;
- Carichi e vincoli sono applicati sulle porzioni di superficie della molla a contatto con gli elementi adiacenti della sospensione.

Per vincolare e caricare opportunamente il componente si sfruttano gli elementi monodimensionali RBE2. I nodi dipendenti sono rigidamente collegati al nodo master (indipendente), che governa il movimento dell'intero insieme. Di conseguenza, un carico o un vincolo impostato sul nodo master è automaticamente trasmesso a tutti i nodi dipendenti.



Figura 4.9: Elementi monodimensionali RBE2 e RBE3

Il punto di applicazione del carico esplorativo di 1185 N coincide con la posizione del master node; il carico viene ripartito sui nodi dipendenti appartenenti alla superficie superiore della molla, vincolati a muoversi solamente lungo l'asse Z verticale.



Figura 4.10: Nodi caricati e vincolati appartenenti alla superficie superiore della molla

I nodi che giacciono sulla superficie inferiore della molla sono completamente vincolati. Anche in questo caso si impiegano dei rigidi RBE2 per collegare il nodo master, appartenente all'asse del componente, con i nodi dipendenti.



Figura 4.11: Nodi vincolati appartenenti alla superficie inferiore della molla

Ultimata la costruzione del modello FEM del componente è possibile eseguire delle analisi statiche lineari e visionare i risultati delle simulazioni in HyperView, ambiente di post-processing del pacchetto Altair.

A questo punto, è fondamentale uno studio di convergenza del modello al fine di ricercare il giusto trade-off tra accuratezza dei risultati e costo computazionale delle analisi. Infatti, le successive simulazioni a fatica sono costruite sulla base del modello FEM statico. Pertanto, è immediato comprendere che un'adeguata ottimizzazione del modello di partenza garantisce un risparmio notevole di tempo di calcolo.

Si sottolinea che il numero di analisi a fatica da eseguire potrebbe essere decisamente elevato ed è funzione della qualità dei risultati derivanti dall'addestramento della rete neurale.

4.4 Analisi di convergenza del modello

Le analisi di convergenza consentono di ottimizzare un modello FEM, ovvero ricavare la tipologia ed il numero di elementi idonei a descrivere il campo di tensioni e di spostamenti che caratterizzano il problema strutturale esaminato.

Un modello giunge a convergenza quando l'incremento del numero di nodi ed elementi della mesh non produce variazioni significative sui risultati delle simulazioni.

Per essere validi, i test di convergenza si effettuano con elementi non distorti e mesh autocontenute, Una mesh si dice autocontenuta se un singolo elemento ad essa appartenente può essere suddiviso in elementi più piccoli della medesima forma.

L'ottimizzazione del modello FEM della molla della sospensione ha richiesto diverse analisi, che hanno chiarito il legame tra la dimensione degli elementi della mesh e lo stato di tensione presente nel componente.

Il procedimento che ha condotto ad individuare il modello ottimale viene descritto suddividendo l'analisi di convergenza nei due step illustrati di seguito.

4.4.1 Fase 1

Nella prima fase di studio della convergenza la mesh è generata impostando nel software le variabili L ed N_{base} . Per chiarire il significato della simbologia adottata si allega una lista riepilogativa:

- *L* [*mm*]: dimensione dell'elemento in direzione circonferenziale;
- *N*_{base}: numero di elementi presenti sulla semicirconferenza;
- *N_{tot}*: numero di elementi complessivo del modello;
- *X_{max}* [*mm*]: spostamento massimo in direzione *X*;
- *Y_{max}* [*mm*]: spostamento massimo in direzione *Y*;
- Z_{master} [mm]: spostamento massimo del "master node" in direzione Z;
- σ [*MPa*]: massima tensione equivalente di von Mises;
- $ERR_{X,max}$ [%]: errore relativo percentuale sullo spostamento massimo X_{max} ;
- $ERR_{Y,max}$ [%]: errore relativo percentuale sullo spostamento massimo Y_{max} ;
- $ERR_{Z,master}$ [%]: errore relativo percentuale sullo spostamento massimo Z_{master} ;
- ERR_{σ} [%]: errore relativo percentuale sulla tensione massima equivalente σ ;
- $t_{sim}[s]$: tempo di simulazione.

Gli errori relativi sono valutati assumendo "esatti" gli output della simulazione numero 1. Di seguito si riportano sinteticamente i risultati ottenuti sotto forma di grafici e tabelle. Si rammenta che il carico esplorativo impostato nel software è pari a 1185 N.

Nsim	P	arametri l	Mesh	Risultati								
/	L [mm]	Nbase	Ntot	X _{max} [mm]	Y _{max} [mm]	Zmaster [mm]	σ [MPa]	ERRx,max [%]	ERRY,max [%]	ERRz,master [%]	ERRø [%]	tsim [S]
1	1	18	333999	1,423	-1,745	-69,016	565,18	/	/	/	/	145
2	1,5	12	122220	1,385	-1,700	-67,329	449,77	2,67%	2,58%	2,44%	20,42%	37
3	2	9	41420	1,349	-1,658	-65,565	432,70	5,20%	4,99%	5,00%	23,44%	12
4	2,5	7	22724	1,325	-1,630	-64,388	421,10	6,89%	<mark>6,</mark> 59%	6,71%	25,49%	7
5	3	6	15288	1,308	-1,610	-63,572	415,12	8,08%	7,74%	7,89%	26,55%	5

Tabella 4.4: Risultati dell'analisi di convergenza dopo lo step 1



Figura 4.12: Andamento dello spostamento verticale in funzione del numero di elementi



Figura 4.13: Andamento della tensione equivalente in funzione del numero di elementi



Figura 4.14: Andamento dell'errore relativo allo spostamento verticale in funzione del numero di elementi



Figura 4.15: Andamento dell'errore relativo alla tensione equivalente in funzione del numero di elementi



Figura 4.16: Andamento del tempo di simulazione in funzione del numero di elementi

Visionando i risultati conseguiti, si nota che all'aumentare del numero di elementi:

- Incrementano gli spostamenti lungo tutte le direzioni;
- Incrementa la tensione massima equivalente di von Mises;
- Incrementa il costo computazionale del modello;
- Diminuiscono gli errori relativi sugli spostamenti e le tensioni.

Un modello agli elementi finiti è più rigido di un oggetto reale. Infatti, la discretizzazione del continuo obbliga i punti interni ad un elemento a seguire il comportamento dei nodi, introducendo nel modello un vincolo che non ha corrispondenza nel mondo fisico. All'aumentare del numero degli elementi, il modello diviene meno rigido con un conseguente incremento degli spostamenti e delle tensioni.

Tuttavia, il metodo selezionato per costruire la mesh non garantisce la convergenza del modello. Osservando la tabella 4.4, si deduce che tra la quinta e la seconda simulazione, a fronte di un aumento di 106932 elementi, la tensione massima agente nella molla subisce una variazione pari a 34,65 MPa. Viceversa, confrontando la prima e la seconda analisi si registra un aumento dello stress pari a 115,41 MPa, decisamente elevato se rapportato ad un incremento di 211779 elementi.

Si potrebbe ipotizzare che il componente presenti una concentrazione locale di tensione che viene catturata dal modello solamente nelle simulazioni con dimensione degli elementi minore di 1 mm.

La suddetta ipotesi trova conferma esaminando le figure 4.17 e 4.18 (visibili a pagina seguente), nelle quali si riportano i risultati della prima analisi.

In prossimità della superfice vincolata del componente si assiste ad un'intensificazione locale della tensione, che per essere colta necessita dell'utilizzo di elementi con ridotta dimensione circonferenziale. D'altra parte, il corpo centrale della molla non richiede una mesh di dettaglio.



Figura 4.17: a) Mappa degli spostamenti ; b) Mappa delle tensioni equivalenti



Figura 4.18: Area del componente sottoposta alla massima tensione equivalente

4.4.2 Fase 2

Alla luce delle osservazioni effettuate, la seconda fase dell'analisi di convergenza prevede una costruzione alternativa della mesh. Il volume della molla è diviso in cinque zone distinte, riassunte nella seguente tabella e visibili in figura 4.19 (la lettera "C" nella denominazione delle zone indica l'abbreviazione di "Cilindro").

Zona	Caratteristica
C1	RBE2 carico
C2	Zona transizione superiore
C3	Corpo centrale componente
C4	Zona transizione inferiore
C5	RBE2 incastro

Tabella 4.5: Suddivisione del componente in 5 zone per la fase 2 dell'analisi di convergenza



Figura 4.19: Zone di suddivisione del componente per la fase 2 dell'analisi di convergenza

Le zone C2 e C4 richiedono una mesh fine, poiché corrispondono al tratto di superficie del componente dove si riscontra un marcato gradiente locale di tensione. Viceversa, le zone C1, C3 e C5 non sono caratterizzate da particolari criticità e vengono discretizzate con elementi di dimensione maggiore.

Nello step 2 dello studio di convergenza, gli errori relativi delle singole simulazioni sono valutati assumendo "esatti" gli output dell'analisi di riferimento numero 1, visibili nella tabella 4.6.

Si noti che impostando una dimensione circonferenziale degli elementi di 0,2 mm, il modello giunge a convergenza. Infatti, un incremento di 229152 elementi, tra le mesh di riferimento 1 e 2, produce una variazione della tensione equivalente di soli 1,12 MPa.

RISULTATI CON MESH DI RIFERIMENTO: 1								
L [mm]	Nbase	Ntot	X _{max} [mm]	Y _{max} [mm]	Zmaster [mm]	σ [MPa]	tsim [S]	
0,2	12	916608	1,330	-1,630	-64,405	680,00	345	

	RISULTATI CON MESH DI RIFERIMENTO: 2							
L [mm]	Nbase	Ntot	X _{max} [mm]	Y _{max} [mm]	Zmaster [mm]	σ [MPa]	tsim [s]	
0,2	14	1145760	1,337	-1,633	-64,416	681,12	504	

ΔN	ERRZ,master [%]	ERR _o [%]
229152	0,017%	0,164%

Tabelle 4.6 - 4.7 - 4.8: Risultati delle simulazioni di riferimento

A questo punto, si ricercano gli opportuni parametri della mesh nelle diverse zone del componente affinché siano rispettati i seguenti vincoli:

- Numero di elementi del modello inferiore a 60000;
- Errore relativo percentuale sullo spostamento massimo Z_{master} inferiore all'1%;
- Errore relativo percentuale sulla tensione massima equivalente σ inferiore allo 0,5%;
- Tempo di calcolo inferiore a 20 secondi.

Le analisi con i risultati migliori sono riportate nelle tabelle allegate di seguito (le differenze tra due analisi consecutive sono evidenziate in giallo).

	NUMERO ANALISI: 1									
Zona	L [mm]	Ntot	ERRx,max [%]	ERRY,max [%]	ERRz,master [%]	ERRø [%]	tsim[s]			
C1	4	61572	5,26%	4,91%	1,63%	1,05%	21			
C2	0,2									
C3	4									
C4	0,2									
C5	4	1								

NUMERO ANALISI: 2									
Zona	L [mm]	Ntot	ERRx,max [%]	ERRY,max [%]	ERRz,master [%]	ERRσ [%]	tsim[s]		
C1	4	67368	0,23%	0,06%	0,12%	0,29%	22		
C2	0,2								
C3	3,5								
C4	0,2								
C5	4								

Tabelle 4.9 - 4.10: Risultati della prima e seconda simulazione

NUMERO ANALISI: 3								
Zona	L [mm]	Ntot	ERRx,max [%]	ERRY,max [%]	ERRZ,master [%]	ERRσ [%]	tsim[s]	
C1	4	59136	0,15%	0,06%	0,10%	10,16%	19	
C2	0,4							
C3	3,5							
C4	0,4							
C5	4							

NUMERO ANALISI: 4								
Zona	L [mm]	Ntot	ERRX,max [%]	ERRY,max [%]	ERRz,master [%]	ERRø [%]	tsim[s]	
C1	4	55356	0,38%	0,00%	0,07%	0,36%	18	
C2	0.4*							
C3	3,5							
C4	0.4*							
C5	4							

Tabelle 4.11 - 4.12: Risultati della terza e quarta simulazione

L'ultimo modello proposto rispetta i vincoli di progetto. Nelle zone C2 e C4 la dimensione nominale degli elementi è di 0,4 mm. In realtà, la lunghezza circonferenziale dell'elemento è funzione della posizione occupata. Infatti, impostando nel software una legge esponenziale per definire la disposizione spaziale degli elementi è possibile infittire localmente la mesh (in prossimità delle aree della molla vincolate e caricate) per descrivere opportunamente lo stato tensionale del componente senza rendere il modello computazionalmente oneroso.

Nella tabella 4.13 si riassumono i parametri della mesh ed i risultati del modello ottimizzato.

Lo studio di convergenza permette di risparmiare una notevole quantità di tempo di calcolo; infatti, la differenza tra il tempo di simulazione del modello ottimizzato e quello di riferimento ammonta a 327 secondi.

MODELLO OTTIMIZZATO								
Parametri mesh			Risultati					
Zona	L [mm]	Ntot	ERRx,max [%]	ERRY,max [%]	ERRZ,master [%]	ERRø [%]	tsim[s]	
C1	4	55356	0,38%	0,00%	0,07%	0,36%	18	
C2	0.4 (E)							
C3	3,5							
C4	0.4 (E)							
C5	4							

Tabella 4.13: Modello selezionato a valle dell'analisi di convergenza

4.5 Modello ottimizzato: definizione del carico

Fino ad ora, tutte le simulazioni sono state effettuate con un carico esplorativo di 1185 N. Una volta individuato il modello ottimizzato che descrive adeguatamente il problema fisico, è necessario determinare il carico statico reale esercitato sull'elemento elastico della sospensione.

Si rammenta che le dimensioni del componente prima e dopo il montaggio in vettura sono ricavate a partire dai disegni tecnici della sospensione anteriore oggetto di studio. Per la valutazione del carico si definiscono le seguenti grandezze:

- L_{def} : lunghezza della molla nella configurazione deformata;
- Δz : differenza di lunghezza della molla tra la configurazione indeformata e quella deformata;
- F'_s : carico statico che produce una compressione della molla pari a Δz .

$$\begin{cases} L_{def} = 245,4 \ mm \\ \Delta z = L_{ind} - L_{def} = 120,05 \ mm \\ F'_{s} = 2212,3 \ N \end{cases}$$

La forza F'_s rappresenta il carico esercitato dalla massa sospesa del veicolo su una delle due sospensioni anteriori. Tuttavia, è opportuno considerare anche il supplemento di forza agente sulla molla dovuto al peso del conducente.

Ipotizzando che la massa del guidatore M_g sia pari a 65 kg e che si ripartisca equamente sulle quattro ruote dell'auto, si ottiene:

$$\begin{cases} M_g = 65 \ kg \\ F_s = F'_s + \frac{M_g}{4} \cdot g = 2372,1 \ N \\ \Delta z_{tot} = 128,72 \ mm \end{cases}$$

 F_s è il carico statico reale gravante sul componente che provoca una compressione dell'elemento elastico della quantità Δz_{tot} . Data la linearità del modello, la curva caratteristica forza-spostamento è una retta e la costante elastica *k* della molla vale:

$$k = \frac{F_s}{\Delta z_{tot}} = 18,43 \ \frac{N}{mm}$$

Per verificare che il carico trovato sia corretto si effettua un confronto con la forza esercitata dalla massa sospesa del veicolo di prova sulla sospensione:

- *M_{tot}*: massa del veicolo;
- M_{sos} : massa sospesa del veicolo, pari al 90% della massa totale;
- F_e : forza effettiva agente sulla sospensione.

$$\begin{cases} M_{tot} = 1025 \ kg \\ M_{sos} = 0.9 \cdot M_{tot} = 922.5 \ kg \\ F_e = \frac{M_{sos}}{4} \cdot g + \frac{M_g}{4} \cdot g \\ F_e = 2419.4 \ N \end{cases}$$

5 Molla della sospensione anteriore: analisi FEM a fatica

5.1 Introduzione al fenomeno

In generale, un organo di macchina può essere sollecitato da carichi statici o dinamici; se le sollecitazioni sono variabili nel tempo si dice che il componente è sottoposto a condizioni affaticanti. Nel nostro caso di studio, il contatto tra la ruota dell'auto ed il manto stradale definisce lo spettro di carico random che grava sulla molla della sospensione.



Figura 5.1: Esempi di carichi random

Da un punto di vista fenomenologico, sollecitazioni affaticanti apportano al componente un danno irreversibile, progressivo e solitamente circoscritto ad un'area ristretta. Inoltre, la rottura dell'oggetto esaminato potrebbe avvenire a causa di tensioni inferiori al limite di snervamento del materiale. Il processo del cedimento a fatica è contraddistinto da 3 fasi:

- 1) Nucleazione: il carico affaticante genera delle modifiche microstrutturali nel materiale; il movimento delle dislocazioni innesca un meccanismo che conduce allo scorrimento di piani cristallini favorevolmente orientati, con formazione di micro-cricche lungo i grani;
- 2) Propagazione: coalescenza di cricche appartenenti a piani cristallografici diversi, origine dei bordi radiali (variazione del piano di propagazione cricca) e delle linee di arresto;
- 3) Collasso finale: la cricca raggiunge una dimensione critica che porta alla rottura del componente senza alcun preavviso.

In figura 5.2 è mostrato l'aspetto tipico della zona di rottura finale.



Figura 5.2: Rottura per fatica

5.2 Criteri di calcolo

L'obiettivo delle analisi a fatica è stimare la vita (o il danneggiamento) di un componente sottoposto a sollecitazioni variabili nel tempo. Esistono 3 differenti approcci per il calcolo a fatica:

- SAFE LIFE: l'elemento strutturale non viene ispezionato durante tutto l'arco della sua vita in quanto si ipotizza che nessun difetto possa propagare fino a raggiungere una dimensione critica. I metodi previsti da questo approccio per studiare il fenomeno della fatica sono lo stress-life e lo strain-life;
- 2) FAIL SAFE: criterio estremamente conservativo; individuato il componente del sistema maggiormente critico, questo viene ispezionato o sostituito prima che possa avvenire il cedimento;
- 3) DAMAGE TOLERANCE: si ipotizza che l'elemento strutturale presenti dei difetti ancor prima della messa in opera. Si programmano delle manutenzioni ad intervalli regolari di tempo e tra due ispezioni successive nessun difetto dovrebbe crescere al punto da portare a rottura il componente. Il metodo previsto da questo approccio per studiare il fenomeno della fatica è quello della meccanica della frattura.

Quando le tensioni e le deformazioni sviluppate nell'elemento strutturale risiedono nel campo elastico, le durate sono elevate (tipicamente superiori ai 1000 cicli) e si parla di fatica ad alto numero di cicli. Viceversa, nella fatica oligociclica (a basso numero di cicli) il componente è soggetto a tensioni e deformazioni principalmente plastiche.

Il metodo strain-life è impiegato per analisi a fatica oligociclica, dove il meccanismo di rottura è governato dalla deformazione plastica locale. D'altra parte, il metodo stress-life risulta adeguato in applicazioni ad alto numero di cicli e si basa sulla valutazione delle tensioni medie ed alternate mediante i diagrammi empirici di Wöhler ed Haigh.

Nel nostro progetto si utilizza il metodo stress-life per il calcolo a fatica per due motivi:

- il limite di snervamento del materiale è estremamente alto, pertanto le sollecitazioni si sviluppano in campo elastico;
- la molla della sospensione presenta durate elevate, in genere di diversi anni.

5.3 Metodo stress-life

Una storia temporale di carico può essere ridotta, mediante i metodi di conteggio, alla somma di singoli blocchi di sollecitazione caratterizzati da sforzo medio, alternato e numero di cicli per cui tale sollecitazione è applicata.

I parametri caratteristici di un qualsiasi carico sinusoidale sono:



Figura 5.3: Tipologie di cicli di tensione

Nel caso di tensione media costante, la durata a fatica può essere calcolata mediante il diagramma di Wöhler, che esprime l'andamento della tensione alternata in funzione del numero di cicli che portano alla rottura il componente. La costruzione di tale curva richiede un elevato numero di test sperimentali condotti in condizioni standard, ad esempio:

- Prova in flessione rotante con tensione media nulla e rapporto di tensione pari a -1;
- Provino con diametro 10 mm;
- Superficie del materiale lucidata a specchio.



