



POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Sviluppo di un algoritmo per la Manutenzione Predittiva nell'ambito dei sistemi generali del velivolo

Relatori:

Prof. Massimo Sorli

Prof. Andrea De Martin

Tutor aziendali:

Ing. Sandro Milano

Ing. Francesco Allegra

Ing. Matteo Moscufo

Candidato:

Pietro Gontero

ANNO ACCADEMICO 2024/2025

ABSTRACT

Reliability and predictive maintenance play a crucial role in the aerospace sector, where safety and operational efficiency depend on the ability to predict and mitigate the degradation of critical components. In this context, prognostics emerges as an innovative approach for estimating the *Remaining Useful Life (RUL)* of a component and for planning targeted maintenance interventions, reducing operational costs and improving system availability.

This thesis first introduces the concept of prognostics, analyzing its main methodologies and potential future applications in the field. The integration of prognostic algorithms, employing artificial intelligence, into modern aircraft represents a crucial step toward developing more efficient maintenance strategies, optimizing and improving the management of component replacement interventions.

The primary objective of this study is to develop an algorithm for estimating the *RUL* of an electric motor within an electromechanical actuator (EMA) used in the flight control systems of a regional aircraft. The reference model is a simplified version of the one described in "*Simulation of an All-Electric Flight Control System for the Evaluation of Power Consumption*" [\[1\]](#), which analyzes the power requirements of electromechanical actuators used to control aircraft surfaces. The focus is specifically on the three-phase brushless synchronous motor within the EMA, investigating the degradation of resistance and inductance, key parameters in monitoring operational conditions.

Due to the lack of real data and established empirical models for the degradation of these parameters, synthetic data was generated based on reasonable assumptions taken from existing literature. The developed Matlab code utilizes specific toolboxes to extract meaningful features from the data and formulate an *RUL* prediction for the analyzed component.

The primary goal of this study is not to achieve definitive and applicable results but rather to establish a structured methodological process that can serve as a foundation for future advancements in predictive maintenance.

Indice:

Indice delle figure:.....	4
1 INTRODUZIONE ALLA PROGNOSTICA.....	7
1.1 Definizione ed obiettivi della Prognostica.....	7
1.2 Tecniche di Prognostica	9
1.3 Applicazioni della prognostica.....	10
1.4 Applicazione della Prognostica nel settore aeronautico.....	11
1.4.1 Introduzione ai servomeccanismi nei comandi di volo	11
1.4.2 Comandi di volo primari	11
1.4.3 Comandi di volo secondari.....	13
1.4.4 Prognostica nei comandi di volo primari	14
1.4.5 Prognostica nei comandi di volo secondari.....	14
1.4.6 Tecniche di supporto per prognostica nei comandi di volo	15
2 STATO DELL'ARTE DELLA PROGNOSTICA.....	17
2.1 Modelli fisici (Physics-Based Models).....	17
2.1.1 Sfide dei modelli fisici.....	19
2.2 Modelli basati sui dati (Data-Driven Models)	20
2.2.1 Metodi di apprendimento e Neural Networks.....	21
2.2.2 K-means	25
2.2.3 Fuzzy Logic	25
2.2.4 Alberi decisionali	26
2.2.5 Support Vector Machine (SVM).....	26
2.2.6 Algoritmi di rilevamento delle anomalie	27
2.2.7 Tecniche numeriche convenzionali.....	28
2.3 Modelli ibridi	29
2.4 Sfide nella Prognostica	31
3 MODELLO ATTUATORE ELETTROMECCANICO	33
3.1 Gli attuatori elettromeccanici in generale	33
3.2 Modello EMA proposto	36
3.3 Modello EMA utilizzato.....	37
3.3.1 Modello matematico	38
3.3.2 Modello Matlab-Simulink.....	40
4 SVILUPPO DELL'ALGORITMO PER LA MANUTENZIONE PREDITTIVA	53
4.1 Introduzione alla Manutenzione Predittiva	53

4.2	Remaining Useful Life (<i>RUL</i>)	59
4.3	Sviluppo dell'algoritmo	64
4.3.1	Definizione del parametro di degrado	64
4.3.2	Generazione dei dati	68
4.3.3	Diagnostic Feature Designer	72
4.3.4	Health Indicator Designer	79
4.3.5	Creazione del modello <i>RUL</i>	82
4.3.6	Stima della vita utile rimanente (<i>RUL</i>)	84
5	CONCLUSIONI	89
	Bibliografia	93

Indice delle figure:

Figura 1 : Progressione temporale di un guasto.....	8
Figura 2 : Comandi di volo primari.....	12
Figura 3 : Comandi di volo primari e secondari	13
Figura 4 : Processo Physics-based model.....	18
Figura 5 : Processo Data-driven Model	21
Figura 6 : Processo semplificato di funzionamento di un ANNs feedforward	22
Figura 7 : Processo dei movimenti dei dati all'interno di una rete neurale.....	23
Figura 8 : Flowchart dei modelli prognostici ibridi	30
Figura 9 : Modello ibrido con implementazione di modello fisico e modello basato sui dati	31
Figura 10 : Classificazione delle incertezze presenti nella prognostica	32
Figura 11 : (a) Motore spazzolato DC, (b) Motore brushless	33
Figura 12 : Vite a ricircolo di sfere.....	34
Figura 13 : Layout del FCS	36
Figura 14 : Modello Matlab-Simulink proposto dallo studio	37
Figura 15 : Segnale di comando di ingresso	41
Figura 16 : Seconda tipologia di comando di ingresso	41
Figura 17 : Modello Matlab-Simulink	42
Figura 18 : Schema del Controller	44
Figura 19 : Schema del sottosistema di riferimento EMA	45
Figura 20 : Schema del Plant.....	45
Figura 21 : Blocco di valutazione della velocità angolare ω_m	46
Figura 22 : Schema valutazione ω_m	46
Figura 23 : Blocco di valutazione della corrente diretta i_d	47
Figura 24 : Schema di valutazione della corrente diretta i_d	47
Figura 25 : Blocco di valutazione della corrente di quadratura i_q	48
Figura 26 : Schema di valutazione della corrente di quadratura i_q	48
Figura 27 : Blocco di valutazione della posizione attuale dell'attuatore x_{pos}	49
Figura 28 : Schema di valutazione della posizione attuale dell'attuatore x_{pos}	49
Figura 29 : Blocco di valutazione delle potenze del sistema.....	50
Figura 30 : Schema di valutazione delle Potenze.....	51
Figura 31 : Funzionamento Manutenzione Reattiva	53
Figura 32 : Funzionamento Manutenzione Preventiva.....	54
Figura 33 : Funzionamento Manutenzione Predittiva	54
Figura 34 : Flusso di lavoro con cui sviluppare l'algoritmo per la Manutenzione Predittiva.....	55
Figura 35 : Esempio di preprocessing dei dati	56
Figura 36 : Tipologie di Condition Indicators	57
Figura 37 : Rilevamento delle anomalie	57
Figura 38 : Identificazione del tipo di guasto	57
Figura 39 : Stima della vita utile residua (RUL)	58
Figura 40 : Distribuzione dell'algoritmo su dispositivo edge e cloud	58
Figura 41 : Stima della RUL	59
Figura 42 : Similarity Model.....	60
Figura 43 : Degradation Model	60
Figura 44 : Survival Model	60
Figura 45 : Tipologie di Similarity Model.....	62
Figura 46 : Tipologie di Degradation Model	63