Figura 5.4: Curva di Wöhler

Nel campo della fatica ad alto numero di cicli $(10^3 \le N \le 2 \cdot 10^6)$ la curva di Wöhler è descritta da una funzione lineare: la retta di Basquin. Si precisa che in questo caso entrambi gli assi sono espressi in scala logaritmica. Per stimarne l'equazione si ricavano le coordinate dei punti "F" e "G" mediante le proprietà meccaniche del materiale. In particolare:

- $F = (10^3; 0,9 \cdot R_m);$
- $G = (2 \cdot 10^6; \sigma_{D-1,F});$ Equazione retta: $\sigma_a^k \cdot N = B;$
- $\sigma_{D-1,F}$ [*MPa*]: limite di fatica per flessione alternata simmetrica ($\sigma_m = 0$); •
- $R_m[MPa]$: tensione limite di rottura; •
- $R_{p0,2}[MPa]$: tensione limite di snervamento; •
- $\sigma_a[MPa]$: tensione alternata; •
- *N*: numero di cicli;
- B, k: parametri della retta di Basquin. •



Imponendo il passaggio per i punti "F" e "G" si ottengono i parametri *B*, *k* della retta di Basquin.

5.3.1 Effetto della tensione media

La curva di Wöhler opera al medesimo livello di σ_m ; tuttavia, esiste una stretta dipendenza tra il limite di fatica e la tensione media presente nel componente. Per tale ragione è richiesto l'utilizzo combinato della curva di Wöhler con il diagramma di Haigh, che permette di correggere il limite di fatica nel diagramma $\sigma_a - N$ in funzione della σ_m . Sperimentalmente si nota che tensioni medie di compressione non influenzano particolarmente la resistenza a fatica del materiale, mentre tensioni medie di trazione sono dannose poiché tendono a favorire la propagazione delle cricche.

Il diagramma di Haigh può essere tracciato a partire dalle proprietà meccaniche del materiale: tensione di snervamento, di rottura e resistenza a fatica per vita infinita (ottenuta dal Wöhler).

Per interpolare i dati sperimentali nella zona delle $\sigma_m \ge 0$ si sceglie di utilizzare la retta di Goodman (modello maggiormente diffuso ed impiegato).

Di seguito si riporta un diagramma di Haigh con le relative equazioni lineari che contraddistinguono le 4 aree in cui è diviso (in figura 5.6 la lettera "R" sta per "Retta").



Figura 5.6: Diagramma di Haigh

Equazione R1:

 $\sigma_a = \sigma_m + R_{p0,2} - R_{p0,2} \le \sigma_m \le \sigma_{D-1} - R_{p0,2}$

Equazione R2:

 $\sigma_a = \sigma_{D-1} \quad \sigma_{D-1} - R_{p0,2} \le \sigma_m \le 0$

Equazione R3 (retta di Goodman):

$$\frac{\sigma_a}{\sigma_{D-1}} + \frac{\sigma_m}{R_m} = 1 \qquad 0 \le \sigma_m \le \frac{R_{p0,2} - \sigma_{D-1}}{1 - \frac{\sigma_{D-1}}{R_m}}$$

Equazione R4:

$$\sigma_{a} = -\sigma_{m} + R_{p0,2} \qquad \frac{R_{p0,2} - \sigma_{D-1}}{1 - \frac{\sigma_{D-1}}{R_{m}}} \le \sigma_{m} \le R_{p0,2}$$
5.3.2 Fattori che influenzano la vita a fatica

Le caratteristiche di resistenza a fatica dei materiali sono ottenute con dispendiose campagne sperimentali poiché il fenomeno fisico della fatica è intrinsecamente disperso nei risultati: le curve $\sigma_a - N$ sono associate ad un livello di probabilità. Infatti, applicando il medesimo carico alternato a diversi provini, essi non giungono a rottura con lo stesso numero di cicli.

I diagrammi di Wöhler sono costruiti conducendo differenti tipologie di prove, tra cui quelle in flessione rotante, in trazione/compressione oppure in torsione (tutte con $\sigma_m = 0$ e R = -1, si eseguono cicli di tensione alternata simmetrica). Ovviamente, il comportamento a fatica dei provini utilizzati durante i test non rispecchiano le effettive condizioni di lavoro che un generico componente meccanico affronta durante il suo utilizzo. Per tale ragione, è necessario modificare i limiti di fatica del materiale ricavati sperimentalmente con diversi fattori correttivi.



Figura 5.7: Fattori che influenzano la vita a fatica

Un carico affaticante di trazione-compressione danneggia il componente in modo più severo rispetto ai carichi di flessione. Infatti, una distribuzione uniforme di tensione sviluppa nelle zone di processo delle cricche di fatica (appena sotto la superficie del pezzo) delle tensioni superiori rispetto alle distribuzioni con gradiente.

I risultati sperimentali dimostrano che $\sigma_{D-1}^{T/C} = 0.6 \div 0.85 \sigma_{D-1}^{F}$, dove $\sigma_{D-1}^{T/C}$ indica il limite di fatica a trazione-compressione, mentre σ_{D-1}^{F} rappresenta il limite di fatica a flessione (R = -1). In figura 5.8 è illustrato un diagramma di Wöhler che evidenzia l'influenza del tipo di carico sulla

In figura 5.8 è illustrato un diagramma di Wöhler che evidenzia l'influenza del tipo di carico sulla resistenza a fatica.



Figura 5.8: Influenza della tipologia di carico sul limite di fatica

Si riportano sinteticamente gli effetti degli altri fattori che influenzano il limite di fatica:

- Fattore di scala: a parità di condizioni, un componente di dimensioni maggiori presenta una minor resistenza a fatica, soprattutto se sollecitato a flessione;
- Finitura superficiale: all'aumentare della rugosità superficiale del pezzo si abbassa il limite di fatica, in quanto la nucleazione delle cricche è spesso dovuta a difetti superficiali. Inoltre, l'effetto è ulteriormente amplificato per materiali alto-resistenziali;
- Trattamenti superficiali: i trattamenti termici (tempra, nitrurazione, cementazione) e meccanici (rullatura a freddo, pallinatura) consentono di indurre delle tensioni superficiali di compressione sul componente, con effetti benevoli sul comportamento a fatica;
- Intagli: come accade in statica, brusche variazioni geometriche nel pezzo determinano incrementi locali di tensione, con effetto negativo sulla durata del componente;
- Ambiente aggressivo: la resistenza a fatica di un componente che opera in un ambiente corrosivo diminuisce drasticamente;
- Temperatura: nelle applicazioni ad alta temperatura il limite di fatica decresce sensibilmente. Inoltre, il componente si deforma principalmente in campo plastico ed un approccio di calcolo a fatica di tipo stress-life non è più valido.

5.4 Obiettivi delle analisi e metodologia di calcolo

Le simulazioni a fatica della molla della sospensione anteriore si effettuano con l'ausilio del software FEMFAT (Finite Element Method Fatigue). Gli obiettivi delle analisi a fatica sono:

- Ricerca di opportuni parametri di input da fornire al software per effettuare l'analisi;
- Stima del danneggiamento del componente in funzione delle condizioni di sollecitazione;
- Creazione di un database per correlare le grandezze di input dell'analisi a fatica con quella di output, ovvero il danno subito dalla molla; i dati raccolti verranno utilizzati per il successivo training della rete neurale.

Il modello di accumulo del danno impiegato nel progetto è quello lineare di Palmgren-Miner. Per comprenderne la logica si consideri un componente caratterizzato da un andamento degli sforzi sinusoidale, con tensione media costante e tensione alternata variabile.



Figura 5.9: Modello di danneggiamento lineare di Palmgren-Miner

Con riferimento al diagramma di Wöhler in figura 5.9, si introducono le seguenti grandezze:

- n_1 : numero di cicli applicati al componente con tensione alternata pari a σ_{N1} ;
- n_2 : numero di cicli applicati al componente con tensione alternata pari a σ_{N2} ;
- N_1 : numero di cicli al livello di sollecitazione alternata σ_{N1} che provocherebbe la rottura del componente;
- N_2 : numero di cicli al livello di sollecitazione alternata σ_{N2} che provocherebbe la rottura del componente;
- *D*: danno accumulato dal componente.

In generale, il danneggiamento complessivo è un numero adimensionale ed è pari alla somma dei singoli danni parziali:

$$D_{totale} = \sum_{i=1}^{k} \frac{n_i}{N_i}$$

La rottura si verifica quando:

$$D_{totale} = \sum_{i=1}^{k} \frac{n_i}{N_i} = 1$$

Si sottolinea che nell'esempio riportato è corretto utilizzare il medesimo diagramma di Wöhler per ricavare il danno totale poichè la tensione media è costante per ipotesi. Tuttavia, la grande maggioranza delle applicazioni necessita dell'impiego combinato dei diagrammi di Wöhler ed Haigh, per tener conto dell'effetto della variazione della tensione media.

Per illustrare il procedimento di calcolo in quest'ultimo caso, si consideri un organo meccanico sottoposto ad uno sforzo sinusoidale, come mostrato in figura 5.10.



Figura 5.10: Cicli sinusoidali con tensione media e alternata variabili

I passi da seguire per stimare il danno accumulato dal componente sono:

- Valutazione della tensione media ed alternata i-esime σ_{mi} , σ_{ai} ;
- Conteggio del numero di cicli n_i per cui è applicato il ciclo i-esimo;
- Valutazione del limite di fatica σ_{Di} per il ciclo con tensione media σ_{mi} mediante il diagramma di Haigh;



Figura 5.11: Valutazione della tensione limite di fatica i-esima

• Costruzione del diagramma di Wöhler i-esimo;



 $k_i = \frac{\log(N_G) - \log(N_F)}{\log(\sigma_{Fi}) - \log(\sigma_{Gi})}$ $B_i = \sigma_{Di}^{k_i} \cdot N_G$ $F_i = (10^3; 0.9 \cdot (R_m - \sigma_{mi}))$ $G_i = (2 \cdot 10^6; \sigma_{Di})$

Retta di Basquin: $\sigma_a^{k_i} \cdot N = B_i$

Figura 5.12: Costruzione del diagramma di Wöhler i-esimo

• Calcolo dell'i-esimo danneggiamento $D_i = \frac{n_i}{N_i}$.



Figura 5.13: Valutazione dell'i-esimo danno subito dal componente

Analogamente al caso precedente, i danneggiamenti causati dai singoli cicli di sollecitazione vengono sommati per ottenere il danno complessivo subito dall'organo meccanico. Si precisa che le unità di misura di B, k sono:

$$[B] = MPa^{\frac{1}{\log(MPa)}}$$
$$[k] = \frac{1}{\log(MPa)}$$

Nota la storia temporale di carico con cui è sollecitato il componente, i parametri che permettono di stimarne il danneggiamento sono riportati nella tabella seguente.

PARAMETRI DI INPUT			GRANDEZZE PER CALCOLI INTERMEDI				OUTPUT
$\sigma_{ extsf{m}}$ [Mpa]	σ_{a} [Mpa]	n	σ₀ [Mpa]	k	В	N	D
σ m1	σ a1	n 1	σ D1	k 1	B 1	N 1	D1
σ m2	σ a2	n ₂	σ D2	k 2	B 2	N2	D2
∂ m3	σ a3	n ₃	σ D3	kз	Вз	Νз	Dз
σ mi	σai	ni	σ Di	k i	Bi	Ni	Di
σ mk	O ak	nk	σ Dk	k k	Bk	Nk	Dĸ

Tabella 5.1: Parametri per la stima del danneggiamento del componente nel caso di cicli di sollecitazione sinusoidali con tensione media ed alternata variabili

5.4.1 Calcolo a fatica con storie di carico irregolari

Negli esempi del paragrafo precedente abbiamo considerato storie di sollecitazione con cicli sinusoidali caratterizzati da tensioni medie ed alternate entrambi variabili.

Tuttavia, nel nostro caso di studio il contatto tra la ruota della vettura e la strada definisce uno spettro di carico random che grava sulla molla della sospensione. In questo caso, i passaggi appena descritti per la valutazione del danneggiamento complessivo non sono direttamente applicabili. Fortunatamente, esistono numerosi metodi di conteggio dei cicli che consentono di ricondurre una sollecitazione casuale ad una somma di cicli sinusoidali con σ_m e σ_a variabili. Si sottolinea che i cicli possono essere conteggiati utilizzando la storia temporale di un qualsiasi parametro caratteristico del problema analizzato, come ad esempio spostamento, tensione, forza, accelerazione o deformazione. In generale, prima di applicare un metodo di conteggio è necessario processare i dati della storia di carico in modo da ridurla ad una sequenza di soli picchi e valli, trascurando le variazioni del segnale che non si ritengono significative ai fini dell'analisi a fatica.

In questo paragrafo viene presentato il metodo di conteggio rainflow a tre punti. La condizione per l'estrazione di un ciclo è la seguente: in una combinazione valle-picco-valle (oppure picco-valle picco), il secondo semiciclo possiede un'ampiezza maggiore o uguale a quella del primo semiciclo. In figura 5.14 la condizione è rispettata solamente nei casi 1 e 2.



Figura 5.14: Condizione per l'estrazione di un ciclo nel metodo rainflow a tre punti

Partendo dal primo picco (o valle) della storia temporale, si applica sequenzialmente la suddetta condizione di conteggio, confrontando l'ampiezza del semiciclo i-esimo con quella del semiciclo successivo (i+1). Ogni volta che la condizione è verificata si registrano il valor medio ed alternato del ciclo. Successivamente, i punti che definiscono le estremità dei cicli identificati vengono eliminati dalla storia di carico. Queste iterazioni sono ripetute finché non si identifica nessun nuovo ciclo o il numero di estremi della sequenza risulta inferiore a 3.

Con riferimento alla figura 5.15, si mostra un esempio applicativo del metodo rainflow:

- Inizio dell'identificazione dei cicli a partire dal punto A;
- Il primo ciclo conteggiato è BCB' in quanto BC < CD;
- Il secondo ciclo conteggiato è FGF' in quanto FG < GH;
- I punti *B*, *C* sono eliminati dalla sequenza ed il punto *A* è direttamente collegato con *D*;
- I punti F, G sono eliminati dalla sequenza ed il punto E è direttamente collegato con H;
- Il terzo ciclo conteggiato è EHE' in quanto EH < HI;
- I punti *E*, *H* sono eliminati dalla sequenza ed il punto *D* è direttamente collegato con *I*;
- Il quarto ciclo conteggiato è *ADA*' in quanto *AD* < *DI*;
- Fine delle operazioni di conteggio.

In tabella 5.2 sono riassunti i risultati del conteggio.



Figura 5.15: Esempio di applicazione del metodo di conteggio rainflow a tre punti

Ciclo	da	а	Media	Ampiezza
BC	1	2	1,5	0,5
FG	-1	1	0	1
EH	3	-2	0,5	2,5
AD	5	-3	1	4

Tabella 5.2: Risultato del conteggio con il metodo rainflow per la storia di carico in figura 5.15

Una volta scomposto lo spettro di carico random in una successione di cicli sinusoidali con $\sigma_m e \sigma_a$ variabili, la valutazione del danneggiamento complessivo subito dal componente può essere effettuata con il medesimo procedimento illustrato nel paragrafo precedente. Tale metodologia è applicata per stimare il danno accumulato nella molla della sospensione anteriore oggetto di studio.



Figura 5.16: Spostamento verticale della molla della sospensione in funzione del tempo

In figura 5.16 è visibile un grafico ricavato dai dati sperimentali acquisiti mediante il sensore laser ILD1220-200. Nel nostro caso, la storia di carico irregolare rappresenta lo spostamento del piattello di supporto inferiore della molla in funzione del tempo. Infatti, il comportamento strutturale dell'elemento elastico della sospensione è completamente descritto dalla suddetta grandezza. Se sono rispettate le ipotesi con cui sono stati condotti i test in vettura, lo spostamento del supporto inferiore della molla rispetto alla configurazione statica permette di conoscere lo stato di tensione all'interno del componente.

A partire dai rilevamenti sperimentali, i passi da seguire per stimare il danneggiamento dell'organo meccanico sono:

- 1. Processing dei dati per ottenere una storia di carico composta da una sequenza di soli picchi e valli;
- 2. Applicazione del metodo di conteggio rainflow a tre punti per ricondurre la storia di carico casuale ad una somma di cicli sinusoidali con spostamento medio ed alternato variabili;
- 3. Ricavare tramite il software FEMFAT le tensioni medie ed alternate equivalenti i-esime corrispondenti agli spostamenti medi ed alternati i-esimi (σ_{mi} , σ_{ai});
- 4. Conteggio del numero di cicli n_i per cui è applicato il ciclo i-esimo;
- 5. Valutazione del limite di fatica σ_{Di} per il ciclo con tensione media σ_{mi} mediante il diagramma di Haigh;
- 6. Costruzione del diagramma di Wöhler i-esimo;
- 7. Calcolo del danneggiamento i-esimo D_i ;
- 8. Stima del danneggiamento complessivo del componente mediante la legge di Miner.

Si evidenzia che il procedimento descritto è identico a quello presentato nel paragrafo 5.4, ad eccezione dei punti 1,2 e 3.

Nella tabella seguente sono riassunte le grandezze caratteristiche del problema, che consentono di correlare gli spostamenti della molla con i relativi danni provocati.

PARAMETRI DI INPUT GRANDEZZE F				GRANDEZZE PER CALCOLI INTERMEDI				OUTPUT	
Z _m [mm]	Z₁ [mm]	n	$\sigma_{ extsf{m}}$ [Mpa]	σ₁ [Mpa]	σ₀ [Mpa]	k	В	N	D
Zm1	Za1	n 1	♂ m1	σ a1	σ D1	k 1	B 1	N 1	D 1
Zm2	Za2	n2	♂ m2	♂ a2	σ D2	k 2	B 2	N2	D2
Zm3	Za3	n 3	∂ m3	∂ a3	σ D3	k 3	Вз	Νз	Dз
Zmi	Zai	ni	σ mi	σ ai	σDi	ki	Bi	Ni	Di
Zmk	Zak	nĸ	σ mk	σ ak	σDk	k k	Bk	Nk	Dk

Tabella 5.3: Parametri per la stima del danneggiamento della molla della sospensione anteriore

5.5 Modello agli elementi finiti in FEMFAT

5.5.1 Definizione dei parametri di input-output

Le simulazioni a fatica sono effettuate con il tool "ChannelMAX" del software FEMFAT.

File View Analysis Options	Templates Help		
🔿 🔒 🐸 🖪 📓 🕻	👔 📝 📲 Current Wor	king Directory: C:/Users/lalunni	
ChannelMAX	FE Entities		
FE Entities	Input Files		
Groups	File Format	File Name:	17028
	ALTAIR H3D V	sh_0,4mm_0,7exp_128,72mm.h3d	
Channels		STH Result File Name:	
🚰 Material Data			
Node Characteristics			
San Influence Factors	Nodes:	64049	WELD Definition
Strain Cage Data	Elements:	55356	
Guain Gage Data	Physical Property Tables	: 3	
Analysis Parameters	Groups:	4	
Scratch Setting	SPOT Welding Nuggets	(Stress): 0	
Cutput	SPOT Welding Nuggets	Extended (Stress): 0	
Report	SPOT Welding Points (F	orce): 0	
	SPOT Rivet Nuggets (Str	ress): 0	
Analyze	SPOT Superelements (S	Stress): 0	
Visualization V	WELD Nodes:	0	
BASIC	WELD Elements:	0	
ChannelMAX	SOLID WELD Nodes:	0	
Channelwikk			
TransMAX	Modify Dimension		
HEAT Sehitoglu	Length Unit Multiplication Fact	tor: 1.00000 Multiply	
SPECTRAL	365,45 mm		Shell Thickness Min: 0.000 [mm] Max: 0.000 [mm]
SPOT Remeshing	-		
STRAIN Calc	154 84 mm	1540	94 mm
Results Manager	10 1,0 1 1111	101,	

Figura 5.17: Interfaccia grafica del software FEMFAT

I parametri di input di un'analisi sono lo spostamento medio, alternato ed il numero di cicli di applicazione della sollecitazione. La grandezza di output è il danneggiamento subito dalla molla della sospensione.



Figura 5.18: Identificazione dei parametri di input delle analisi a fatica

Le sollecitazioni che agiscono sulla molla sono di due tipologie:

- Statiche: il peso dell'auto, dei passeggeri e di eventuali bagagli induce una compressione dell'organo meccanico, anche a veicolo fermo;
- Dinamiche: il contatto tra lo pneumatico ed il manto stradale determina lo spettro di carico affaticante che grava sull'elemento elastico.

La figura 5.18 chiarifica il significato dello spostamento medio ed alternato del componente; i valori numerici riportati derivano dai calcoli mostrati nel capitolo 4 e riassunti nella tabella 5.4.

Lunghezza componente [mm]					
Tipologia configurazione	L [mm]	Lo [mm]	$\Delta L = L_0 - L [mm]$		
1: Pre-montaggio	365,45	365,45	0		
2: Post-montaggio	245,4	365,45	120,05		
3: Post-montaggio e conducente a bordo	236,73	365,45	128,72		

Tabella 5.4: Lunghezza della molla della sospensione in funzione delle condizioni di lavoro

Prima che il veicolo intraprenda la marcia la molla è sollecitata staticamente e presenta al suo interno uno specifico campo di tensione. In prima battuta, lo spostamento medio Z_m può essere visto come lo schiacciamento statico del componente nella configurazione post-montaggio e con conducente a bordo della vettura. D'altra parte, lo spostamento alternato Z_a è correlato al movimento del piattello inferiore della molla, rigidamente collegato allo pneumatico. Applicando il metodo di conteggio rainflow a tre punti ad una storia di carico random si ottengono gli spostamenti medi, alternati ed il numero di ripetizioni che caratterizzano ogni ciclo sinusoidale estratto. È evidente che per valutare fedelmente il danneggiamento della molla è necessario effettuare simulazioni a fatica con Z_m , Z_a ed n variabili. Gli intervalli di variazione delle grandezze di input sono stati definiti dopo aver visionato gli andamenti delle acquisizioni sperimentali e sono indicati nella tabella 5.5.

	Z _m [mm]	Z₁ [mm]	n
Zm0 [mm]	Intervallo di variazione	Intervallo di variazione	
128,72	[-50+Zm0, 50+Zm0]	[1, 50]	[1000, 100000]

Tabella 5.5: Intervalli di variazione delle grandezze di input per le simulazioni a fatica

Ad esempio, se venisse estratto un ciclo sinusoidale con $Z_m = 15 mm$, $Z_a = 10 mm$, n = 1000, il suo andamento nel tempo sarebbe il seguente:



Figura 5.19: Esempio di ciclo sinusoidale

5.5.2 Set-up delle simulazioni

Per effettuare un'analisi a fatica in FEMFAT si procede nel modo seguente:

- Definizione delle proprietà del materiale;
- Costruzione dei diagrammi di Wöhler e Haigh;
- Assegnazione delle proprietà del materiale agli elementi della mesh;
- Importazione dei file dei risultati delle analisi statiche nella sezione "Channels";
- Definizione dei fattori che influenzano le analisi a fatica nella sezione "Influence factors";
- Scelta dell'obiettivo dell'analisi alla sezione "Analysis parameters";
- Esecuzione della simulazione;
- Visualizzazione dei risultati.

Si rammenta che la molla della sospensione è realizzata con l'acciaio 55 SiCr 6-3, le cui caratteristiche principali sono riportate nelle tabelle 5.6 e 5.7.

	Tensione di snervamento [MPa]	1750
Dranziatà magazziaka	Tensione di rottura [MPa]	2100
Proprieta meccaniche	Allungamento a rottura [%]	≥5
	Tensione limite di fatica [Mpa]	945,1
	Densità [kg/m^3]	7800
Bronzietà fisishe	Modulo di Young [MPa]	206000
Proprieta fisicile	Modulo di elasticità tangenziale [MPa]	80000
	Coefficiente di Poisson	0,288
	C [%]	0,5÷0,6
	Si [%]	1,2÷1,6
	S [%]	≤0,025
Composizione chimica	P [%]	≤0,03
	Mn [%]	0,5÷0,9
	Cu [%]	≤0,12
	Cr [%]	0,5÷0,8

Tabella 5.6: Proprietà del materiale

LIMITI DI FATICA (R=-1)				
Flessione alternata simmetrica [MPa]	1086			
Trazione alternata simmetrica [MPa]	945			
Torsione alternata simmetrica [MPa]	546			

Tabella 5.7: Tensioni limite di fatica del materiale per diverse tipologie di sforzi elementari

A partire dai parametri del materiale, si forniscono al software i diagrammi di Wöhler e Haigh. Si sottolinea che FEMFAT richiede che la costruzione dei diagrammi sia realizzata considerando il limite di fatica relativo alle prove di trazione/compressione alternata simmetrica.

Nelle figure a pagina successiva si riportano le curve di Wöhler e Haigh dell'acciaio 55 SiCr 6-3.







Figura 5.21: Diagramma di Haigh del materiale

I risultati delle analisi statiche eseguite in HyperMesh sono salvati in file con estensione "H3D". Quest'ultimi divengono gli input delle simulazioni a fatica e sono caricati in FEMFAT attraverso l'apposita sezione denominata "Channels". In tal modo è possibile costruire la sinusoide che definisce l'andamento dello spostamento della molla in funzione del tempo.

Per ogni analisi a fatica si importano due file, con la conseguente creazione di due canali:

- Nel primo file H3D è salvata la soluzione dell'analisi statica che costituisce la parte media del ciclo;
- Nel secondo file H3D è salvata la soluzione dell'analisi statica che costituisce la parte alternata del ciclo.



Figura 5.22: Definizione dei Channels in FEMFAT

Il software sovrappone i risultati delle analisi statiche e genera lo spostamento sinusoidale risultante con il quale viene sollecitato il componente. Il numero di cicli di ripetizione del carico viene impostato nella sezione "Analysis parameters". Si sottolinea che per la successiva valutazione del danneggiamento subito dalla molla, il software trasforma i valori di spostamento medio ed alternato nei corrispondenti valori di tensione media ed alternata.

Alla voce "Influence factors" si selezionano i parametri che permettono di correggere la pendenza della retta di Basquin e la tensione limite di fatica del diagramma di Wöhler. Nelle nostre analisi si è scelto di considerare l'influenza dei seguenti fattori:

- Gradienti di tensione, ovvero l'intensificazione della sollecitazione che si verifica nelle zone del componente con brusche variazioni geometriche;
- Tensione media, il cui effetto sui limiti di fatica del componente è valutabile mediante il diagramma di Haigh;
- Livello di probabilità associato alla curva di Wöhler.

In figura 5.23 è visibile la sezione dedicata agli "Influence factors" all'interno del software.

File View Analysis Options	Templates Help			
🔿 🔒 🖆 🖪 📓 🖡	🗿 📝 📲 Current Working Directory: C:/Users/Ialunni			
ChannelMAX	Influence Factors			
FE Entities				
Groups	Stress Gradient			
	Gradient Computation Method	FEMFAT 2.4	\sim	
Material Data	Indurance Limit I Slope / Cycle Limit	FEMFAT 2.4	~	
Node Characteristics	Mean Stress			
Influence Factors	✓ Endurance Limit	FEMFAT 4.1	\sim	
	Slope / Cycle Limit	FEMFAT 5.1	\sim	
Strain Gage Data	Surface Roughness	FKM / IABG (Rz)	\sim	
Analysis Parameters	Mean (and Amplitude) Stress Rearrangement PLAST	Mean: Without Sequence Influence	$\overline{}$	
Scratch Setting	Modified Haigh Diagram (Ultimate Tensile Strength)	Stress Gradient Influence	$\overline{}$	
📑 Output	Technological Size Influence	FKM-Guideline	\sim	
Report	Statistical Influence	Gauss (LogN)	~	
Analyze	Isothermal Temperature Influence	FEMFAT 4.6	\sim	
	Cast Microstructure			
Visualization v	Effective Plastic Strain	Method of Variable Slopes	\sim	
BASIC	Tempering Influence (for Tempering Steel only)			
No	Surface Residual Stresses			
	Boundary Layer			
FransMAX	Fiber Orientation	Logarithmic interpolation	\sim	
	Local Material Properties			
1EA I Senitogiu	Rotating Principal Stresses Influence	FEMFAT 4.2	\sim	
SPECTRAL	Combination Method Influence Factors	FEMFAT 2.0	~	
SPOT Remeshing				
STRAIN Calc				
Results Manager				

Figura 5.23: Definizione degli Influence Factors in FEMFAT

L'ultimo passaggio da eseguire prima di lanciare una simulazione a fatica è la scelta della tipologia di analisi da implementare: nel nostro caso si seleziona la voce "Damage – MINER Modified", in quanto la grandezza fisica richiesta in output è il danneggiamento della molla. Per tener conto dell'effetto affaticante apportato dai cicli sinusoidali con tensione alternata equivalente inferiore al limite di fatica del componente, il diagramma di Wöhler presenta pendenza pari a 2k - 1 a partire da $N = 2 \cdot 10^6$ cicli. Come già anticipato, in questa sezione del software è possibile impostare il numero di cicli con cui sollecitare il componente.

File View Analysis Options	Templates Help
🔿 🔒 📇 🖪 🚺	🔮 🔊 📲 Current Working Directory: C:/Users/Ialunni
ChannelMAX	Analysis Parameters
	Analysis Parameters Analysis Target Damage Safety Factor due to Cycle Limit Sig_m = const Cycles: 6.5e+04 Static Safety Factor BREAK FEMFAT 5.0 Criterion: Uitimate Strengtn Stress/Strain Comparison STRAIN Comp Degree of Multiaxiality Global Parameters Analysis Filter Cutting Plane Parameters Stress Selection Automatic 97.500000 [%] Rainflow Counting Number of Rainflow Classes: 64 Rainflow Counting Method: FEMFAT 5.1 Amplitude Limit for Class Filter: 0.0 [N/mm2] High Resolution
ChannelMAX TransMAX	
HEAT Sehitoglu	Result Group
SPECTRAL	Group Name: Most Critical Nodes Group Number of Nodes: 10 Number of Nodes: 10
SPOT Remeshing	Label: 0 Create Group
STRAIN Calc	
Results Manager	

Figura 5.24: Scelta del target dell'analisi in FEMFAT

5.5.3 Punto di lavoro nel diagramma di Haigh

Valutare la posizione del punto di lavoro "P" nel diagramma di Haigh è di fondamentale importanza, in quanto la correzione della curva di Wöhler e la stima del danneggiamento del componente dipendono dai valori della tensione media ed alternata equivalenti. Le coordinate del punto P sono:





Figura 5.25: Definizione del punto di lavoro nel diagramma di Haigh

In FEMFAT, i calcoli delle tensioni equivalenti e del danno si basano sul criterio del piano critico. In generale, il piano critico indica la zona del componente in cui avviene la nucleazione delle cricche di fatica. Nel software corrisponde al piano dove si ottiene il minimo coefficiente di sicurezza o il massimo danneggiamento, a seconda della tipologia di analisi condotta.

Il criterio del piano critico consente di ricondurre uno stato di sollecitazione multiassiale ad uno stato di sollecitazione uniassiale equivalente: in tal modo, è nuovamente possibile stimare il danno subito dalla molla a partire dal diagramma di Wöhler.

Il procedimento seguito dal software per effettuare una simulazione a fatica è il seguente:

- Sovrapposizione nel tempo degli stati di tensione salvati nei file di output delle analisi statiche (ovvero i file H3D presenti nella sezione "Channels");
- Calcolo dei tensori delle tensioni per ogni istante temporale nei diversi piani di riferimento, ovvero quei piani che potrebbero rivelarsi critici per il cedimento del componente;
- Valutazione delle tensioni medie ed alternate equivalenti per tutti i piani di riferimento;
- Correzione della curva di Wöhler mediante il diagramma di Haigh e gli influence factors selezionati;
- Stima del danno per ogni piano di riferimento;
- Individuazione del piano critico e visualizzazione dei risultati.



Figura 5.26: Stato di tensione nel piano critico

In figura 5.26 è rappresentato un generico vettore tensione che descrive lo stato tensione del componente nel piano critico. Nel caso di analisi a fatica, le componenti di tensione normale e tangenziale presentano a loro volta una parte media ed alternata.

In FEMFAT, le coordinate del punto di lavoro P sono definite nel seguente modo:

$$\begin{cases} \sigma_{a,eq} = \sqrt{\sigma_{a,n}^2 + \left(\frac{\sigma_{D-1}}{\tau_{D-1}}\right)^2 \cdot \tau_a^2} \\ \sigma_{m,eq} = \sqrt{\sigma_{m,n}^2 + \left(\frac{\sigma_s}{\tau_s}\right)^2 \cdot \tau_m^2} \end{cases}$$

- $\sigma_{a,eq}$: tensione alternata equivalente;
- $\sigma_{a,n}$: tensione alternata normale al piano critico;
- τ_a : tensione alternata tangenziale al piano critico;
- σ_{D-1} : tensione limite di fatica a trazione/compressione;
- τ_{D-1} : tensione limite di fatica a torsione alternata;
- $\sigma_{m,eq}$: tensione media equivalente;
- $\sigma_{m,n}$: tensione media normale al piano critico;
- τ_m : tensione media tangenziale al piano critico;
- σ_s : tensione di snervamento a trazione;
- τ_s : tensione di snervamento a torsione.

Nell'espressione della tensione alternata equivalente è presente il rapporto σ_{D-1}/τ_{D-1} per tener conto dell'effetto delle tensioni tangenziali sulle curve di Wöhler definite con limite di fatica a trazione/compressione alternata simmetrica.

5.5.4 Risultati delle simulazioni

In questo paragrafo si riportano sinteticamente i risultati ottenuti a valle delle analisi a fatica. Il numero totale di simulazioni effettuate ammonta a 32240, frutto della combinazione di:

- 31 simulazioni con differenti spostamenti medi;
- 26 simulazioni con differenti spostamenti alternati;
- 40 simulazioni con differenti numeri di cicli.

Un numero di analisi così elevato è dettato dall'esigenza di creare un database abbastanza ampio da garantire un buon addestramento della rete neurale. La zona della molla che presenta i valori massimi di danneggiamento è la medesima in ogni simulazione e corrisponde all'area in cui si ottengono le tensioni massime anche in campo statico. Pertanto, la regione critica per la nucleazione delle cricche di fatica si trova in prossimità della superficie di base della molla.

A titolo di esempio, in figura 5.27 sono riportati i risultati di una simulazione che ha i seguenti parametri di input:



Figura 5.27: Mappa di danneggiamento del componente nella zona maggiormente sollecitata

In tabella 5.8 sono indicati schematicamente i risultati di alcune simulazioni che hanno i seguenti parametri di input:

- Spostamento medio costante e pari a Z_m = 128,72 mm;
 Spostamento alternato variabile: Z_a = 1 mm, Z_a = 11 mm, Z_a = 25 mm, Z_a = 41 mm;
 Numero di cicli variabile tra 1000 e 100000.

La grandezza di output è il danno "D" subito dal componente.

Zm [mm]	118,2	Zm [I	mm]	118,2
ΔZm	[mm]	-10,52	ΔZm	[mm]	-10,52
Z₁ [mm]	N	D	Z₀ [mm]	Ν	D
1	1000	5,230E-10	11	1000	1,677E-06
1	5000	2,615E-09	11	5000	8,385E-06
1	10000	5,230E-09	11	10000	1,677E-05
1	15000	7,845E-09	11	15000	2,516E-05
1	20000	1,046E-08	11	20000	3,354E-05
1	25000	1,308E-08	11	25000	4,193E-05
1	30000	1,569E-08	11	30000	5,031E-05
1	35000	1,831E-08	11	35000	5,870E-05
1	40000	2,092E-08	11	40000	6,708E-05
1	45000	2,354E-08	11	45000	7,547E-05
1	50000	2,615E-08	11	50000	8,385E-05
1	55000	2,877E-08	11	55000	9,224E-05
1	60000	3,138E-08	11	60000	1,006E-04
1	65000	3,400E-08	11	65000	1,090E-04
1	70000	3,661E-08	11	70000	1,174E-04
1	75000	3,923E-08	11	75000	1,258E-04
1	80000	4,184E-08	11	80000	1,342E-04
1	85000	4,446E-08	11	85000	1,425E-04
1	90000	4,707E-08	11	90000	1,509E-04
1	95000	4,969E-08	11	95000	1,593E-04
1	100000	5,230E-08	11	100000	1,677E-04

Z _m [mm]		118,2	Z _m [mm]		118,2
ΔZm	[mm]	-10,52	ΔZm	[mm]	-10,52
Za [mm]	Ν	D	Za [mm]	Ν	D
25	10000	6,048E-04	41	10000	4,712E-03
25	15000	9,072E-04	41	15000	7,068E-03
25	20000	1,210E-03	41	20000	9,424E-03
25	25000	1,512E-03	41	25000	1,178E-02
25	30000	1,814E-03	41	30000	1,414E-02
25	35000	2,117E-03	41	35000	1,649E-02
25	40000	2,419E-03	41	40000	1,885E-02
25	45000	2,722E-03	41	45000	2,120E-02
25	50000	3,024E-03	41	50000	2,356E-02
25	55000	3,326E-03	41	55000	2,592E-02
25	60000	3,629E-03	41	60000	2,827E-02
25	65000	3,931E-03	41	65000	3,063E-02
25	70000	4,234E-03	41	70000	3,298E-02
25	75000	4,536E-03	41	75000	3,534E-02
25	80000	4,838E-03	41	80000	3,770E-02
25	85000	5,141E-03	41	85000	4,005E-02
25	90000	5,443E-03	41	90000	4,241E-02
25	95000	5,746E-03	41	95000	4,476E-02
25	100000	6,048E-03	41	100000	4,712E-02

Tabella 5.8: Danneggiamento del componente con spostamento medio costante pari a 118,2 mm

I risultati di tutte le simulazioni sono salvati in un file Excel sotto forma di tabelle del tutto analoghe a quella presentata nell'esempio di pagina precedente.

Per comprendere come i parametri di input influenzino gli output delle analisi a fatica, si riportano sinteticamente alcuni andamenti del danneggiamento subito dalla molla in funzione dello spostamento medio, alternato e del numero di cicli (gli assi dei grafici sono in scala logaritmica).



Figura 5.28: Andamento del danno in funzione del numero di cicli di sollecitazione



Figura 5.29: Andamento del danno in funzione della componente alternata di spostamento

Dai grafici riportati nelle figure 5.28 e 5.29 si possono trarre le seguenti conclusioni:

- A parità di spostamento medio ed alternato, il danno cresce linearmente con il numero di cicli;
- A parità di spostamento alternato, le curve D-N si spostano su valori di danneggiamento superiori all'aumentare dello spostamento medio;
- A parità di spostamento medio e numero di cicli, il danno cresce con l'aumentare dello spostamento alternato;
- A parità di numero di cicli, le curve D-Za si spostano su valori di danneggiamento superiori all'aumentare dello spostamento medio.

6 Intelligenza artificiale

6.1 Tecnologie alla base dell'IA

Il termine "Intelligenza artificiale" (IA) venne introdotto negli anni Cinquanta del Novecento e si riferisce alla disciplina che consente alle macchine di emulare le funzioni cognitive dell'uomo per risolvere problemi più o meno complessi con procedimenti logici tipici dell'intelligenza umana. Marco Somalvico, ingegnere italiano esperto di IA, ha fornito la seguente definizione di intelligenza artificiale:

" L'intelligenza artificiale è una disciplina appartenente all'informatica che studia i fondamenti teorici, le metodologie e le tecniche che consentono la progettazione di sistemi hardware e sistemi di programmi software capaci di fornire all'elaboratore elettronico prestazioni che, a un osservatore comune, sembrerebbero essere di pertinenza esclusiva dell'intelligenza umana."