Figura 47 : Tipologie di Survival Model	63
Figura 48 : Grafico del parametro di degrado D in funzione di diverse temperature degli avvolgimenti del motore elettrico	66
Figura 49 : Individuazione dei dieci valori del parametro di degrado D	67
Figura 50 : Variazione della corrente diretta	
Figura 51: Variazione della corrente di quadratura	69
Figura 52 : Variazione della potenza apparente	
Figura 53 : Variazione della potenza totale persa	69
Figura 54 : Variazione della potenza meccanica	
Figura 55 : Variazione della potenza attiva	69
Figura 56 : Variazione della potenza elettrica persa	
Figura 57 : Variazione della potenza meccanica persa	70
Figura 58 : Variazione della coppia	
Figura 59 : : Variazione della velocità angolare.....	70
Figura 60 : Variazione dell'accelerazione angolare	
Figura 61 : Variazione della posizione attuale	70
Figura 62 : Variazione della velocità lineare dell'attuatore	71
Figura 63 : Formato della Tabella da fornire in input al Diagnostic Feature Designer	71
Figura 64 : Preprocessing dei segnali forniti	72
Figura 65 : Scelta della tipologia di feature	73
Figura 66 : Selezione delle feature da estrarre.....	76
Figura 67 : Istogrammi delle features per il segnale della corrente diretta <i>id</i>	77
Figura 68 : Classificazione delle feature con il metodo "Prognostic Ranking"	78
Figura 69 : Health Indicator Designer	80
Figura 70 : Variazione degli indicatori di salute nel tempo t.....	82
Figura 71 : Stima della vita utile rimanente del componente per ogni settimana.....	86
Figura 72 : Variazione degli indicatori di salute HI per ogni iterazione	87
Figura 73 : Stima della RUL per dieci dataset diversi di parametri R ed L	88

1 INTRODUZIONE ALLA PROGNOSTICA

La prognostica è una tecnica di manutenzione che negli ultimi anni ha guadagnato sempre più rilevanza in diversi settori, tra cui quello aeronautico, automobilistico ed industriale, per la previsione dello stato di salute e del tempo di vita residuo di un sistema o di un componente.

Questa nuova disciplina è chiamata “*Prognostic and Health Management*” (PHM) e si pone l’obiettivo principale di stimare in anticipo il momento in cui un componente raggiungerà un punto di guasto, in modo da pianificare interventi di manutenzione più efficienti e mirati.

Questo approccio segna un passaggio importante dalla manutenzione preventiva a quella predittiva e permette di gestire in maniera più funzionale le risorse e la sicurezza operativa. Nel campo aeronautico, ad esempio, la prognostica potrebbe essere applicata per aumentare la sicurezza dei voli e per ridurre i tempi di fermo dei velivoli, consentendo di pianificare interventi manutentivi basati sulle reali condizioni dei componenti e non affidandosi a controlli programmati e sostituzioni periodiche.

1.1 Definizione ed obiettivi della Prognostica

La prognostica può essere definita come: “*Una disciplina ingegneristica focalizzata sulla stima del tempo al quale un sistema o un componente non svolgerà più la sua funzione prevista*” [2]. Essa può, quindi, prevedere il comportamento futuro di un determinato elemento in base a dati di monitoraggio raccolti in precedenza e quindi già archiviati, oppure a dati raccolti sul momento dal componente stesso.

In particolare, i due compiti principali della prognostica sono monitorare lo stato di salute, rilevando eventuali segnali di degrado o guasto, e predire la Vita Utile Residua, nota con il termine inglese “*Remaining Useful Life*” (RUL). Questo secondo compito è di fondamentale importanza, in quanto la RUL rappresenta il tempo stimato prima che un sistema necessiti manutenzione o che sia a rischio guasto.

Tale stima assume un’importanza ancora maggiore se viene applicata a componenti critici per la sicurezza di un velivolo come, ad esempio, i comandi di volo. Attualmente la manutenzione che viene eseguita su tali sistemi segue l’approccio bastato sulla “*safe life*”, secondo cui un componente viene ispezionato o sostituito dopo un periodo programmato, indipendentemente dal suo stato di salute attuale. I vantaggi legati a questa tipologia di manutenzione sono relativi all’ampio margine di sicurezza, in quanto questo non verrà utilizzato in una condizione di vita utile residua vicina alla massima, creando però uno svantaggio a livello economico, poiché anche i componenti ancora ben funzionanti vengono sostituiti, facendo lievitare i costi.