L'utilizzo delle tecnologie di IA permette agli elaboratori elettronici di svolgere attività specifiche attraverso l'elaborazione di elevate quantità di informazioni. L'intelligenza artificiale trova impiego in svariati settori dell'ingegneria, della medicina, dell'economia, della finanza e dell'agricoltura. Di seguito si riportano alcuni esempi di applicazione dell'IA:

- Trading algoritmico;
- Guida autonoma;
- Image processing e Computer vision;
- Sicurezza informatica;
- Manutenzione predittiva;
- Shopping online;
- Automatizzazione di processi produttivi;
- Elaborazione del linguaggio;
- Chatbot;
- Identificazione di cellule tumorali.

Le enormi potenzialità e l'incredibile multidisciplinarità dell'intelligenza artificiale hanno spinto numerose aziende ad investire cospicue somme di denaro nel settore, portando il mercato mondiale dell'IA ad uno sviluppo continuo, che è destinato a proseguire ancora per molti anni.

Il vantaggio di utilizzare tecniche di IA per la risoluzione di problemi complessi risiede nella possibilità di analizzare grandi quantità di dati, scoprirne le relazioni nascoste e prendere decisioni sulla base dei risultati forniti dagli algoritmi di IA, che hanno ormai raggiunto ottimi livelli di affidabilità.

Le tecnologie alla base dell'intelligenza artificiale sono:

- Machine Learning;
- Deep Learning;
- Natural Language Processing;
- Computer Vision.

Il machine learning (apprendimento automatico) è una branca dell'IA inteso come "la capacità di una macchina di apprendere senza essere programmata esplicitamente". Si tratta di un processo di analisi dei dati in grado di ricavare automaticamente modelli analitici.

Il deep learning, sottoinsieme del machine learning, elabora le informazioni fornite dall'utente sfruttando metodi di apprendimento ispirati al funzionamento del cervello umano. Infatti, il deep learning è spesso implementato mediante l'impiego di reti neurali profonde.

Il natural language processing è un sottogruppo dell'IA che ha l'obiettivo di insegnare ai computer la comprensione del linguaggio umano scritto e parlato.

La computer vision è un campo dell'IA che permette agli elaboratori elettronici di ricavare informazioni dalla realtà esterna, attraverso l'analisi di immagini digitali o video provenienti da appositi dispositivi di visione.



Figura 6.1: Classificazione di alcune tecniche di intelligenza artificiale

6.2 Machine Learning

L'apprendimento automatico insegna ai computer ad imparare dall'esperienza e dagli errori. Gli algoritmi di machine learning utilizzano metodi computazionali per ricavare informazioni direttamente dai dati di input, senza basarsi su modelli noti a priori. Inoltre, essi migliorano le proprie prestazioni in modo adattivo, in funzione del numero di dati disponibili per l'addestramento. Esistono tre modelli di apprendimento del machine learning:



Figura 6.2: Modelli di apprendimento del machine learning



Figura 6.3: Tecniche di machine learning

Il supervised learning sviluppa modelli predittivi basati sulla conoscenza dei dati di input ed output di un generico problema. In particolare, un algoritmo di supervised learning associa gli input con i corrispondenti output ed allena un modello capace di fornire risposte affidabili quando riceve in input un nuovo dataset.

Il supervised learning impiega le tecniche della classificazione e della regressione:

• Le tecniche di classificazione consentono di effettuare previsioni su grandezze di output discrete. I modelli di classificazione suddividono i dati di input in diverse classi, in funzione delle caratteristiche che li contraddistinguono.

Esempi: stabilire se una mail è genuina oppure spam, comprendere se un tumore è maligno o benigno;

• Le tecniche di regressione consentono di effettuare previsioni su grandezze di output continue. Esempi: variazioni di temperatura di un fluido, fluttuazioni della richiesta di potenza di un veicolo.

Viceversa, gli algoritmi di unsupervised learning trovano schemi nascosti nelle informazioni fornite basandosi esclusivamente sui dati di input, senza essere a conoscenza dei corrispondenti output. La principale tecnica di unsupervised learning è il clustering ed è impiegata per individuare gruppi di dati con caratteristiche simili non note a priori. Tra i classici esempi di applicazione ci sono le ricerche di mercato ed il riconoscimento di oggetti.

Il reinforcement learning è una tecnica di machine learning che si pone l'obiettivo di generare "agenti autonomi" capaci di attuare un'opportuna sequenza di azioni che permettano di ottenere l'output desiderato. Il termine agente autonomo si riferisce ad un qualsiasi dispositivo in grado di interagire con l'ambiente circostante mediante l'utilizzo di sensori ed attuatori, come ad esempio robot o macchine automatiche.

Le azioni compiute dall'agente sono associate ad un valore numerico che esprime la bontà della decisione presa, in modo da incoraggiare lo svolgimento di azioni corrette e disincentivare quelle errate. Al contrario delle altre tipologie di apprendimento automatico, il reinforcement learning si basa sullo scambio dinamico di informazioni tra la realtà esterna e la macchina, il cui addestramento passa inevitabilmente per l'esecuzione di alcune azioni sbagliate: infatti, per ottenere dei miglioramenti il sistema deve fallire.

L'implementazione del reinforcement learning presuppone l'impiego di sistemi di controllo in anello chiuso e di complessi algoritmi per la modellizzazione dei processi decisionali.



Figura 6.4: Schema di funzionamento del reinforcement learning

6.3 Introduzione alle reti neurali

Il deep learning è un sottoinsieme del machine learning che utilizza reti neurali profonde per elaborare grandi quantità di dati con complessi pattern nascosti. Una rete neurale artificiale è un modello computazionale matematico-informatico, il cui funzionamento è ispirato alle reti neurali biologiche del cervello umano. Se quest'ultime sono composte da neuroni, le artificial neural network (ANN) si presentano come un insieme di nodi interconnessi che si scambiano informazioni e modificano il proprio stato a seconda dei segnali che attraversano la rete durante la fase di addestramento. Gli esseri umani percepiscono l'ambiente circostante mediante i recettori sensoriali, i quali raccolgono gli stimoli esterni che sono poi inviati ai neuroni del sistema nervoso; quest'ultimi analizzano i segnali di ingresso e producono una risposta adeguata agli stimoli esterni.

Analogamente, le reti neurali artificiali sono strumenti di modellazione non lineari: ricevono gli input su uno strato di nodi iniziali che elaborano tali informazioni e trasmettono i risultati ai nodi degli strati successivi.

Nell'ambito del progetto di tesi, l'impiego di un'ANN consente di sostituire le analisi agli elementi finiti condotte sulla molla della sospensione ed effettuare previsioni in tempo reale sullo stato di salute del componente.

6.3.1 Struttura

Le unità fondamentali delle reti neurali sono i nodi o neuroni. Essi sono organizzati in layers, ovvero strati composti da uno o più neuroni. Ogni nodo di un layer è connesso con tutti i nodi del layer precedente e di quello successivo. Tutte le connessioni sono caratterizzate da un numero detto peso; analogamente, ad ogni neurone è associato un bias.

In generale, una rete neurale è composta da tre tipologie di strati:

- Input Layer: riceve i dati in ingresso alla rete, pertanto è costituito da un numero di neuroni pari a quello delle variabili di input. I nodi di questo layer sono passivi, ovvero inviano ai nodi dei layers successivi le informazioni ricevute senza modificarne il contenuto;
- Hidden Layers: applicano delle trasformazioni ai dati in arrivo dall'input layer, permettendo l'addestramento della rete. Quest'ultima è tanto più profonda quanto è elevato il numero di hidden layers;
- Output Layer: può essere composto da uno o più nodi in cui vengono restituite le previsioni delle grandezze di output.



Figura 6.5: Struttura delle reti neurali

In figura 6.5 è presentato un esempio di rete neurale artificiale profonda con le seguenti caratteristiche:

- Input layer costituito da 5 nodi;
- 4 hidden layers composti da 7 nodi ciascuno;
- Output layer con 4 nodi.

I segnali di input ricevuti dalla rete sono modificati grazie alla presenza dei neuroni. Per descrivere la modalità di elaborazione delle informazioni confluenti in un nodo si consideri la figura 6.6. Supponiamo che il neurone rappresentato in verde appartenga al primo hidden layer di una generica rete con input layer costituito da *n* nodi. Il segnale di output (y_{pred}) che il neurone verde invia ai nodi del secondo hidden layern è ottenuto nel seguente modo:

- Salvataggio degli input $X_1, X_2, ..., X_n$ nei nodi dell'input layer;
- Moltiplicazione degli input per i rispettivi pesi delle connessioni $w_1, w_2, ..., w_n$;
- Valutazione del parametro z in ingresso al neurone, dove $z = \sum_{i=1}^{n} w_i \cdot x_i + b$;
- Previsione dell'output del neurone: $y_{pred} = f(z)$.



Figura 6.6: Neurone di una rete neurale

Il calcolo della grandezza in uscita dal neurone è effettuato mediante la funzione di attivazione f. Questo passaggio permette di introdurre delle non linearità all'interno della rete. Infatti, se i valori numerici in ingresso ai nodi non fossero elaborati mediante la funzione di attivazione, la rete neurale approssimerebbe le relazioni tra i dati di input ed output con semplici modelli di regressione lineare.

6.3.2 Addestramento

Una rete neurale è in grado di effettuare previsioni affidabili solo a valle di un buon addestramento. Il training consiste nell'attribuire dei valori numerici opportuni ai parametri della rete, ovvero pesi delle connessioni e bias appartenenti ai nodi. Per addestrare una rete neurale è necessario definire:

- Una funzione di costo per confrontare i risultati della rete con quelli attesi;
- Un algoritmo di apprendimento, per le reti neurali si utilizza il back-propagation algorithm;
- Un metodo di ottimizzazione, con le reti si impiegano svariate tecniche di discesa del gradiente.



Figura 6.7: Processo di addestramento della rete neurale

L'addestramento consiste nella ricerca del punto di minimo della funzione di costo ed è caratterizzato dai seguenti passaggi:

- Inizializzazione di pesi e bias con valori random oppure ottenuti casualmente a partire da distribuzioni statistiche;
- Selezione di una riga del dataframe fornito dall'utente, in modo da fornire alla rete i dati di input ed i corrispondenti output attesi;

- Processo di Forward Propagation, ovvero i dati di input attraversano i neuroni degli hidden layers e vengono modificati in funzione dei valori di pesi-bias. Infine, i segnali giungono ai nodi dell'output layer dove vengono restituiti i risultati;
- Confronto dell'output generato dalla rete con quello atteso mediante l'utilizzo di una funzione di costo;
- Processo di Backward Propagation, ovvero i valori di pesi e bias vengono aggiornati percorrendo la rete in verso opposto rispetto a quanto fatto durante la forward propagation. La relazioni che permettono di calcolare i nuovi parametri della rete sono:

$$w_i^{p+1} = w_i^p - \frac{\partial F_C}{\partial w_i} \cdot \eta$$
$$b_i^{p+1} = b_i^p - \frac{\partial F_C}{\partial b_i} \cdot \eta$$

 w_i^{p+1} rappresenta il peso i-esimo all'iterazione p + 1, b_i^{p+1} rappresenta il bias i-esimo all'iterazione p + 1, F_c indica la funzione di costo e η indica il learning rate, un fattore che controlla la velocità di apprendimento della rete;

- Selezione di una nuova riga del dataframe e ripetizione del processo di forward-backward propagation. L'operazione è eseguita fino a processare tutte le righe del dataframe; pesi e bias si aggiornano ad ogni iterazione;
- Il procedimento è ripetuto nuovamente su tutte le righe del dataframe, finché le previsioni della rete neurale non risultano sufficientemente accurate.

In realtà, la frequenza con cui vengono aggiornati pesi e bias dipende dal metodo di ottimizzazione scelto per il training. Inoltre, l'addestramento di una rete neurale richiede di impostare molteplici parametri caratteristici per ottenere dei risultati affidabili e precisi.



Figura 6.8: Algoritmo di forward-backward propagation

6.3.3 Parametri

I parametri da impostare in Python per creare una rete neurale ed addestrarla sono riportati nelle tre tabelle seguenti.

PARAMETRO RETE NEURALE	DESCRIZIONE		
Neurone	Unità fondamentale della rete		
Layer	Gruppo di neuroni appartenenti al medesimo livello		
Activation function	Funzione che riceve in ingresso la somma pesata degli input ad un neurone, permette di introdurre delle non linearità nella rete neurale		
Kernel initializer	Distribuzione statistica utilizzata per stabilire il valore iniziale dei pesi della rete		
Bias initializer	Distribuzione statistica utilizzata per stabilire il valore iniziale dei bias della rete		
Kernel regularizer	Regolarizzatore adottato per ridurre l'overfitting, si aggiungono delle penalità alla funzione di costo a partire dai valori dei pesi della rete		
Bias regularizer	Regolarizzatore adottato per ridurre l'overfitting, si aggiungono delle penalità alla funzione di costo a partire dai valori dei bias della rete		
Activity regularizer	Regolarizzatore adottato per ridurre l'overfitting, si aggiungono delle penalità agli output di ogni layer agendo sui valori di pesi e bias		
Dropout	Percentuale di neuroni che viene disattivata in ogni layer durante il training, permette di evitare l'overfitting e di generalizzare l'addestramento della rete		

Tabella 6.1: Parametri per la generazione di una rete neurale

PARAMETRO RETE NEURALE	DESCRIZIONE		
Loss function	Funzione che permette di confrontare le previsioni della rete con i risultati attesi. Durante il training, minimizzando la loss function si aggiornano pesi e bias della rete		
Optimizer	Algoritmo di ottimizzazione che durante il training della rete permette di trovare la combinazione di pesi e bias tale da minimizzare la loss function		
Learning rate Controlla la velocità di apprendimento della rete, stabilendo di quanto aggiornati pesi e bias durante l'addestramento			

Tabella 6.2: Parametri per la configurazione di una rete neurale

PARAMETRO RETE NEURALE	DESCRIZIONE		
Batch size	Numero di righe del database (samples) che vengono processate prima di confrontare le previsioni della rete con gli output attesi, per poi aggiornare i valori di pesi e bias		
Epochs	Numero di volte che l'algoritmo di addestramento processa l'intero database; le iterazioni in un'epoca sono pari al rapporto tra il numero di samples e il batch size		

Tabella 6.3: Parametri per l'addestramento di una rete neurale

6.3.3.1 Activation functions

Le funzioni di attivazione hanno il compito fondamentale di introdurre delle non linearità nel modello della rete neurale, applicando delle trasformazioni alle somme pesate degli input diretti a tutti i neuroni degli hidden ed output layers. Inoltre, l'utilizzo di alcune activation function consente di mantenere l'uscita dei nodi all'interno di intervalli limitati: tale operazione permette di migliorare i risultati dell'addestramento delle reti.

In tabella 6.4 si riportano sinteticamente alcuni esempi di funzioni di attivazione.

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \ge \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \le -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks	\rightarrow
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0,z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	

Tabella 6.4: Esempi di activation functions

Nell'ambito del progetto di tesi, le tipologie di activation functions impiegate in Python durante le molteplici prove di training della rete sono riassunte di seguito:

- Sigmoide;
- Softplus;
- Relu;
- Leaky Relu;
- Tangente Iperbolica;
- Softsign.

Funzione Sigmoide:



Figura 6.9: Funzione di attivazione "Sigmoide"

La funzione Sigmoide limita gli output dei neuroni ad assumere un valore compreso tra 0 ed 1. In particolare, per input superiori a 5 l'output è prossimo ad 1, mentre per input inferiori a -5 l'output è vicino allo 0. Questa activation function potrebbe causare una convergenza lenta durante il training; inoltre, è computazionalmente costosa. Spesso è impiegata per problemi di classificazione binaria.

Funzione Softplus:



Figura 6.10: Funzione di attivazione "Softplus"

La funzione restituisce degli output prossimi allo zero per valori di input inferiori a -4. Nonostante sia computazionalmente costosa permette una buona convergenza del modello in fase di training.

Funzione Relu:



Figura 6.11: Funzione di attivazione "Relu"

È una delle activation function maggiormente impiegate nel campo delle reti neurali poiché è poco onerosa dal punto di vista computazionale. Infatti, se l'argomento della funzione è negativo viene restituito un output nullo: in fase di training alcuni neuroni vengono disattivati. Solitamente garantisce una buona convergenza del modello durante l'addestramento.

Funzione Leaky Relu:



Figura 6.12: Funzione di attivazione "Leaky Relu"

Questa funzione è vista come la versione successiva della Relu. Infatti, la Leaky Relu non annulla gli output dei neuroni che presentano un input negativo, garantendo l'aggiornamento di tutti i pesi ed i bias durante il training.

Funzione Tangente Iperbolica:



Figura 6.13: Funzione di attivazione "Tangente iperbolica"

Ha caratteristiche simili alla funzione Sigmoide. Limita l'uscita a valori compresi tra -1 ed 1. Funzione Softsign:



Figura 6.14: Funzione di attivazione "Softsign"

Computazionalmente costosa ma solitamente non causa problemi di convergenza del modello.

6.3.3.2 Kernel e bias initializers

Inizializzare manualmente pesi e bias è un'operazione sconsigliata. Infatti, l'algoritmo di apprendimento è sensibile ai valori iniziali assunti dai parametri della rete; un'impostazione inadeguata potrebbe comportare lunghi tempi di addestramento oppure una mancata convergenza del modello. Per tali ragioni, pesi e bias sono inizializzati mediante distribuzioni statistiche. Le principali tipologie di kernel e bias initializer disponibili in Python sono:

• RandomNormal: pesi e bias sono inizializzati mediante una distribuzione normale con media μ e deviazione standard σ definite dall'utente;



Figura 6.15: Esempi di distribuzioni normali

- GlorotNormal: pesi e bias sono inizializzati mediante una distribuzione normale troncata con media $\mu = 0$ e deviazione standard $\sigma = \sqrt{\frac{2}{f_{in} + f_{out}}}$, dove f_{in} è il numero di unità in ingresso nella matrice dei pesi e f_{out} è il numero di unità in uscita;
- GlorotUniform: pesi e bias sono inizializzati mediante una distribuzione uniforme con valori compresi nell'intervallo [-a, a], dove $a = \sqrt{\frac{6}{f_{in} + f_{out}}}$;
- HeNormal: pesi e bias sono inizializzati mediante una distribuzione normale troncata con media $\mu = 0$ e deviazione standard $\sigma = \sqrt{\frac{2}{f_{in}}}$;
- HeUniform: pesi e bias sono inizializzati mediante una distribuzione uniforme con valori compresi nell'intervallo [-a, a], dove $a = \sqrt{\frac{6}{f_{in}}}$.

6.3.3.3 Tecniche di regolarizzazione

della rete.

Nelle reti neurali profonde un problema abbastanza ricorrente è l'overfitting, ovvero un eccessivo adattamento del modello ai dati utilizzati per l'addestramento. Le tecniche di regolarizzazione nascono per prevenire tale fenomeno, permettendo alla rete di restituire risultati corretti quando è richiesto di effettuare previsioni partendo da dataset differenti da quelli utilizzati durante la fase di training. Di conseguenza, il modello non impara mnemonicamente le relazioni tra input ed output del database di addestramento ma è in grado di generalizzare su un qualsiasi set di dati fornito. Le tecniche di regolarizzazione disponibili in Python sono:

• Dropout: durante l'addestramento viene inibita una determinata percentuale di nodi per gli hidden layer che costituiscono la rete; per ogni iterazione all'interno di un'epoca, i neuroni disattivati sono selezionati casualmente. In questo modo, i risultati del training possono essere visti come la media dei risultati di differenti reti neurali addestrate in parallelo;



• Regolarizzazione L1: si aggiunge una penalità alla funzione di costo selezionata per la valutazione degli errori. Tale penalità è rappresentata dalla somma dei valori assoluti dei pesi

$$F_C = f_C + \lambda \cdot \sum_{i=1}^N |w_i|$$

 F_C indica la nuova funzione di costo, f_C è la loss function originale e λ rappresenta il fattore di regolarizzazione. Un elevato valore di λ potrebbe portare i pesi ad un valore prossimo allo zero, causando underfitting ed impedendo alla rete di cogliere i pattern presenti nei dati;

• Regolarizzazione L2: si aggiunge una penalità alla funzione di costo selezionata per la valutazione degli errori. Tale penalità è rappresentata dalla somma dei quadrati dei pesi della rete.

$$F_C = f_C + \lambda \cdot \sum_{i=1}^N w_i^2$$

6.3.3.4 Funzioni di costo

La scelta della loss function è un passaggio fondamentale per la costruzione adeguata di una rete neurale. Queste funzioni rappresentano un valido strumento di valutazione dei risultati del modello, in quanto permettono di confrontare le grandezze di output previste della rete con quelle attese. Inoltre, l'addestramento del modello si basa sulla minimizzazione della funzione di costo, che consente di aggiornare opportunamente i valori di pesi e bias durante il processo di backward propagation.