L'utilizzo di un approccio basato sul *PHM* ha, perciò, una serie di numerosi vantaggi rispetto all'approccio *safe life*, tra cui:

- L'ottimizzazione e la riduzione dei costi di manutenzione, poiché consente di evitare o ridurre gli interventi manutentivi non necessari, ispezionando più di un componente alla volta;
- l'aumento della sicurezza e dell'affidabilità dei componenti e dei sistemi, evitando la comparsa di guasti improvvisi e mantenendo costantemente sotto controllo lo stato di salute di questi;
- l'ottimizzazione dell'uso dei componenti, aumentando il ciclo di vita dei componenti e riducendo i ricambi.

Per comprendere al meglio il ruolo che ha la prognostica all'interno del campo manutentivo, bisogna innanzitutto definire la relazione che c'è tra la capacità di diagnosi e di prognosi. Come riportato in "*Intelligent Fault Diagnosis And Prognosis For Engineering Systems*" [3], immaginare una sequenza temporale di un guasto iniziale che conduce a rottura è un buon modo per visualizzare la relazione tra le due modalità. In Figura 1 viene riportata tale sequenza con un nuovo componente funzionante, il quale ad un certo istante raggiunge un punto in cui si sviluppa un guasto incipiente precoce. Continuando ad utilizzare il componente esso raggiungerà la condizione di rottura del sistema e se non si interviene ulteriormente, esso raggiungerà una condizione di rottura secondaria, che potrebbe portare ad una rottura catastrofica.

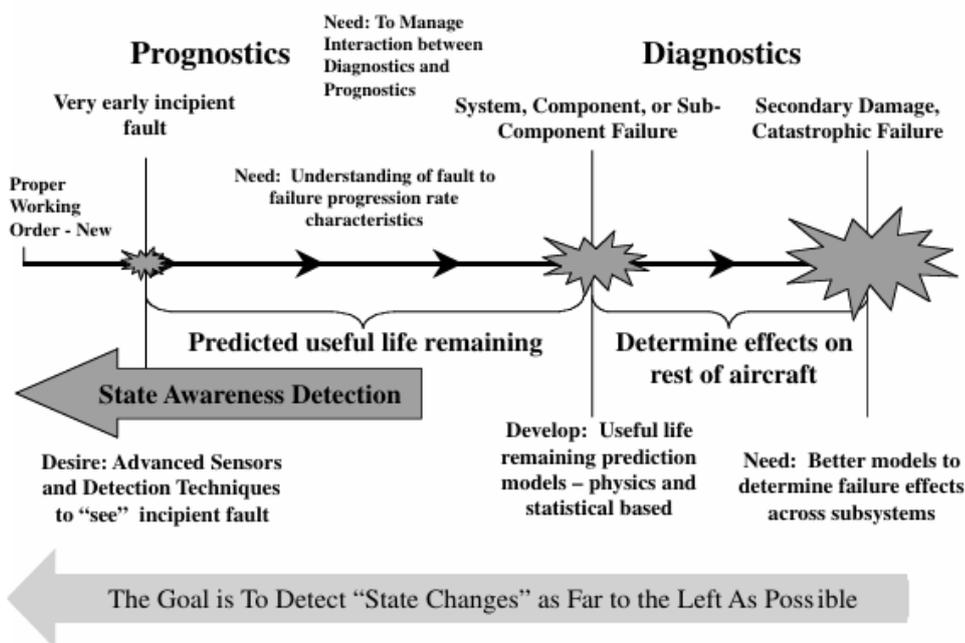


Figura 1 : Progressione temporale di un guasto [3]

La diagnostica è sempre stata utilizzata tradizionalmente tra l'istante in cui viene rilevato un danno al componente e l'istante in cui il guasto diventa catastrofico. Le nuove tecnologie diagnostiche prevedono di effettuare dei rilevamenti del danno incipiente in fasi molto precedenti, in modo da massimizzare i vantaggi del continuo utilizzo operativo di un sistema. La manutenzione viene quindi ritardata fino a quando il guasto precoce non raggiunge uno stato di salute più grave rispetto al caso precedente, ma prima di un'effettiva rottura del componente. Quest'area di studio tra il rilevamento del guasto incipiente precoce e la progressione verso lo stato di "failure" del sistema fa proprio riferimento alla prognostica e alla stima della Vita Utile Residua, la quale è in grado di predire accuratamente la vita rimanente per un particolare sistema o componente.

1.2 Tecniche di Prognostica

La prognostica può essere effettuata con diverse tecniche, ma le principali possono essere classificate in tre grandi categorie: tecniche basate sui dati, tecniche basate sui modelli fisici e tecniche basate sui modelli ibridi. Ognuna di queste ha i propri vantaggi e le proprie limitazioni specifiche e viene selezionata facendo uno studio preliminare sulle caratteristiche del sistema in esame e sui dati disponibili.

Nel successivo capitolo verrà svolta un'approfondita analisi sullo stato dell'arte di ognuna di queste tecniche, mentre qui ne vengono riportate solo le caratteristiche principali e gli strumenti maggiormente utilizzati in questi modelli.

Modelli basati sui dati (Data-driven Models):

I Data-driven models sono dei modelli che sfruttano una vasta quantità di dati storici raccolti durante il funzionamento del componente specifico per poter tracciare un pattern ed individuare delle correlazioni che indichino lo sviluppo di un degrado all'interno del sistema. In particolare, i pattern generati vengono sfruttati per prevedere quando un guasto potrà verificarsi e ciò viene fatto grazie all'utilizzo di tecniche di machine learning ed intelligenza artificiale, le quali sono alla base di tali modelli. Tra le principali in utilizzo si trovano:

- *Reti neurali artificiali (ANN)*: sono reti neurali che apprendono pattern complessi sfruttando i dati a loro conoscenza.
- *Support Vector Machines (SVM)* ed alberi decisionali: sono strumenti utili a classificare il livello di degrado del sistema e a stimare il tempo residuo del componente.

Modelli fisici (Physics-based Models):

Le tecniche di prognostica basate sui modelli fisici prevedono una dettagliata comprensione dei materiali che compongono il componente e dei processi che possono causare un degrado in questo. Tramite questi modelli è possibile simulare i fenomeni che portano al degrado e al guasto del componente o del sistema, ma per far sì che funzionino al meglio è necessaria una conoscenza approfondita delle proprietà di questi. I modelli fisici più utilizzati sono.

- *Modelli a fatica dei materiali:* sono dei modelli basati sulla conoscenza della resistenza dei materiali sottoposti a diversi cicli di stress.
- *Modelli termodinamici:* sono dei modelli che vengono utilizzati per stimare il degrado causato da surriscaldamenti o sbalzi di temperatura all'interno del componente.
- *Analisi del deterioramento meccanico:* viene effettuata un'analisi approfondita del comportamento meccanico di sistemi che operano in condizioni molto complesse, come possono essere i servomeccanismi.

Modelli ibridi:

I modelli ibridi sono genericamente i migliori e, dove si può, sono quelli più utilizzati nella prognostica. Essi combinano i vantaggi dei modelli basati sui dati con quelli dei modelli fisici, fondendo informazioni empiriche e simulazioni fisiche per generare delle previsioni più accurate, sfruttando la disponibilità dei dati storici del componente e le conoscenze teoriche su di esso. Un modello ibrido utilizza, dunque, un modello fisico per rappresentare i parametri principali al quale integra un algoritmo di machine learning per migliorare la stima del degrado del sistema.

1.3 Applicazioni della prognostica

La prognostica sta trovando ed ha trovato applicazione in diversi settori del campo industriale, dove la previsione dei guasti può incrementare la sicurezza, ottimizzare i costi e prolungare la vita dei componenti.