Esistono due categorie fondamentali di funzioni di costo:

- Regression loss function: utilizzate nei problemi di regressione, dove gli output del modello possono assumere un qualsiasi valore all'interno di un intervallo continuo;
- Classification loss function: impiegate nei problemi di classificazione, dove è richiesto di assegnare un pattern alla classe corretta. Si precisa che un pattern è un campione di dati da cui estrarre delle informazioni, mentre una classe è un insieme di pattern con caratteristiche comuni. Sovente, nei problemi di classificazioni gli output prodotti dalla rete neurale sono dei vettori che indicano la probabilità di appartenenza dei dati di input a determinate categorie specificate a priori.

Di seguito si riportano le principali funzioni di costo di regressione:

 Mean Squared Error (MSE): definita come la media delle differenze al quadrato tra i valori obiettivo e quelli predetti. Elevare gli errori al quadrato rende questa loss function particolarmente sensibile alla presenza di outliers. Nell'espressione della funzione di costo, y_{t,i} indica il risultato target i-esimo e y_{r,i} rappresenta

Nell'espressione della funzione di costo, $y_{t,i}$ indica il risultato target i-esimo e $y_{r,i}$ rappresenta il risultato i-esimo predetto dalla rete;

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_{t,i} - y_{r,i})^2$$

• Mean Absolute Error (MAE): definita come la media dei valori assoluti delle differenze tra i valori obiettivo e quelli predetti. È particolarmente usata quando i database di addestramento contengono molti outliers;

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_{t,i} - y_{r,i}|$$

• Mean Absolute Percentage Error (MAPE): definita come la media percentuale dei valori assoluti degli errori relativi tra i risultati attesi e quelli predetti. Considerando gli errori relativi, questa loss function permette all'utente di valutare criticamente la bontà dell'addestramento;

$$MAPE = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{y_{t,i} - y_{r,i}}{y_{t,i}} \right|$$

• Mean Squared Logarithmic Error (MSLE): definita come la media degli errori logaritmici tra valori obiettivo e predetti al quadrato. Si consiglia di impiegare questa funzione di costo quando si lavora con dati i cui valori hanno diversi ordini di grandezza di differenza. In tal modo, si evita di punire eccessivamente il modello in presenza di grandi errori.

$$MSLE = \frac{1}{N} \sum_{i=1}^{N} \left[\log(y_{t,i} + 1) - \log(y_{r,i} + 1) \right]^{2}$$

Un esempio di funzione di costo di classificazione è la Binary Cross Entropy (BCE):

$$BCE = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot log(p_{y_i}) + (1 - y_i) \cdot log(1 - p_{y_i})$$

Nell'espressione della Binary Cross Entropy y_i indica l'i-esimo input, mentre p_{y_i} rappresenta la probabilità di y_i di ricadere in una specifica categoria.

Infatti, la BCE loss function è adottata nei modelli di classificazione binaria, il cui obiettivo è determinare a quale delle due classi definite dall'utente appartengono i dati di input.

La funzione di costo permette di costruire la cosiddetta curva di apprendimento, un importante grafico che consente di visionare le performance della rete durante la fase di training. Prima di descrivere la learning curve è necessario introdurre le definizioni di training set, validation set e test set.

Ogni modello basato su una rete neurale riceve dall'utente un database composto da due matrici: la prima contiene le grandezze di input e la seconda i corrispondenti output. Per effettuare l'addestramento non si utilizzano tutte le righe del database ma solamente una porzione prestabilita. Infatti, entrambe le matrici sono divise in tre parti:

- Training set: porzione del database impiegata per il training della rete neurale, si ricercano i valori di pesi e bias che minimizzano la loss function. A partire da questi dati, il modello cerca di apprendere la relazione che lega gli input agli output;
- Validation set: porzione del database adottata per verificare la qualità delle previsioni della rete a partire da dati di input che non conosce. Si valutano le prestazioni del modello durante l'impostazione degli iperparametri;
- Test set: porzione del database utilizzata per una validazione conclusiva della rete. Anche in questo caso, vengono sottoposti al modello dei dati ignoti ma caratterizzati dalla medesima distribuzione di probabilità del training set.

Si sottolinea che sia i dati appartenenti al training set, che quelli relativi al validation set sono sfruttati per l'addestramento della rete neurale.

Si precisa che gli iperparametri sono delle variabili di configurazione esterne al modello che vengono impostate dall'utente. Esempi di iperparametri sono il learning rate, il batch size ed il numero di epoche.

A pagina seguente si riporta uno schema che chiarifica la suddivisione del database di una rete neurale.


Figura 6.17: Suddivisione del database

A questo punto è opportuno introdurre la learning curve, ossia un grafico che riporta gli andamenti delle funzioni di costo relative al training set e al validation set, visibile dall'utente una volta terminata la fase di addestramento della rete.



Figura 6.18: Esempio di curva di apprendimento

Un modello addestrato correttamente è caratterizzato dalla pressoché totale sovrapposizione delle curve relative alla training loss function e alla validation loss function; viceversa, se gli andamenti delle due curve si discostano notevolmente è opportuno modificare gli iperparametri della rete e tentare un nuovo addestramento.

A pagina seguente si riportano alcuni esempi di learning curve e le considerazioni che possono essere effettuate a valle della loro analisi.

• Underfitting: il modello non è in grado di prevedere gli output in modo affidabile. La funzione di costo relativa al training set è praticamente orizzontale, sintomo che la rete non migliora il proprio comportamento con il passare delle epoche.

Il fenomeno dell'underfitting è identificabile anche nel caso di loss function con valori elevati. Per contrastare questa tendenza è consigliato aggiungere nuovi dati al database di partenza, aumentare il numero di epoche oppure modificare il learning rate;



Figura 6.19: Esempio di learning curve con underfitting

• Overfitting: la rete ha estratto adeguatamente le relazioni che legano gli input e gli output del training set, ma non è capace di restituire risultati altrettanto corretti quando deve analizzare un nuovo dataset. Osservando la learning curve si nota che la training loss function decresce fino ad un valore minimo, mentre la validation loss function continua ad aumentare con il passare delle epoche.

Per evitare l'overfitting è opportuno modificare il learning rate o la struttura della rete. In alternativa si può usare una tecnica di regolarizzazione, ad esempio il dropout;



Figura 6.20: Esempio di learning curve con overfitting

• Training set non rappresentativo: il training set non fornisce al modello informazioni sufficienti per comprendere le relazioni che legano gli input con gli output. Questo fenomeno è principalmente causato dalle differenze tra i parametri statistici che descrivono le distribuzioni di dati del training set e del validation set. Per evitarlo è conveniente incrementare la dimensione del database ed assicurarsi di attribuire i dati al training set e al validation set in modo randomico. Osservando la learning curve si nota che entrambe le funzioni di costo decrescono con l'aumentare delle epoche, ma è presente un elevato gap tra le due curve;



Figura 6.21: Esempio di learning curve con training set non rappresentativo

• Validation set non rappresentativo: il validation set non consente alla rete di generalizzare le previsioni a causa di informazioni insufficienti. In alcuni casi il modello riesce a prevedere più facilmente gli output del validation set rispetto a quelli del training set. Di conseguenza, la funzione di costo relativa al validation set risiede sempre al di sotto della loss function corrispondente al training set.



Figura 6.22: Esempio di learning curve con validation set non rappresentativo

6.3.3.5 Algoritmi di ottimizzazione

Gli algoritmi di ottimizzazione, anche detti optimizers, sono fondamentali per l'addestramento di una rete neurale in quanto permettono di aggiornare pesi e bias, di minimizzare la funzione di costo e di migliorare l'accuratezza del modello. Gli optimizers comunemente impiegati nell'ambito del deep learning basano il loro funzionamento sulla tecnica di discesa del gradiente, ovvero un metodo numerico per la ricerca e l'individuazione dei punti di minimo o massimo di una funzione a più variabili.

Il "gradient descent algorithm" consente di minimizzare una loss function $F_C(\vartheta)$, descritta con i parametri ϑ della rete neurale (pesi e bias), mediante l'aggiornamento delle variabili in direzione opposta al gradiente della funzione di costo calcolato rispetto ai parametri stessi.

Si ricorda che il tasso di variazione delle variabili durante il training è determinato dal valore del learning rate η , che risulta cruciale anche per la convergenza del modello.



Figura 6.23: Visualizzazione dell'algoritmo di discesa del gradiente su una funzione quadratica a due variabili

I principali algoritmi di ottimizzazione disponibili in Python sono:

• Batch gradient descent: i parametri vengono aggiornati una sola volta dopo aver calcolato il gradiente della funzione di costo per ogni riga del database. Di conseguenza, è lecito aspettarsi lunghi tempi di calcolo ed un elevato costo computazionale; pertanto, è una tecnica sconsigliata se il database presenta ampie dimensioni. Inoltre, il batch gradient descent non garantisce la convergenza in un punto di minimo globale della loss function;

$$\vartheta' = \vartheta - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} \nabla_{\vartheta_i} F_C(\vartheta_i)$$

• Stochastic gradient descent: i parametri vengono aggiornati dopo aver processato una singola riga del database, pertanto il tempo di calcolo e il costo computazionale sono minori rispetto al batch gradient descent. È una tecnica consigliata per trattare elevate quantità di dati ed evitare il problema dell'intrappolamento in un minimo locale.

In generale, la convergenza in un minimo globale della loss function non è garantita;

- Mini-batch gradient descent: i parametri vengono aggiornati dopo aver valutato i gradienti di *n* righe del database. È un algoritmo che presenta caratteristiche intermedie rispetto al batch gradient descent e allo stochastic gradient descent;
- Stochastic gradient descent (SGD) con Momentum: è un algoritmo che aggiunge un termine addizionale nell'equazione di aggiornamento dei parametri, considerando così anche i valori assunti dai gradienti nelle iterazioni precedenti. Riducendo le fluttuazioni dei gradienti lungo le direzioni irrilevanti viene accelerata la convergenza verso il punto di minimo della funzione di costo. Per utilizzare questa tecnica di ottimizzazione è necessario impostare un nuovo iperparametro, ovvero il momentum rate γ (solitamente variabile tra 0 e 1);

$$\vartheta_{i+1} = \vartheta_i - \eta \cdot \nabla_{\vartheta_i} F_C(\vartheta_i) + \gamma \cdot \Delta \vartheta_i$$

 $\Delta \vartheta_{i} = \vartheta_{i} - \vartheta_{i-1} = \gamma \cdot \Delta \vartheta_{i-1} - \eta \cdot \nabla_{\vartheta_{i-1}} F_{\mathcal{C}}(\vartheta_{i-1})$



Figura 6.24: Influenza del momentum sulla velocità di convergenza del modello

• Adagrad: è un algoritmo di ottimizzazione del secondo ordine che cambia il valore del learning rate in funzione del tasso di variazione dei singoli parametri della rete. Maggiore è la velocità di aggiornamento di un parametro, minore è il relativo learning rate e viceversa.

$$\vartheta_{t+1,i} = \vartheta_{t,i} - \frac{\eta_i}{\sqrt{G_{t,ii} + \epsilon}} \cdot \nabla_{\vartheta_{t,i}} F_C(\vartheta_{t,i})$$

 $\vartheta_{t,i}$ indica l'i-esimo parametro della rete al tempo t, ϵ è un numero che evita la divisione per zero e $G_{t,ii}$ è una matrice diagonale i cui elementi sono pari alla somma dei quadrati dei gradienti calcolati rispetto a $\vartheta_{t,i}$ fino all'istante temporale t.

La presenza del termine $G_{t,ii}$ comporta una drastica riduzione dei learning rates con il procedere delle iterazioni, rallentando il training della rete neurale;

• RMSprop - Adadelta: questi algoritmi sono nati per superare il principale limite di Adagrad, ovvero la netta diminuzione dei learning rates con il passare delle epoche. Invece di accumulare la somma dei quadrati di tutti i gradienti, i parametri del modello sono aggiornati considerando solo una parte di essi. Lo svantaggio di tali tecniche è l'elevato costo computazionale; Adam (Adaptive Moment Estimation): è uno degli algoritmi di ottimizzazione maggiormente usati nell'ambito del deep learning. Combina in un unico optimizer le caratteristiche di Adagrad, RMSprop e SGD con momentum: durante l'addestramento, le variabili interne sono modificate tenendo conto dei valori dei gradienti delle precedenti iterazioni, introducendo sia termini lineari che quadratici nell'equazione ricorsiva di aggiornamento di pesi e bias. Adam permette una rapida convergenza del modello, è computazionalmente efficiente, non necessita di impostare molti iperparametri, è adatto a processare database con elevate dimensioni e richiede piccole quantità di memoria.

$$\begin{split} m_t &= \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla_{\vartheta_t} F_C(\vartheta_t) \\ v_t &= \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla_{\vartheta_t} F_C(\vartheta_t))^2 \\ \widehat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \widehat{v}_t &= \frac{v_t}{1 - \beta_2^t} \\ \vartheta_{t+1} &= \vartheta_t - \frac{\eta}{\sqrt{\widehat{v}_t + \epsilon}} \cdot \widehat{m}_t \end{split}$$

I termini $m_t e v_t$ rappresentano rispettivamente la media mobile esponenziale dei gradienti e la media mobile esponenziale dei gradienti al quadrato (al tempo t); $\beta_1 e \beta_2$ sono dei decay rates, il cui valore è definibile dall'utente (solitamente $\beta_1 = 0.9 e \beta_2 = 0.999$). L'ultima equazione indica la legge di aggiornamento dei parametri della rete.

A titolo di esempio, si riportano in figura 6.25 gli andamenti della validation loss relativi ad una rete neurale addestrata sul database MNIST (Modified National Institute of Standards and Technology).



Figura 6.25: Influenza degli optimizers sull'addestramento di una rete neurale impiegata per classificazione di immagini

In conclusione, l'algoritmo di discesa del gradiente può essere riassunto come segue.

In una generica rete neurale profonda siano $\vartheta(w, b)$ e F_c i parametri del modello e la funzione di costo, rispettivamente. Durante il training, l'aggiornamento delle variabili è ottenuto a partire dall'opposto del gradiente della funzione di costo calcolato rispetto ai parametri stessi. Il gradiente di una generica funzione a più variabili è diretto lungo la sua direzione di massima crescita; pertanto, l'introduzione del segno meno è necessaria per ricercare il punto di minimo della loss function. Si ricorda che l'obiettivo dell'addestramento è trovare i valori dei parametri del modello che minimizzano gli errori tra output attesi e output previsti dalla rete. In simboli matematici:

$$\Delta \vartheta = \begin{pmatrix} \Delta w_{jk}^{(l)} \\ \cdots \\ \Delta b_{j}^{(l)} \\ \cdots \end{pmatrix}; \ \nabla F_{c} = \begin{pmatrix} \frac{\partial F_{c}}{\partial w_{jk}^{(l)}} \\ \frac{\partial F_{c}}{\partial b_{j}^{(l)}} \\ \cdots \end{pmatrix}; \ \Delta \vartheta = -\eta \cdot \nabla F_{c}$$
$$w_{jk}^{\prime(l)} = w_{jk}^{(l)} - \eta \cdot \frac{\partial F_{c}}{\partial w_{jk}^{(l)}}$$
$$b_{j}^{\prime(l)} = b_{j}^{(l)} - \eta \cdot \frac{\partial F_{c}}{\partial b_{j}^{(l)}}$$

 $w_{jk}^{(l)}$ indica il peso relativo al k-esimo input del *j*-esimo neurone appartenente al layer *l*, $b_j^{(l)}$ rappresenta il bias del *j*-esimo neurone appartenente al layer *l* ed η è il learning rate.



Figura 6.26: Ricerca del minimo globale di una funzione a più variabili

6.3.3.6 Learning rate

Il learning rate η è l'iperparametro che controlla la velocità di apprendimento della rete in quanto definisce il tasso di variazione di pesi e bias durante il training; solitamente assume dei valori compresi tra 10^{-6} e 1. Un elevato learning rate consente di ridurre i tempi di addestramento ma potrebbe causare una mancata convergenza del modello; viceversa, un valore troppo piccolo porterebbe ad un eccessivo incremento delle iterazioni necessarie a raggiungere il minimo globale della funzione di costo.

Osservando la figura 6.27, si nota che un learning rate ottimale conduce ad un compromesso tra lunghi tempi di calcolo e affidabilità della rete nell'effettuare previsioni.



Figura 6.27: Influenza del learning rate sulla ricerca del punto di minimo della loss function

L'effetto del learning rate sulla loss function di una generica rete è visibile in figura 6.28. Inizialmente la funzione di costo è praticamente orizzontale, sintomo che il tasso di apprendimento della rete è troppo basso. Successivamente inizia a decrescere: in questo intervallo il learning rate assume dei valori ottimali. Infine, la loss function è caratterizzata da forti oscillazioni a causa di un learning rate molto alto.



Figura 6.28: Influenza del learning rate sull'andamento della loss function

Il tasso di apprendimento è uno degli iperparametri più difficili da impostare. Per agevolarne il tuning, oltre agli optimizers con learning rate adattativo, Python mette a disposizione dell'utente alcune funzioni che ne modificano automaticamente il valore, come ad esempio:

- ExponentialDecay: ogni n iterazioni definite dall'utente il valore di η viene aggiornato mediante una funzione di decadimento esponenziale;
- PiecewiseConstantDecay: per ogni intervallo di epoche specificate dall'utente vengono assegnati differenti valori di η ;
- PolynomialDecay: ogni n iterazioni definite dall'utente il valore di η viene aggiornato mediante una funzione di decadimento polinomiale.

Un'altra funzione presente in Python e comunemente impiegata nell'ambito del deep learning è chiamata "**ReduceLROnPlateau**": il valore di η viene ridotto ogni volta che una determinata metrica non presenta miglioramenti. Per utilizzarla è necessario impostare i seguenti parametri:

- Monitor: rappresenta la variabile da monitorare, ad esempio la validation loss function;
- Factor: coefficiente k di riduzione del learning rate, $\eta' = k \cdot \eta \ (k < 1)$;
- Patience: numero di epoche senza miglioramenti dopo il quale viene ridotto il learning rate;
- Min_delta: cambiamento minimo della variabile monitorata per qualificare un miglioramento.

Infine, si riportano le curve di apprendimento derivanti da training effettuati con diversi valori di learning rate. Si ricorda che il caso di studio riguarda la stima del danneggiamento subito dalla molla di una sospensione anteriore. I parametri di input ed output del modello sono visibili nella tabella 5.3 del paragrafo 5.4.1. Al momento, si tralasciano i dettagli riguardanti la struttura e gli iperparametri della rete per valutare esclusivamente l'incidenza del learning rate sull'esito degli addestramenti.



Figura 6.29: Influenza del learning rate sull'addestramento della rete

Si nota che per $\eta = 0,1$ gli andamenti delle loss functions presentano notevoli oscillazioni senza giungere a convergenza. Per $\eta = 10^{-6}$ le funzioni di costo hanno valori estremamente elevati e non garantiscono risultati soddisfacenti. Il giusto compromesso è rappresentato da un learning rate pari a 0,001: le funzioni di costo di training e validation sono ottimamente sovrapposte, con errori minimi tra output attesi e output predetti dalla rete neurale.

6.3.3.7 Batch size ed epochs

Il batch size è un iperparametro che indica il numero di righe del database processate prima di confrontare le previsioni della rete con gli output di riferimento, per poi aggiornare i valori di pesi e bias. Nei problemi caratterizzati da database di elevate dimensioni è sconsigliato adottare il batch gradient descent come algoritmo di ottimizzazione, pertanto il training set viene diviso in diversi sottogruppi, ognuno con dimensione pari al batch size. Se da un lato l'adozione del mini-batch gradient descent permette di ridurre il costo computazionale dell'addestramento, dall'altro la stima del gradiente della funzione di costo dopo la forward propagation potrebbe essere poco accurata, soprattutto per piccoli valori di batch size.

Analogamente al paragrafo precedente, si mostrano gli andamenti delle learning curves derivanti da addestramenti effettuati con diversi valori di batch size. Anche in questo caso, si tralasciano i dettagli riguardanti la struttura e gli iperparametri della rete. Si precisa solamente che l'algoritmo di ottimizzazione adottato è Adam e il numero di righe del database è pari a 4000.



Figura 6.30: Influenza del batch size sull'addestramento della rete

Si osserva che incrementando il valore del batch size diminuiscono i tempi di training della rete, si regolarizzano gli andamenti delle funzioni di costo e si ottiene una buona convergenza del modello.

L'ultimo iperparametro di interesse nell'ambito delle reti neurali è rappresentato dalle epoche, che indicano il numero di volte che l'algoritmo di training processa l'intero database; le iterazioni in un'epoca sono pari al rapporto tra il numero di samples (righe del database) e il batch size. Solitamente, il numero di epoche per un addestramento corretto dipende dalla complessità dei pattern celati nei dati che si richiede alla rete di individuare: le epoche necessarie per ottenere un modello affidabile incrementano con l'ampiezza e la disomogeneità del database. Tuttavia, un numero eccessivamente alto potrebbe condurre alla comparsa dell'overfitting; viceversa, l'impiego di poche epoche potrebbe causare underfitting. Analogamente al learning rate, il tuning di questo iperparametro è particolarmente ostico per l'utente. Per tale ragione, in Python è disponibile la funzione "EarlyStopping", che permette di interrompere l'addestramento della rete quando non si verificano miglioramenti nell'andamento di una metrica predefinita. Per utilizzarla è necessario impostare i seguenti parametri:

- Monitor: rappresenta la variabile da monitorare, ad esempio la validation loss function;
- Min_delta: cambiamento minimo della variabile monitorata per qualificare un miglioramento;
- Patience: numero di epoche senza miglioramenti dopo il quale viene terminato il training;
- Start_from_epoch: numero di epoche da attendere prima di iniziare a monitorare l'andamento della metrica selezionata.

6.4 Modello di danneggiamento predittivo del componente

L'obiettivo del progetto di tesi è lo sviluppo di una metodologia per controllare in tempo reale lo stato di salute di un componente meccanico. In particolare, lo studio si concentra sulle valutazioni dei danneggiamenti subiti dalla molla della sospensione anteriore durante la marcia del veicolo di prova. L'impiego di tecniche di deep learning si sposa perfettamente con la necessità di effettuare previsioni sul comportamento del componente durante il suo funzionamento.

Di seguito si riporta una tabella che riassume le operazioni da compiere per poter usufruire delle predizioni di una rete neurale costruita in Python.

Passo	Azione	Strumento
1	Creazione del database	Simulazioni FEM in Optistruct e Femfat
2	Upload del database	Libreria Pandas di Python
3	Pre-processing dei dati di input e di output	Libreria Scikit-Learn di Python
4	Costruzione della rete	
5	Configurazione della rete	Libraria Karas di Duthan
6	Addestramento della rete	Librena Keras di Python
7	Salvataggio della rete	
8	Analisi delle prestazioni della rete	
9	Uso della rete correttamente addestrata	Script Python e libreria Keras
10	Post-processing dei risultati	

Tabella 6.5: Step da seguire per l'utilizzo della rete neurale

6.4.1 Creazione del database e pre-processing dei dati

Il primo step è relativo alla creazione del database, elemento imprescindibile in qualsiasi modello di intelligenza artificiale. Nei capitoli dedicati alle simulazioni agli elementi finiti sono descritte dettagliatamente le grandezze che permettono di monitorare lo stato della molla. Si rammenta che le storie di carico sperimentali descrivono lo spostamento verticale del componente in funzione del tempo; una tecnica di conteggio consente di ridurle alla somma di singoli di cicli di sollecitazione caratterizzati dai seguenti parametri:

- Z_m [mm]: spostamento medio;
- *Z_a* [*mm*]: spostamento alternato;
- *n*: numero di cicli;
- *D*: danneggiamento della molla, grandezza adimensionale compresa tra 0 e 1.

Gli input di ogni simulazione FEM sono Z_m , Z_a e n, mentre la grandezza di uscita è il danno prodotto dal relativo ciclo di sollecitazione. Si sottolinea che il database è stato generato con un processo automatizzato in Python che prevede i seguenti passaggi:

- Definizione di tre vettori contenenti i valori degli spostamenti medi, alternati e del numero di cicli (le tre grandezze possono variare all'interno degli intervalli riportati nella tabella 5.5 del paragrafo 5.5.1);
- Selezione di una combinazione casuale dei tre parametri di input;
- Modifica dei file .ffj, .max e .his di FEMFAT;
- Esecuzione del run dell'analisi a fatica in batch mode;
- Lettura del file .pro in cui sono contenuti i risultati; da qui si ricava il valore del danneggiamento subito dalla molla;

Al termine del processo il database per la rete neurale è costituito dalle matrici [I] e [O]: nella prima sono salvati i dati di input e nella seconda quelli di output. Le loro dimensioni sono:

- $[I] \in \mathbb{R}^{N_s X 3}$
- $[0] \in \mathbb{R}^{N_s X \mathbf{1}}$

Si precisa che N_s indica il numero totale di simulazioni effettuate.

PARAMETRI DI INPUT			GRANDEZZA DI OUTPUT
Z _m [mm]	Z₁ [mm]	n	D
Zm1	Zal	n 1	D1
Zm2	Za2	n ₂	D2
Zm3	Za3	n 3	D3
Zmi	Zai	ni	Di
Zmk	Zak	Nk	Dk

Tabella 6.6: Input e output della rete neurale

Le matrici $[I] \in [O]$ sono salvate come file PKL e vengono importate in Python mediante una funzione della libreria Pandas. Per ottenere risultati soddisfacenti è necessario effettuare un pre-processing dei dati. Infatti, la rete non è in grado di individuare i pattern nascosti nel database se il suo contenuto non è opportunamente scalato.

I dati di input sono normalizzati nell'intervallo [a, b] = [0,1] con la seguente relazione:

$$i_s = a + \frac{(b-a) \cdot (i_v - i_{min})}{i_{max} - i_{min}}$$

 i_s indica il nuovo input scalato, i_v il valore vero dell'input, i_{min} e i_{max} rappresentano rispettivamente il valore minimo e massimo assunto dalle grandezze di input nel database.

La trasformazione dei dati di input appena descritta è compiuta mediante uno scaler disponibile nella libreria Scikit-Learn di Python, che viene salvato per essere successivamente utilizzato in combinazione con la rete neurale addestrata.

I danneggiamenti o_v subiti dalla molla sono trasformati con una funzione logaritmica per ottenere gli output scalati o_s , che assumono dei valori compresi tra 0 e 10.

$$o_s = -log(o_v)$$

Le matrici $[I] \in [O]$ sono divise in tre parti per formare il training set, il validation set e il test set, come mostrato nello schema di figura 6.31.



Figura 6.31: Divisione del database di progetto

In definitiva, il 60% dei dati è assegnato al training set e il rimanente 40% è equamente ripartito tra validation set (20%) e test set (20%). Tale suddivisione è piuttosto tipica nell'ambito delle reti neurali profonde.

6.4.2 Costruzione e utilizzo della rete neurale

I modelli predittivi sono realizzati mediante i seguenti metodi forniti dalla libreria Keras di Python:

- *Model.add*: consente di costruire input layer e output layer della rete, nonché di aggiungere hidden layers in sequenza. Per ogni hidden layer è possibile specificare il numero di nodi, la funzione di attivazione e la percentuale di dropout. Inoltre, si definiscono le tipologie di kernel e bias initializer per inizializzare i parametri interni del modello;
- *Model.compile*: permette di scegliere il tipo di loss function e di algoritmo di ottimizzazione;
- *Model.fit*: metodo per l'addestramento della rete, si impostano i valori di batch size ed epochs. Inoltre, è possibile settare i parametri caratteristici delle funzioni "EarlyStopping" e "ReduceLROnPlateau";
- *Model.save*: a valle del training, tale metodo consente di salvare il modello in un file dall'estensione .h5;
- *Model.predict*: permette di utilizzare una rete addestrata per ottenere gli output relativi a nuovi dati di input.

La valutazione delle prestazioni di un modello è effettuata mediante delle righe di codice che forniscono l'errore relativo medio, l'errore assoluto medio e l'errore assoluto massimo tra gli output del test set e quelli previsti dalla rete. Inoltre, un processo automatizzato in Python consente di confrontare i risultati degli addestramenti relativi a differenti combinazioni casuali degli iperparametri. In figura 6.32 si riporta un estratto dello script che permette all'utente di impostare le variabili per l'addestramento della rete neurale.



Figura 6.32: Impostazione degli iperparametri per l'addestramento in Python

Una rete neurale adeguatamente addestrata può essere adoperata per stimare il danneggiamento della molla della sospensione attenendosi ai passaggi illustrati di seguito:

- Upload dello scaler impiegato per normalizzare i dati di input prima del training;
- Upload del modello della rete;
- Definizione di un vettore con tre componenti in cui inserire i nuovi input;
- Applicazione dello scaler al vettore;
- Utilizzo del metodo *predict* per stimare il valore della grandezza di output.

Il codice per l'utilizzo di una rete neurale è visibile in figura 6.33.

```
import numpy as np
from joblib import load
from keras.models import load_model
from pathlib import Path
scaler_path = Path('C:/.../scaler_NN_13_08_2023__10_43_17.save')
# Inserire il percorso della rete neurale da utilizzare
NN_name = 'C:/.../NN_13_08_2023__10_43_17.h5'
# Spostamento medio, spostamento alternato, numero di cicli
input_values = [2, 16, 22300]
NN = load model(NN name)
scaler = load(scaler_path)
# Da lista ad array
variabili = np.array(input_values).reshape(1, len(input_values))
variabili_scal = scaler.transform(variabili)
output_non_scalato = NN.predict(variabili_scal)
# Danneggiamento previsto dal modello
danno = 10**(-output_non_scalato)
```

Figura 6.33: Utilizzo di una rete neurale addestrata in Python

6.4.3 Risultati degli addestramenti

In questo paragrafo sono sintetizzati i risultati degli addestramenti di reti neurali aventi differenti combinazioni degli iperparametri. La ricerca degli opportuni valori da assegnare alle diverse variabili ha permesso di costruire un modello surrogato capace di prevedere il danneggiamento del componente in modo affidabile e preciso. Si puntualizza che in tutti i training è utilizzata la funzione "ReduceLROnPlateau"; viceversa, la funzione "EarlyStopping" non viene impiegata poiché non consente un'adeguata minimizzazione delle funzioni di costo. Analogamente, la tecnica di regolarizzazione del dropout non viene adottata in quanto non ha effetti benefici sui risultati degli addestramenti. Inoltre, si sottolinea che gli errori commessi da ogni rete sono calcolati confrontando i valori dei danni scalati. La relazione matematica che consente di conoscere il vero danneggiamento subito dal componente è:

$$D_{\nu} = 10^{-D_s}$$

Nell'equazione riportata D_{ν} indica il danno vero e D_s quello scalato. Purtroppo, questa conversione causa una riduzione dell'accuratezza delle reti nella previsione degli output. Per tal motivo, è stato necessario modificare gli iperparametri dei modelli fino ad ottenere dei risultati affetti da errori (relativo medio, assoluto medio e assoluto massimo) estremamente bassi. Inizialmente, il database è costituito da 16926 righe. Nella tabella 6.7 di pagina seguente sono visibili i parametri e i risultati relativi all'addestramento della prima rete neurale.

ADDESTRAMENTO RETE NEURALE 1					
PARAMETRI		STRUTTURA E RISULTATI			
PERCENTUALE TRAINING [%]	80	Numero di samples	16926		
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[300,150,70,30,15,5]		
ACTIVATION FUNCTION	Softplus	Errore relativo medio [%]	0,54017		
OUTPUT ACTIVATION FUNCTION	Softplus	Errore assoluto medio	0,01356		
KERNEL INITIALIZER	He-uniform	Errore assoluto massimo	0,16729		
BIAS INITIALIZER	He-uniform	Tempo Training [s]	2945		
DROPOUT	[0,0,0,0,0,0]				
LEARNING RATE INIZIALE	0,01				
OPTIMIZER	Adam				
LOSS FUNCTION	MAPE				
BATCH SIZE	16				
EPOCHS	2000				
RIDUTTORE	0,75				
MONITOR	Validation loss				
PATIENCE	40				
DELTA MINIMO	0,01				

Tabella 6.7: Parametri e risultati dell'addestramento della rete neurale 1

La funzione di attivazione "Softplus" non garantisce risultati soddisfacenti; di conseguenza, nel secondo training si sceglie di adottare la "Leaky-relu".

ADDESTRAMENTO RETE NEURALE 2				
PARAMETRI		STRUTTURA E RISULTATI		
PERCENTUALE TRAINING [%]	80	Numero di samples	16926	
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[300,150,70,30,15,5]	
ACTIVATION FUNCTION	Leaky-relu	Errore relativo medio [%]	0,43174	
OUTPUT ACTIVATION FUNCTION	Leaky-relu	Errore assoluto medio	0,01060	
KERNEL INITIALIZER	He-uniform	Errore assoluto massimo	0,10986	
BIAS INITIALIZER	He-uniform	Tempo Training [s]	3125	
DROPOUT	[0,0,0,0,0,0]			
LEARNING RATE INIZIALE	0,01			
OPTIMIZER	Adam			
LOSS FUNCTION	MAPE			
BATCH SIZE	16			
EPOCHS	2000			
RIDUTTORE	0,75			
MONITOR	Validation loss			
PATIENCE	40			
DELTA MINIMO	0,01			

Tabella 6.8: Parametri e risultati dell'addestramento della rete neurale 2

L'errore relativo medio è diminuito dello 0,11% ma quello assoluto medio è ancora elevato. Nel terzo addestramento si utilizza la distribuzione "He-normal" per inizializzare pesi e bias.

ADDESTRAMENTO RETE NEURALE 3					
PARAMETRI		STRUTTURA E RISULTATI			
PERCENTUALE TRAINING [%]	80	Numero di samples	16926		
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[300,150,70,30,15,5]		
ACTIVATION FUNCTION	Leaky-relu	Errore relativo medio [%]	0,41148		
OUTPUT ACTIVATION FUNCTION	Leaky-relu	Errore assoluto medio	0,00933		
KERNEL INITIALIZER	He-normal	Errore assoluto massimo	0,05715		
BIAS INITIALIZER	He-normal	Tempo Training [s]	3066		
DROPOUT	[0,0,0,0,0,0]				
LEARNING RATE INIZIALE	0,01				
OPTIMIZER	Adam				
LOSS FUNCTION	MAPE				
BATCH SIZE	16				
EPOCHS	2000				
RIDUTTORE	0,75				
MONITOR	Validation loss				
PATIENCE	40				
DELTA MINIMO	0,01				

Tabella 6.9: Parametri e risultati dell'addestramento della rete neurale 3

L'unico miglioramento significativo rispetto al precedente addestramento riguarda il valore dell'errore assoluto massimo. Pertanto, la quarta rete neurale è costruita aumentando il numero di neuroni.

ADDESTRAMENTO RETE NEURALE 4					
PARAMETRI		STRUTTURA E RISULTATI			
PERCENTUALE TRAINING [%]	80	Numero di samples	16926		
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[400,200,100,50,25,12]		
ACTIVATION FUNCTION	Leaky-relu	Errore relativo medio [%]	0,37861		
OUTPUT ACTIVATION FUNCTION	Leaky-relu	Errore assoluto medio	0,00876		
KERNEL INITIALIZER	He-normal	Errore assoluto massimo	0,10525		
BIAS INITIALIZER	He-normal	Tempo Training [s]	3478		
DROPOUT	[0,0,0,0,0,0]				
LEARNING RATE INIZIALE	0,01				
OPTIMIZER	Adam				
LOSS FUNCTION	MAPE				
BATCH SIZE	16				
EPOCHS	2000				
RIDUTTORE	0,75				
MONITOR	Validation loss				
PATIENCE	40				
DELTA MINIMO	0,01				

Tabella 6.10: Parametri e risultati dell'addestramento della rete neurale 4

L'accuratezza degli output previsti dalla rete è ancora troppo bassa. Dunque, si decide di incrementare la dimensione del database fino ad ottenere un numero di righe quasi doppio del precedente, pari a 32240.

ADDESTRAMENTO RETE NEURALE 5					
PARAMETRI		STRUTTURA E RISULTATI			
PERCENTUALE TRAINING [%]	80	Numero di samples	32240		
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[400,200,100,50,25,12]		
ACTIVATION FUNCTION	Leaky-relu	Errore relativo medio [%]	0,19312		
OUTPUT ACTIVATION FUNCTION	Leaky-relu	Errore assoluto medio	0,00524		
KERNEL INITIALIZER	He-normal	Errore assoluto massimo	0,07757		
BIAS INITIALIZER	He-normal	Tempo Training [s]	6312		
DROPOUT	[0,0,0,0,0,0]				
LEARNING RATE INIZIALE	0,01				
OPTIMIZER	Adam				
LOSS FUNCTION	MAPE				
BATCH SIZE	16				
EPOCHS	2000				
RIDUTTORE	0,75				
MONITOR	Validation loss				
PATIENCE	40				
DELTA MINIMO	0,01				

Tabella 6.11: Parametri e risultati dell'addestramento della rete neurale 5

Osservando la tabella 6.11, si nota che il tempo di training è aumentato di 2834 secondi ma l'errore relativo medio percentuale è praticamente dimezzato rispetto al quarto addestramento. A questo punto, si decide di accrescere ulteriormente il numero di neuroni in ogni hidden layer della rete.

ADDESTRAMENTO RETE NEURALE 6					
PARAMETRI		STRUTTURA E RISULTATI			
PERCENTUALE TRAINING [%]	80	Numero di samples	32240		
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[600,300,150,75,37,18]		
ACTIVATION FUNCTION	Leaky-relu	Errore relativo medio [%]	0,08162		
OUTPUT ACTIVATION FUNCTION	Leaky-relu	Errore assoluto medio	0,00236		
KERNEL INITIALIZER	He-normal	Errore assoluto massimo	0,04959		
BIAS INITIALIZER	He-normal	Tempo Training [s]	8427		
DROPOUT	[0,0,0,0,0,0]				
LEARNING RATE INIZIALE	0,01				
OPTIMIZER	Adam				
LOSS FUNCTION	MAPE				
BATCH SIZE	16				
EPOCHS	2000				
RIDUTTORE	0,75				
MONITOR	Validation loss				
PATIENCE	40				
DELTA MINIMO	0,01				

Tabella 6.12: Parametri e risultati dell'addestramento della rete neurale 6

Dopo il sesto addestramento l'errore relativo medio è estremamente contenuto (0,08%). L'errore assoluto medio è pari a 0,0024, mentre il tempo di training è incrementato di 2115 secondi.

Il prossimo addestramento è effettuato con un numero di epoche superiore al precedente e pari a 3000.