Vengono riportati a titolo di esempio alcuni ambiti in cui la *PHM* ha riscontrato maggiore utilizzo. Tra questi è presente il settore automobilistico, il quale ha adottato tecniche di prognostica per monitorare lo stato di salute di componenti come il motore, le batterie e sistemi frenanti. Particolare attenzione viene posta sui nuovi modelli di veicoli elettrici e nello specifico sulle batterie agli ioni di litio, la cui capacità residua viene monitorata e stimata grazie all'utilizzo di sensori per la temperatura, la tensione e la corrente. Un altro importante settore in cui la prognostica ha trovato ampio utilizzo è quello manifatturiero. In questo ambito la *PHM* aiuta a monitorare l'usura dei macchinari, controllando le

vibrazioni e la temperatura ad esempio dei robot impiegati per operazioni di assemblaggio, in modo da ridurre i tempi di inattività.

Il settore di maggiore interesse è però quello aeronautico, in cui la prognostica assume un ruolo decisivo per il monitoraggio di componenti critici come i motori, i sistemi di controllo di volo e i servomeccanismi. Particolare attenzione viene posta dunque all'applicazione in questo settore, poiché è rappresentativo di ciò di cui tratta questa tesi.

1.4 Applicazione della Prognostica nel settore aeronautico

Nell'ambito aeronautico, la prognostica assume un ruolo rilevante per la sicurezza dei voli, l'ottimizzazione dei tempi di manutenzione e la riduzione dei costi operativi. Un componente particolarmente critico che predilige un approccio manutentivo di questo tipo è il sistema di controllo di volo, il quale è composto dai servomeccanismi dei comandi di volo primari e secondari. Tali sistemi assumono un ruolo di fondamentale importanza per il controllo aerodinamico e la stabilità di un velivolo ed un loro guasto può comportare conseguenze catastrofiche per la sicurezza del volo.

1.4.1 Introduzione ai servomeccanismi nei comandi di volo

I servomeccanismi nei comandi di volo hanno avuto una grossa evoluzione negli ultimi anni grazie ai grandi miglioramenti che ci sono stati nelle tecnologie del settore aeronautico. Inizialmente i comandi di volo erano governati solamente dalla forza pura imposta dal pilota, la quale, negli anni a venire, fu sostituita grazie all'introduzione degli attuatori idraulici e pneumatici. Questi sistemi assunsero sempre più importanza con il passare degli anni e una particolare attenzione venne posta sullo studio di questi componenti. L'aumento della loro complessità portò però ad una riduzione dell'affidabilità e dell'efficienza dell'intero velivolo: una piccola perdita nel sistema pneumatico o idraulico può infatti causare un non corretto funzionamento dell'intero sistema, provocando guasti o conseguenze catastrofiche. Il passo successivo fu quello di avvicinarsi ad una filosofia che sta prendendo sempre più piede nel mondo moderno, ossia quella del "*more electric*". I servomeccanismi introdotti seguendo questa filosofia furono gli elettromeccanici (*EMA*) che saranno oggetto di studio nei capitoli successivi.

Lo scopo dei comandi di volo è quello di modificare delle superfici aerodinamiche, in modo da cambiare le caratteristiche di volo del velivolo stesso. Essi vengono suddivisi in due categorie: i comandi di volo primari e i comandi di volo secondari.

1.4.2 Comandi di volo primari

I comandi di volo primari vengono utilizzati per generare delle coppie di squilibrio attorno ai tre assi corpo del velivolo, variando quindi il momento di rollio, di beccheggio e di imbardata.

Questi tre momenti generano delle variazioni nella stabilità del velivolo e sono governati da delle specifiche superfici di controllo situate in zone apposite:

- *Ailerons*: sono situati sul bordo di fuga di ciascuna delle due ali; il loro ruolo è quello di agire sul momento di rollio generando uno squilibrio laterale grazie ad un'attuazione opposta tra le due ali.
- *Elevators*: sono situati sul piano di coda orizzontale, dietro allo stabilizzatore; il loro compito è quello di far variare il momento di beccheggio, generando un aumento o una diminuzione di angolo d'attacco del velivolo.
- *Rudder*: situato sull'impennaggio di coda verticale; fa variare il momento di imbardata causando un cambio di direzione dell'aeromobile.