ADDESTRAMENTO RETE NEURALE 7					
PARAMETRI		STRUTTURA E RISULTATI			
PERCENTUALE TRAINING [%]	80	Numero di samples	32240		
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[600,300,150,75,37,18]		
ACTIVATION FUNCTION	Leaky-relu	Errore relativo medio [%]	0,07705		
OUTPUT ACTIVATION FUNCTION	Leaky-relu	Errore assoluto medio	0,00235		
KERNEL INITIALIZER	He-normal	Errore assoluto massimo	0,21409		
BIAS INITIALIZER	He-normal	Tempo Training [s]	12337		
DROPOUT	[0,0,0,0,0,0]				
LEARNING RATE INIZIALE	0,01				
OPTIMIZER	Adam				
LOSS FUNCTION	MAPE				
BATCH SIZE	16				
EPOCHS	3000				
RIDUTTORE	0,75				
MONITOR	Validation loss				
PATIENCE	40				
DELTA MINIMO	0,01				

 Tabella 6.13: Parametri e risultati dell'addestramento della rete neurale 7

Confrontando i risultati degli addestramenti 6 e 7 si riscontra una diminuzione dell'errore relativo medio a fronte di un aumento dell'errore assoluto massimo e del tempo di training. Pertanto, si modifica il valore del batch size, che passa da 16 a 64.

ADDESTRAMENTO RETE NEURALE 8					
PARAMETRI		STRUTTURA E RISULTATI			
PERCENTUALE TRAINING [%]	80	Numero di samples	32240		
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[600,300,150,75,37,18]		
ACTIVATION FUNCTION	Leaky-relu	Errore relativo medio [%]	0,06273		
OUTPUT ACTIVATION FUNCTION	Leaky-relu	Errore assoluto medio	0,00193		
KERNEL INITIALIZER	He-normal	Errore assoluto massimo	0,06724		
BIAS INITIALIZER	He-normal	Tempo Training [s]	4485		
DROPOUT	[0,0,0,0,0,0]				
LEARNING RATE INIZIALE	0,01				
OPTIMIZER	Adam				
LOSS FUNCTION	MAPE				
BATCH SIZE	64				
EPOCHS	3000				
RIDUTTORE	0,75				
MONITOR	Validation loss				
PATIENCE	40				
DELTA MINIMO	0.01				

Tabella 6.14: Parametri e risultati dell'addestramento della rete neurale 8

Il training dell'ottava rete neurale produce ottimi risultati: il processo di ottimizzazione ha condotto ad un'adeguata combinazione degli iperparametri che consente di minimizzare la differenza tra il danneggiamento atteso e quello previsto dal modello. Fino ad ora, tutti gli addestramenti sono effettuati con un numero di neuroni che decresce man mano che si procede dai primi agli ultimi hidden layers. Per ridurre ancora un po' gli errori, si eseguono ulteriori training in cui non si rispetta più la suddetta condizione. Dopo svariati tentativi, la rete neurale con gli errori più bassi tra gli output del test set e quelli valutati dal modello risulta essere quella descritta nella seguente tabella.

ADDESTRAMENTO RETE NEURALE 9				
PARAMETRI		STRUTTURA E RISULTATI		
PERCENTUALE TRAINING [%]	80	Numero di samples	32240	
VALIDATION SPLIT [%]	25	Dimensioni Hidden Layers	[150,300,600,75,37,18]	
ACTIVATION FUNCTION	Leaky-relu	Errore relativo medio [%]	0,05183	
OUTPUT ACTIVATION FUNCTION	Leaky-relu	Errore assoluto medio	0,00165	
KERNEL INITIALIZER	He-normal	Errore assoluto massimo	0,07600	
BIAS INITIALIZER	He-normal	Tempo Training [s]	4648	
DROPOUT	[0,0,0,0,0,0]			
LEARNING RATE INIZIALE	0,01			
OPTIMIZER	Adam			
LOSS FUNCTION	MAPE			
BATCH SIZE	64			
EPOCHS	3000			
RIDUTTORE	0,75			
MONITOR	Validation loss			
PATIENCE	40			
DELTA MINIMO	0,01			

Tabella 6.15: Parametri e risultati dell'addestramento della rete neurale 9

Nella tabella 6.16 si riportano la struttura e le variabili interne della rete neurale ottimizzata, che è selezionata per valutare il danneggiamento della molla a partire dalla storia di carico irregolare acquisita mediante il sensore laser.

	IL	HL 1	HL 2	HL 3	HL 4	HL 5	HL 6	OL	TOTALE
Nodi	3	150	300	600	75	37	18	1	/
Pesi	/	450	45000	180000	45000	2775	666	18	273909
Bias	/	150	300	600	75	37	18	1	1181
	IL = Input Layer ; HL = Hidden Layer ; OL = Output Layer								275090

Tabella 6.16: Parametri della rete neurale ottimizzata

Si noti che indicando con k il generico hidden layer, il numero di pesi corrispondente è pari al prodotto tra i nodi del layer k e del layer k-1. Invece, la quantità di bias per ogni layer è uguale a quella dei nodi. In definitiva, il training della rete neurale ottimizzata permette di ricavare gli opportuni valori da assegnare ai 275090 parametri del modello.

Nella tabella 6.17 si indica la suddivisione del database tra il training set, il validation set e il test set.

Numero righe database	32240
Dati addestramento	25792
TRAINING SET	19344
VALIDATION SET	6448
TEST SET	6448
Numero di input	3
Numero di output	1

Tabella 6.17: Training set, validation set e test set della rete neurale ottimizzata

Di seguito è visibile un confronto tra alcuni danneggiamenti previsti dalla rete e quelli attesi relativi al test set. La lunga fase di ricerca della giusta combinazione degli iperparametri ha condotto alla costruzione di un modello capace di estrarre ottimamente i pattern nascosti nei dati; infatti, si osserva una discrepanza minima tra gli output attesi e quelli predetti.

Danno atteso	Danno previsto
1,61E-02	1,61E-02
1,76E-06	1,74E-06
1,05E-02	1,05E-02
4,92E-03	4,94E-03
3,03E-02	3,03E-02
3,59E-04	3,58E-04
8,66E-03	8,64E-03
2,89E-02	2,87E-02
1,15E-04	1,15E-04
5,02E-04	5,01E-04
1,00E-03	1,00E-03
3,20E-06	3,21E-06
1,69E-02	1,69E-02
7,72E-07	7,70E-07
3,02E-05	3,03E-05
1,00E-02	1,00E-02
3,49E-02	3,49E-02
1,48E-02	1,47E-02
4,28E-03	4,28E-03
9,52E-04	9,53E-04

Tabella 6.18: Confronto tra il danno atteso e quello predetto dalla rete neurale ottimizzata

Infine, in figura 6.34 sono rappresentati gli andamenti delle loss functions. Fino all'epoca numero 500 la funzione di costo del validation set oscilla notevolmente, nonostante assuma dei valori inferiori al 2%. Successivamente, le due curve appaiono perfettamente sovrapposte, sintomo che il training ha permesso al modello di imparare correttamente le relazioni che legano gli input e gli output del database.



Figura 6.34: Andamenti delle loss functions relative alla rete neurale ottimizzata

Per concludere la trattazione riguardante le reti neurali, si evidenziano delle scelte compiute dall'utente che nella realizzazione del modello potrebbero avere effetti negativi sugli esiti dell'addestramento:

- Numero di neuroni per ogni layer troppo elevato o eccessivamente basso;
- Rete neurale esageratamente profonda;
- Numero di epoche sovrabbondante rispetto alla complessità dei pattern celati nel database;
- Utilizzo della funzione "EarlyStopping" di Python;
- Impiego della tecnica di regolarizzazione "dropout";
- Alti valori di learning rate;
- Piccoli valori di batch size;
- Funzione di attivazione inadatta alla tipologia di problema analizzato;
- Ridotte dimensioni del database;
- Inopportuna suddivisione del database in training set, validation set e test set.

7 Esempio di previsione del danneggiamento

In questo capitolo si riporta un esempio numerico di calcolo del danno subito dalla molla della sospensione anteriore, che viene sollecitata da spettri di carico estremamente irregolari. Nelle figure seguenti sono visibili gli andamenti dello spostamento verticale del componente in funzione del tempo, ricavati a partire dalle acquisizioni dei dati effettuate con il sensore laser ILD1220-200. Si precisa che la valutazione del danneggiamento della molla è effettuata considerando la storia di carico irregolare rilevata durante la prova in vettura numero 5.



Figura 7.1: Dati sperimentali relativi alla prova 3



Figura 7.2: Dati sperimentali relativi alla prova 4



Figura 7.3: Dati sperimentali relativi alla prova 5



Figura 7.4: Dati sperimentali relativi alla prova 10

Il procedimento che permette di valutare il danneggiamento della molla viene automatizzato mediante un codice scritto in Python, riassumibile nei seguenti passaggi:

- Importazione dei dati acquisiti dal sensore nell'ambiente di programmazione;
- Riorganizzazione della storia di carico in modo che inizi con il massimo spostamento (in valore assoluto). Tutti i punti antecedenti a quest'ultimo sono aggiunti alla fine della storia di carico originale;

- Cancellazione del rumore di misura attraverso la funzione "find_peaks", che elimina le oscillazioni del segnale poco significative ai fini dell'analisi;
- Trasformazione della storia di carico in una sequenza ordinata di picchi e valli;
- Applicazione del metodo di conteggio rainflow a 3 punti;
- Salvataggio dei parametri che identificano i cicli elementari estratti dalla storia di carico riorganizzata;
- Suddivisione degli spostamenti medi e alternati in intervalli;
- Assegnazione degli spostamenti medi e alternati di ogni ciclo elementare alla corrispondente banda di appartenenza;
- Valutazione del danneggiamento complessivo utilizzando la rete neurale ottimizzata e la legge di Palmgren-Miner.

Il metodo "find_peaks" filtra i dati di input in funzione del valore di soglia impostato nella variabile "prominence", che rappresenta la distanza verticale minima di un picco dalle valli adiacenti affinché venga considerato rilevante in termini di danno. Nel nostro caso, la prominence è pari a 2 mm. La storia di carico è trasformata in una successione di picchi e valli grazie alla porzione di codice indicata in figura 7.5, che esegue le seguenti operazioni:

- Selezione del primo punto della storia di carico;
- Se è un picco si ricerca la valle più bassa finché non si trova un nuovo picco;
- Se è una valle si ricerca il picco più alto finché non si trova una nuova valle;
- Salvataggio dell'istante temporale e dello spostamento relativi al picco o alla valle trovata;
- Iterazione del processo fino alla completa trasformazione della storia di carico.



Figura 7.5: Estratto di codice per la scomposizione della storia di carico in sequenze di picchi e valli

In figura 7.6 si grafica lo spostamento della molla acquisito durante un test in vettura; in figura 7.7 è visibile l'andamento della medesima grandezza a valle della trasformazione in una sequenza di picchi e valli.



Figura 7.6: Storia di carico prima della scomposizione in una sequenza di picchi e valli



Figura 7.7: Storia di carico dopo la scomposizione in una sequenza di picchi e valli

Con riferimento alla prova 5, gli spostamenti medi e alternati $(Z_m; Z_a)$ possono variare tra:

$$Z_{min} \le Z_m \le Z_{max}$$
$$0 \le Z_a \le \frac{Z_{max} - Z_{min}}{2}$$
$$Z_{min} = -42,939 mm$$
$$Z_{max} = 44,558 mm$$

Ovviamente, l'ampiezza degli intervalli $(A_{int, Z_m}; A_{int, Z_a})$ dipende dal numero di intervalli scelti $(N_{int, Z_m}; N_{int, Z_a})$ per suddividere gli spostamenti medi e alternati:

$$A_{int, Z_m} = \frac{Z_{max} - Z_{min}}{N_{int, Z_m}}$$
$$A_{int, Z_a} = \frac{Z_{max} - Z_{min}}{2 \cdot N_{int, Z_a}}$$

Applicando il metodo di conteggio alla storia di carico si estraggono le coppie $Z_m - Z_a$ dei cicli elementari, che vengono assegnate alle corrispondenti bande di appartenenza. Per stimare i danneggiamenti tramite la rete neurale si considerano i punti medi degli intervalli. Pertanto, l'accuratezza della soluzione finale è tanto maggiore quanto elevato è il numero di intervalli.



Figura 7.8: Estratto di codice per la previsione del danno complessivo accumulato nel componente

Per stabilire il numero minimo di intervalli che garantisce una previsione affidabile si esegue un'analisi di convergenza con la seguente modalità:

- Calcolo del danno complessivo "esatto" considerando 3000 intervalli sia per lo spostamento medio che per quello alternato;
- Valutazione dei danni complessivi al crescere del numero di intervalli relativo allo spostamento medio (lo spostamento alternato è sempre suddiviso in 24 intervalli);
- Valutazione dei danni complessivi al crescere del numero di intervalli relativo allo spostamento alternato (lo spostamento medio è sempre suddiviso in 24 intervalli);
- Calcolo dell'errore relativo tra il danno esatto e quelli ottenuti al variare del numero di intervalli.



Figura 7.9: Convergenza del danno in funzione del numero di intervalli dello spostamento alternato



Figura 7.10: Convergenza del danno in funzione del numero di intervalli dello spostamento medio

Suddividendo gli spostamenti alternati in 134 intervalli e gli spostamenti medi in 24, la soluzione converge a quella esatta.

Nella tabella 7.1 sono mostrati il danno subito dalla molla e una stima della distanza rimanente prima della rottura del componente. Lo spettro di carico utilizzato è relativo alla prova 5.

PROVA 5 - Valutazione danneggiamento componente		
Distanza percorsa in vettura [km]	0,75	
Numero ripetizioni	1000	
Distanza totale [km]	750	
Danno accumulato	2,009E-03	
Danno rimanente pre-rottura	9,980E-01	
Distanza rimanente pre-rottura [km]	372570	
Distanza totale pre-rottura [km]	373320	

Tabella 7.1: Previsione della distanza rimanente prima della rottura del componente

A valle dei risultati ottenuti, si riportano alcune considerazioni conclusive:

- La distanza effettivamente percorsa durante il test in vettura è di 0,75 km e si suppone di ripetere per 1000 volte il medesimo tragitto, coprendo una distanza totale di 750 km;
- Il danno accumulato in 750 km è pari a 0,002;
- Supponendo una relazione lineare tra il danno e la lunghezza del tragitto si ricava che il componente cederà entro 372570 km;
- Rispetto alle condizioni di carico della prova 5, la molla ha una vita complessiva di 373320 km, durata assolutamente in linea con i valori di riferimento reperibili in letteratura.

8 Conclusioni

Lo sviluppo del progetto di ricerca in Capgemini Italia S.p.A. ha permesso di ideare una nuova metodologia per controllare lo stato di salute di un componente automobilistico in tempo reale. Il progressivo incremento della connettività dei dispositivi elettronici installati nelle auto, le tecnologie IoV e IoT contribuiranno alla diffusione della manutenzione predittiva, che nasce dall'integrazione di sensori, modelli virtuali e algoritmi di intelligenza artificiale. La nostra analisi si concentra su un elemento specifico, ovvero la molla della sospensione anteriore della vettura di prova, ma l'approccio adottato nella pianificazione delle diverse fasi del progetto è del tutto generale. Infatti, è possibile applicare le tecniche di manutenzione predittiva ad altre parti dell'autoveicolo. Naturalmente, è fondamentale uno studio di fattibilità preliminare che consenta di identificare:

- Il sistema meccanico da esaminare, le principali cause di malfunzionamento degli elementi costituenti e i corrispondenti rischi per la sicurezza dei passeggeri;
- Le grandezze fisiche caratteristiche dei componenti, che rappresentano i parametri di input delle simulazioni virtuali;
- Il sistema di acquisizione dei segnali, compatibilmente con gli ingombri del sistema e con l'accuratezza richiesta dalla specifica applicazione ingegneristica;
- La tipologia di analisi da effettuare (strutturale, termica, fluidodinamica, elettromagnetica, chimica, acustica);
- I parametri di output delle simulazioni virtuali, in grado di quantificare il livello di danneggiamento dei componenti;
- La modalità di costruzione del database per l'addestramento delle reti neurali;
- Eventuali criticità nella pianificazione del progetto.

Tornando al nostro caso di studio, dai risultati riportati nel capitolo precedente si osserva che la durata della molla ammonta a circa trecentosettantamila chilometri. Si sottolinea che le prove sono effettuate con il solo conducente a bordo e con velocità piuttosto ridotte. Pertanto, è lecito aspettarsi che l'elemento elastico della sospensione abbia una vita notevolmente minore con condizioni di carico più gravose, in cui si considera il peso aggiuntivo dovuto alla presenza di bagagli e di quattro passeggeri a bordo. Nei prossimi mesi si eseguiranno ulteriori test in vettura per comparare i valori di danneggiamento della molla relativi a differenti livelli di sollecitazione, con l'obiettivo di validare definitivamente l'intera metodologia proposta. Infine, si elencano alcuni dei possibili sviluppi futuri del progetto:

- Applicazione delle tecniche di manutenzione predittiva ad altri componenti del veicolo, come freni, filtri, ammortizzatori, motore o trasmissione;
- Impiego delle tecniche di manutenzione predittiva in altri settori ingegneristici, come ad esempio quello railway o aerospace;
- Utilizzo combinato di molteplici sensori per misurare contemporaneamente diverse grandezze fisiche;
- Addestramento di numerose reti neurali per monitorare le condizioni operative di sistemi meccanici complessi.

Appendice A: Raspberry Pi

A.1 Che cos'è?



Figura A.1: Raspberry Pi 3 Model B+

Il Raspberry Pi è un computer a scheda singola sviluppato dalla Raspberry Pi Foundation, un'organizzazione britannica non a scopo di lucro nata nel 2009. Il dispositivo si basa su un sistema SoC (system on a chip), ossia un circuito integrato che combina in un solo chip la CPU (central processing unit), le memorie, la GPU (graphics processing unit), i controllers e le connessioni circuitali di input/output. La grande diffusione del Raspberry Pi è dovuta ad un ottimo rapporto qualità/prezzo del prodotto. Infatti, con un massimo di 70 \in è possibile acquistare un Single Board Computer che permette al consumatore di sfruttare le potenzialità di tecnologie Wi-Fi, Bluetooth e molto altro. Prima di acquistare il laser ILD1220-200, è stato eseguito un tentativo di acquisizione dei dati mediante un sensore sonoro collegato al Raspberry.

Di seguito si riportano alcuni esempi delle operazioni che questo dispositivo elettronico consente di effettuare:

- Installare un sistema operativo;
- Navigare su Internet;
- Testare circuiti elettrici;
- Prototipazione di progetti IoT;
- Scaricare giochi, musica, video;
- Collegare sensori o componenti elettronici;
- Programmare in Python;
- Creare un Web Server.

Inoltre, si elencano anche i modelli disponibili sul mercato:

• Raspberry Pi 1 Model A+;

- Raspberry Pi 1 Model A;
- Raspberry Pi 1 Model B;
- Raspberry Pi 1 Model B+;
- Raspberry Pi 2 Model B;
- Raspberry Pi 3 Model B;
- Raspberry Pi 3 Model B+;
- Raspberry Pi 3 Model A+;
- Raspberry Pi 4 Model B;
- Raspberry Pi Zero.

Si indicano le caratteristiche hardware del Raspberry Pi 4 Model B:

- CPU da 1,5 GHz;
- Memoria: microSD;
- RAM: 4 GB;
- Gigabit Ethernet;
- 2 porte USB 2.0;
- 2 porte USB 3.0;
- 2 porte micro HDMI con supporto 4K;
- Alimentazione 5 V con connettore USB-C;
- 40 GPIO;
- Moduli Wi-Fi e Bluetooth.

Il Raspberry Pi 4 Model B impiegato in azienda è equipaggiato con una scheda micro SD da 32 GB di classe 10, ovvero la velocità di scrittura permessa è di 10 MB al secondo. Nella SD card è installato il sistema operativo che contiene il software, i dati e le procedure di messa a punto indispensabili per il corretto funzionamento di ogni componente del dispositivo.

Per quanto riguarda i software, gli OS (operating system) diffusi dalla Raspberry Pi Foundation sono basati su Linux. Le distribuzioni più popolari sono NOOBS e Raspbian, ma sono disponibili anche sistemi operativi di terze parti come Ubuntu, Windows 10, OSMC o LIBREELEC.

Per realizzare il flashing della scheda SD si sfrutta il tool Etcher, che permette di selezionare un'immagine del sistema operativo da importare nella SD card. In seguito, si esegue una formattazione, si copia l'immagine selezionata e vengono eseguite delle verifiche riguardo il corretto completamento dell'operazione.



Figura A.2: Flashing della scheda SD

Il sistema operativo installato sul Raspberry Pi aziendale è Raspbian.



Figura A.3: Simboli dei sistemi operativi Raspbian e NOOBS

Per iniziare ad adoperare il Raspberry Pi sono sufficienti un mouse, una tastiera, un monitor, un cavo HDMI e un'alimentazione da 5 V DC. Inoltre, il dispositivo può essere configurato anche in headless mode, sfruttando lo schermo di un computer in possesso dell'utente. Per farlo, è sufficiente reperire l'indirizzo IP del Raspberry, connettersi mediante SSH (Secure Shell) ed attivare il VNC (Virtual Network Computing) Viewer per connettersi al proprio dispositivo.

A.2 General purpose input-output pins



Figura A.4: Raspberry Pi 4 GPIO Header

Una caratteristica che rende estremamente stimolante l'impiego del Raspberry è la presenza dei general-purpose input/output (GPIO) pins, che permettono di interagire in tempo reale con il mondo fisico. Infatti, i pin possono essere usati come input per acquisire segnali provenienti da sensori, tasti o altri componenti. Allo stesso modo, è consentito adoperarli come output per accendere un LED, inviare dei comandi ad un motore elettrico o in generale per dialogare con dispositivi esterni. Inoltre, la capacità del Raspberry di connettersi alla rete implica la possibilità di gestione online dei componenti ad esso collegati. Per questo motivo, il suo utilizzo si sposa perfettamente con la prototipazione di progetti IoT.

In figura A.4 viene riportato uno schema dei 40 pin disponibili sul Raspberry Pi 4. Si noti che ad ogni pin, ad eccezione di quelli destinati all'alimentazione o alla messa a terra, sono assegnate due diverse tipologie di numerazione. La prima rispetta l'ordine fisico con cui si susseguono i pin sulla scheda e viene definita "BOARD mode". La seconda è invece detta "BCM mode" (ossia Broadcom mode) e rappresenta la numerazione non sequenziale usata dal chip interno al Raspberry per accedere ai pin. In generale, i pin operano con una tensione di 3,3 V e per alcuni di essi sono disponibili delle funzioni specifiche da attivare durante la fase di programmazione in Python. Si riportano alcuni esempi esplicativi:

- Pin 3 o GPIO 2 con linea seriale di comunicazione per i dati (SDA, ossia Serial DAta);
- Pin 5 o GPIO 3 con linea seriale di comunicazione per il clock (SCL, ossia Serial CLock);
- Pin 32 o GPIO 12 con PWM0, ossia possibilità di effettuare una modulazione di larghezza d'impulso.

Si sottolinea che nel Raspberry Pi sono assenti pin analogici, pertanto per processare letture analogiche è necessario installare un convertitore analogico-digitale esterno.



Figura A.5: Voltaggio e numerazione dei pin

Per creare un circuito elettrico è possibile interfacciare il Raspberry Pi con una breadboard, ossia una basetta in materiale plastico con dei fori equispaziati dove è possibile inserire i componenti desiderati (ad esempio LED, resistenze, sensori, potenziometri, convertitori). Quest'ultimi sono collegati meccanicamente ed elettricamente attraverso agganci metallici presenti al di sotto della struttura (non visibili dall'esterno). In figura A.6 sono visibili le linee di alimentazione (in rosso ed azzurro): in tale zona della breadboard la direzione di equipotenzialità dei fori è orizzontale, mentre nell'area centrale della basetta è verticale.



Figura A.6: Breadboard per Raspberry e Arduino

Una volta acceso e configurato, il Raspberry Pi garantisce la possibilità di scrivere dei codici sorgente mediante Thonny, un development environment di Python 3. In questo modo, l'utente è in grado di testare e controllare, attraverso la programmazione, i circuiti elettrici montati sulla breadboard.

Inizialmente, si è tentato di utilizzare il trasmettitore sonico HC-SR04 per misurare gli spostamenti della molla della sospensione. La risoluzione del sensore, pari a 0,3 cm, non consente di acquisire dati con la precisione richiesta, ma il suo impiego dimostra che il Raspberry è un valido strumento per elaborare dei dati in tempo reale. I vantaggi di adoperare dei sensori compatibili con il Raspberry risiedono nella ridotta complessità di cablaggio del circuito e in spese di acquisto dei componenti contenute.

Successivamente, la compressione-estensione della molla si valuta mediante un sensore laser a triangolazione con accuratezza dell'ordine delle decine di micron.

Il sistema DAQ dedicato consente la connessione tra il trasmettitore di precisione e il computer mediante porta USB: in questo modo la gestione dei dati in tempo reale viene ottimizzata con l'ausilio di un device unico e compatto, sia durante la fase di acquisizione che in quella di post-processing.

A.3 Sensore HC-SR04

Il sensore ad ultrasuoni HC-SR04 consente di misurare la distanza che lo separa dall'oggetto bersagliato mediante l'invio e la ricezione di onde sonore che si propagano nell'aria dell'ambiente. Il suo impiego è giustificato nelle applicazioni dove è necessario un campo di misura piuttosto ampio ed una precisione dell'ordine dei millimetri.

In figura A.7 si nota che il sensore è costituito da due cavità cilindriche (trasmettitori ad ultrasuoni), 4 pin di collegamento (VCC, TRIG, ECHO, GND), un circuito elettronico integrato per la gestione e il controllo dei segnali.



Figura A.7: Sensore ad ultrasuoni HC-SR04

Si elencano le caratteristiche tecniche del dispositivo:

- Tensione di alimentazione: 5 V DC;
- Corrente assorbita a riposo: < 2 mA;
- Corrente assorbita durante il funzionamento: < 20 mA;
- Segnale di output: digitale (0 V oppure 5 V);
- Frequenza di lavoro: 40 kHz;
- Campo di misura: 2 ÷ 400 cm;
- Risoluzione: 0,3 cm;
- Angolo di utilizzo: < 15°;
- Durata impulso al pin Trigger: 10 µs;
- Dimensioni: 45 x 20 x 15 mm;
- Peso: 10 g.
A.3.1 Principio di funzionamento

Il sensore è comandato mediante i pin presenti alla sua base. Le connessioni a VCC e GND sono indispensabili per l'alimentazione e la messa a terra del dispositivo, mentre il controllo di TRIG ed ECHO conduce alla misura della distanza voluta. La prima operazione da compiere riguarda il trigger pin, che deve essere settato allo stato logico "HIGH" per 10 microsecondi.

A questo punto, l'HC-SR04 emette onde sonore ad alta frequenza in successione mediante uno dei due trasmettitori; nel frattempo il pin ECHO commuta il suo stato in "HIGH".

Quando il segnale colpisce il bersaglio desiderato e viene riflesso al trasmettitore ricevente del sensore, l'output del pin ECHO torna allo stato "LOW". La valutazione dell'intervallo temporale Δt consente di risalire alla misura della distanza attraverso la velocità di propagazione delle onde sonore nell'aria. In figura A.8 è visibile uno schema esemplificativo del funzionamento del sensore.



Figura A.8: Andamenti dei segnali elettrici dei pin TRIG e ECHO

A.3.2 Cablaggio

Nella figura seguente viene mostrato il circuito realizzato per interfacciare il sensore ultrasonico con il Raspberry Pi 4.



Figura A.9: Circuito elettrico per la connessione del sensore HC-SR04 al Raspberry Pi 4

Materiale occorrente:

- Raspberry Pi 4;
- Breadboard;
- Sensore HC-SR04;
- Una resistenza da 1 kΩ;
- Una resistenza da 2 kΩ;
- 3 Male-Male jumper wires;
- 4 Male-Female jumper wires.

Con riferimento alla figura A.9, si riassumono le connessioni circuitali:

- Primo cavo rosso: da VCC ad un foro della breadboard con potenziale negativo;
- Secondo cavo rosso: da un foro adiacente al precedente al PIN 4 del Raspberry;
- Primo cavo nero: da GND ad un foro della breadboard con potenziale positivo;
- Secondo cavo nero: da un foro adiacente al precedente al PIN 6 del Raspberry;
- Cavo blu: da TRIG al PIN 40 del Raspberry;
- Cavo verde: da un foro della breadboard adiacente alle resistenze al PIN 38 del Raspberry;
- Cavo marrone: da GND ad un foro adiacente al terminale della resistenza da 2 k Ω .

L'introduzione delle due resistenze in serie consente di realizzare un partitore di tensione: in tal modo si preleva il segnale dal pin ECHO con un valore di tensione compatibile con quella dei pin GPIO (3V).

A.3.3 Codice Python per l'utilizzo

In questa sezione si allega il codice Python per l'utilizzo del sensore ad ultrasuoni HC-SR04.

```
import RPi.GPIO as GPIO
     import time
     GPIO.setwarnings(False)
     istante_iniziale = 0
     istante_finale = 0
     velocità suono = 34300
     soglia_sensibilità = 4
11
     tempo stabilizzazione = 0.2
     tempo_trigger = 0.00001
12
     tempo_attesa = 1
14
15
     filename='distanza_sonar.txt'
17
     GPIO.setmode(GPIO.BCM)
     PIN_TRIGGER = 21
20
     PIN_ECHO = 20
22
     dataframe=[]
     dati_acquisizione=()
     contatore=0
25
     while True:
         print("Misura della distanza con sensore HC-SR04")
         GPIO.setup(PIN_TRIGGER, GPIO.OUT)
         GPIO.setup(PIN_ECHO, GPIO.IN)
         GPIO.output(PIN TRIGGER, GPIO.LOW)
32
         print("Attendere che il segnale fornito dal sensore si stabilizzi")
         time.sleep(tempo_stabilizzazione)
         GPIO.output(PIN_TRIGGER, GPIO.HIGH)
         time.sleep(tempo trigger)
         GPIO.output(PIN_TRIGGER, GPIO.LOW)
         while GPIO.input(PIN_ECHO)==0:
              istante iniziale = time.time()
         while GPIO.input(PIN_ECHO)==1:
42
              istante_finale = time.time()
```

Figura A.10: Prima parte del codice per l'utilizzo del sensore HC-SR04

```
intervallo_temporale = istante_finale - istante_iniziale
         tempo = time.ctime()
         distanza = round(intervallo temporale * velocità suono/2,2)
         if distanza > soglia_sensibilità:
             dati_acquisizione = tempo, distanza
51
             dataframe.append(dati_acquisizione)
52
             file = open(filename, 'a')
             file.write(str(dataframe[contatore]))
             file.write('\n')
             contatore+=1
             file.close()
             print(f"Distanza in cm: {distanza:.2f}")
         else:
             print("La misura non è affidabile, attendere nuovi valori ")
62
         time.sleep(tempo_attesa)
64
     GPIO.cleanup()
```

Figura A.11: Seconda parte del codice per l'utilizzo del sensore HC-SR04

Di seguito si riassume il significato delle righe di comando utilizzate nello script, al fine di facilitare la comprensione del funzionamento del sensore.

- Righe 1-2: importazione delle librerie "GPIO" e "time" per comunicare con i pin del Raspberry ed accedere alle funzioni per la manipolazione del tempo, che nel codice è espresso in secondi;
- Righe 6-13: inizializzazione delle variabili; in particolare, i parametri temporali sono impostati per inviare al sensore un segnale di avviamento e attendere una quantità di secondi opportuna a stabilizzare i valori dei dati acquisiti durante misurazioni consecutive;
- Riga 17: selezione della tipologia di numerazione dei pin, con modalità "BCM";
- Righe 19-20: assegnazione ai pin Trig ed Echo del sensore la corrispondente numerazione dei pin del Raspberry a cui sono collegati;
- Riga 26: ciclo "while" per il salvataggio dei dati, viene eseguito finché l'utente non interrompe l'esecuzione del programma;
- Righe 28-29: impostazione dei pin Trig e Echo come output e input pin, rispettivamente;
- Righe 34-37: il Raspberry commuta lo stato logico del pin Trig del sensore per 10 μs, dando inizio alla fase di acquisizione;

- Righe 39-43: salvataggio degli istanti temporali in cui le onde sonore sono inviate da un trasmettitore e ricevute dall'altro, dopo essere state riflesse dal componente bersagliato. Lo stato logico del pin Echo rappresenta la variabile da monitorare;
- Righe 45-47: calcolo della distanza che separa il sensore e l'oggetto. La velocità del suono è espressa in centimetri al secondo;
- Righe 48-61: le variabili tempo-distanza sono aggiunte al dataframe e salvate in colonne distinte all'interno di un file di testo. Qualora la misura non superi la soglia di sensibilità (pari a 4 cm) in cui è garantita la precisione dello strumento, l'utente visualizza un messaggio di errore;
- Riga 63: la nuova iterazione all'interno del ciclo "while" viene eseguita dopo aver atteso un tempo pari ad 1 secondo. Questa accortezza evita eccessive fluttuazioni del segnale durante acquisizioni successive.

Bibliografia

- [1] David Govi, "Deep learning con LSTM per la manutenzione predittiva", URL: https://www.industry4business.it/servitization/predictivemaintenance/deep-learning-con-lstm-per-la-manutenzione-predittiva/ (2020)
- [2] Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, "Mathematics for Machine Learning", Cambridge University Press (2020)
- [3] Sebastian Ruder, "An overview of gradient descent optimization algorithms", URL: https://www.ruder.io/optimizing-gradient-descent/ (2016)
- [4] Shree Nayar, "Backpropagation Algorithm | Neural Networks", URL: https://www.youtube.com/watch?v=sIX 9n-1UbM (2021)
- [5] Adattato dal libro "Better Deep Learning" di Jason Brownlee, "Diagnosing Model Performance with Learning Curves", URL: https://rstudio-conf-2020.github.io/dl-keras-tf/notebooks/learning-curve-diagnostics.nb.html (2018)
- [6] Keras, URL: https://keras.io [consultato a Maggio 2023]
- [7] Stuart J. Russell, Peter Norvig, "Artificial intelligence: a modern approach" (2021)
- [8] Shubham Jain, "An Overview of Regularization Techniques in Deep Learning", URL: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learningregularization-techniques/ (2023)
- [9] Preprocessing data, URL: https://scikitlearn.org/stable/modules/preprocessing.html [consultato a Maggio 2023]
- [10] IBM Data e AI Team, "AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference?", URL: https://www.ibm.com/blog/ai-vsmachine-learning-vs-deep-learning-vs-neural-networks/ (2023)
- [11] Brembo S.p.A., URL: https://www.brembo.com/it [consultato a Marzo 2023]
- [12] Sidem, URL: https://www.sidem.be/it [consultato a Marzo 2023]
- [13] Reinhoch, "Tipici malfunzionamenti delle sospensioni", URL: https://reinhoch.eu/it/approfondimenti/tipici-malfunzionamenti-dellesospensioni/ [consultato a Marzo 2023]
- [14] Suspension Secrets, "How to Calculate Installation Ratio", URL: https://suspensionsecrets.co.uk/installation-ratio/ [consultato a Maggio 2023]
- [15] Micro-Epsilon, "Sensore di spostamento laser compatto per OEM e utilizzo in serie", URL: https://www.micro-epsilon.it/displacement-position-sensors/laser sensor/optoNCDT_1220/ [consultato ad Aprile 2023]
- [16] Micro-Epsilon, "Modulo di interfaccia IF2001/USB per la conversione del segnale da RS422 a USB", URL: https://www.microepsilon.it/accessories/IF2001USB/ [consultato ad Aprile 2023]

- [17] Luchsinger Srl, URL: https://www.luchsinger.it/it/ [consultato ad Aprile 2023]
- [18] Instrumentation devices Srl, URL: https://www.instrumentation.it/it/home/?gclid=EAIaIQobChMI2-bdnMfgQMVWQWLCh1pbQWNEAAYASAAEgJgkfD BwE [consultato ad Aprile 2023]
- [19] Politecnico di Bari, "Caratterizzazione dei Sensori", URL: https://svelto.faculty.polimi.it/didattica/materiale_didattico/materiale_didattico_M A/Sensori%20Caratterizzazione.pdf [consultato ad Aprile 2023]
- [20] A.C. Bertolino, A. De Martin, A. Nesci, M. Sorli, "Meccatronica Analisi, progettazione e modellazione di servosistemi" [consultato ad Aprile 2023]
- [21] G.Petrucci, "Lezioni di Costruzione di Macchine, Fatica ad ampiezza variabile", URL:https://sites.unipa.it//giovanni.petrucci/Disp/Fatica3AmpVar.pdf (2022)
- [22] N. Bonora, "Fatica (HCF): conteggio dei cicli", URL: http://www.cdmunicas.it/notes/CDM2016/Lecture5_Fatica_HCF_Cycle_counting. pdf (2016)
- [23] Massimo Rossetto, "Fatica multiassiale", URL: https://imechanica.org/files/Slide%208%20-%20Fatica%20multiassiale.PDF [consultato a Maggio 2023]
- [24] FEMFAT software, URL: https://femfat.magna.com [consultato a Giugno 2023]
- [25] Joseph E. Shigley, Richard G. Budynas, J. Keith Nisbett, "Progetto e costruzione di macchine", McGraw-Hill (2020)
- [26] De Paulis Antonio, "Costruzione di macchine: criteri di base e applicazioni principali" (2019)
- [27] Gustav Niemann, Hans Winter, Bernd-Robert Höhn, "Manuale degli organi delle macchine: metodologie di progettazione, giunzioni, collegamenti a vite, molle, cuscinetti, lubrificazione e usura, alberi, calettamenti e tenute" (2006)
- [28] Guido Bongiovanni, Giovanni Roccati, "Le molle: tipi e criteri di calcolo" (1994)
- [29] HyperMesh Tutorials, URL: https://2021.help.altair.com/2021.2/hwsolvers/os/topics/chapter_heads/hypermesh tutorials r.htm [consultato a Giugno 2023]
- [30] RaspberryPi, URL: https://www.raspberrypi.com [consultato ad Aprile 2023]

Ringraziamenti

Questa sezione è dedicata a tutte le persone che mi hanno sostenuto durante il lungo percorso di studi in Ingegneria Meccanica. Ringrazio il mio relatore, il Professor Massimo Sorli, sempre disponibile ad elargire le giuste indicazioni in ogni fase della realizzazione del presente elaborato. Un ringraziamento particolare è destinato al mio tutor aziendale, l'ing. Giovanni Pesare, per il supporto costante e i preziosi consigli che hanno permesso di approfondire al meglio le tematiche legate al progetto di ricerca. Ringrazio tutto lo staff dell'azienda Capgemini Italia S.p.A., in cui ho svolto un tirocinio formativo della durata di 7 mesi finalizzato alla stesura della tesi, per l'ospitalità e per le competenze acquisite sul campo. Nello specifico, ringrazio Edoardo Compagnone ed Enzo Cardano per il loro impagabile ed incessante sostegno. Ringrazio tutti i miei amici e i colleghi dell'università per i momenti di spensieratezza trascorsi insieme. Ringrazio i miei coinquilini con cui ho condiviso un periodo stupendo della mia vita professionale e privata. Ringrazio i miei genitori, Enio e Daniela, per l'indiscussa fiducia che da sempre ripongono in me, per la vicinanza e l'immenso affetto che mi trasmettono, nonostante la notevole distanza che spesso ci separa. Siete le persone più incredibili che il destino potesse regalarmi, la mia maggiore fonte d'ispirazione. Inoltre, ringrazio tutti i componenti della mia splendida e numerosa famiglia per i continui incoraggiamenti. Infine, un pensiero speciale è dedicato alla mia ragazza, Isabella. Sei stata al mio fianco in ogni momento di difficoltà, ascoltandomi, comprendendomi e soprattutto spronandomi. Abbiamo affrontato fragilità e insicurezze nel migliore dei modi: insieme. Grazie per aver creduto in me anche nelle circostanze in cui era faticoso starmi accanto. Non dimenticherò mai tutto quello che abbiamo condiviso.