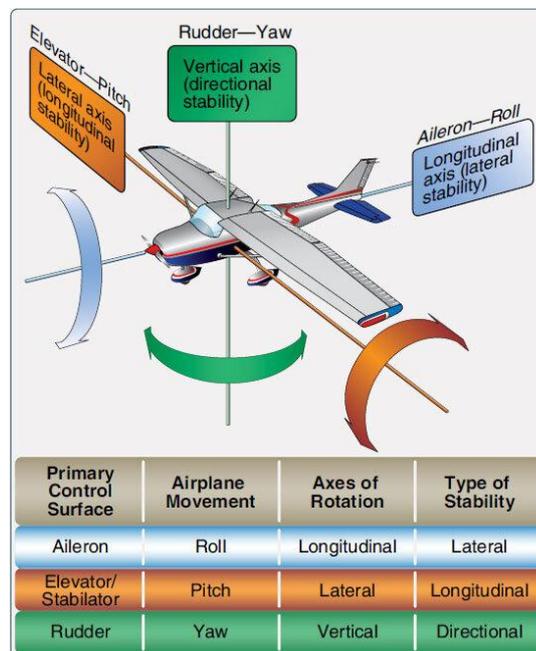


Figura 2 : Comandi di volo primari [5]

I comandi di volo primari non svolgono da soli tutto il lavoro di controllo della stabilità del velivolo, ma sono necessarie numerose altre superfici ed alette di controllo che permettono al pilota di manovrare nel miglior modo possibile nelle condizioni più critiche, come possono essere il decollo, l'atterraggio e in situazioni di manovre a basse velocità. Tali superfici di controllo aggiuntive sono i comandi di volo secondari.

1.4.3 Comandi di volo secondari

I comandi di volo secondari hanno il compito di modificare geometricamente e aerodinamicamente le superfici portanti, in particolare le ali, per adattare l'aeromobile a tutte le condizioni di volo. Tra questi i più importanti sono:

- *Flaps*: sono installati sul bordo di fuga delle ali come gli *Ailerons*; il loro compito è quello di aumentare il coefficiente di portanza C_L , aumentando la superficie dell'ala e abbassando la velocità di stallo; per questo motivo vengono generalmente utilizzati durante le fasi di atterraggio e decollo, quando le velocità del velivolo sono inferiori.
- *Slats*: sono situati sul bordo d'attacco dell'ala dell'aeromobile; il loro obiettivo è quello di reindirizzare il flusso d'aria in arrivo sull'ala in modo da aumentarne l'aderenza su di essa e ritardare la separazione del flusso per alti angoli di attacco.
- *Spoilers*: si trovano sul dorso dell'ala e la loro funzione è quella di diminuire la portanza ed aumentare la resistenza; per questo motivo vengono utilizzati in crociera per controllare la velocità del velivolo ed in fase di atterraggio per arrestare la sua corsa.

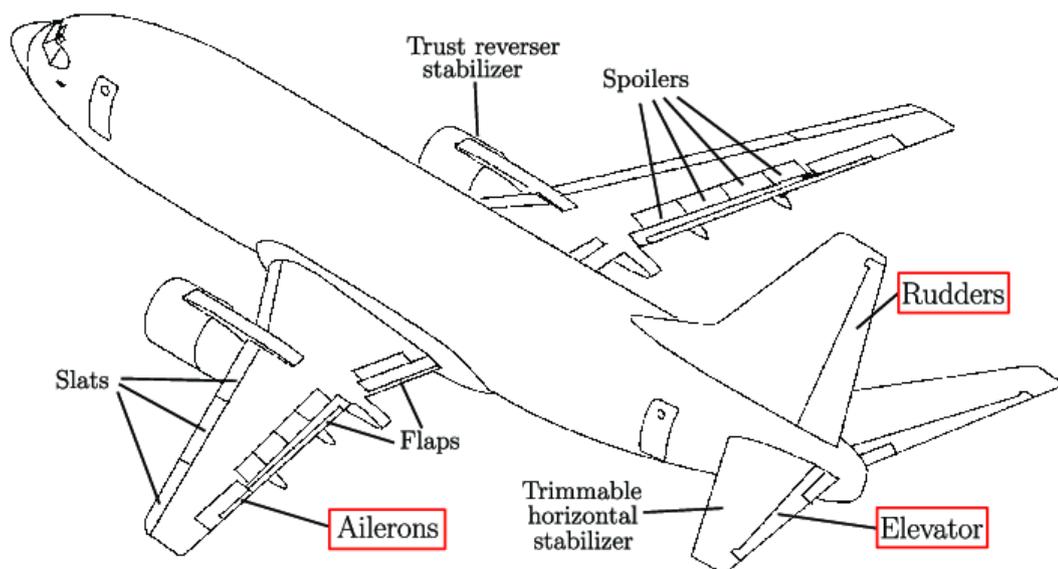


Figura 3 : Comandi di volo primari e secondari [6]

Da tutto ciò si evince che i servomeccanismi svolgono un ruolo cruciale per la sicurezza del volo e devono funzionare con alta precisione in condizioni ambientali spesso estreme. Grazie alla prognostica è possibile quindi monitorare il loro stato di salute e prevederne il degrado, in modo da garantire una manutenzione mirata e diminuire il rischio di guasti improvvisi.

Si riportano ora a titolo esemplificativo alcune applicazioni della prognostica su questa tipologia di sistemi.

1.4.4 Prognostica nei comandi di volo primari

La prognostica nei servomeccanismi dei comandi di volo primari sta trovando sempre più spazio in diversi aspetti di utilizzo di questi. Avendo un ruolo cruciale ed essendo composti da elementi complessi sono diverse le condizioni che vanno monitorate per assicurarsi un corretto funzionamento del sistema.

Tra questi uno dei principali è il monitoraggio delle vibrazioni. Le vibrazioni anomale possono, infatti, essere un indicatore di problemi legati all'usura o al disallineamento dei componenti meccanici del servomeccanismo, come i cuscinetti o gli ingranaggi. Un buon utilizzo della prognostica prevede il rilevamento delle vibrazioni nelle frequenze operative tramite l'utilizzo di sensori e accelerometri e l'analisi di tali dati campionati tramite tecniche che prevedono l'uso della trasformata di *Wavelet* o della trasformata di *Fourier* per identificare i segnali di degrado precoce.

Una seconda tipologia di applicazione della prognostica può essere legata all'analisi dell'assorbimento di corrente. Una variazione nell'assorbimento della corrente elettrica può infatti essere legata a dei problemi di attrito interno o di sovraccarico nei servomeccanismi. Un aumento di questa è indicatore di un intensificarsi del lavoro svolto dal motore, per superare, ad esempio, degli attriti dovuti ad una lubrificazione insufficiente o a componenti usurati. Tramite l'utilizzo di modelli predittivi è possibile stimare la vita utile residua (*RUL*), basandosi sulla tendenza di aumento della corrente.

Anche la temperatura operativa del sistema va monitorata per individuare eventuali surriscaldamenti anomali, causati da attriti eccessivi o sovraccarichi elettrici. Entrambi sono, infatti, segnali di un possibile guasto e grazie alla prognostica è possibile prevedere quando tale surriscaldamento raggiungerà livelli critici, in modo da intervenire in tempo.

1.4.5 Prognostica nei comandi di volo secondari

I comandi di volo secondari, sebbene siano componenti meno critici nel controllo della dinamica del velivolo, devono comunque svolgere dei ruoli essenziali per la sicurezza complessiva del volo. Per questo motivo una corretta manutenzione supportata dalla prognostica deve essere effettuata anche su questa tipologia di sistemi.

I servomeccanismi dei *flap* e degli *spoiler* eseguono dei movimenti di attuazione complessi e devono rispondere rapidamente ai comandi forniti dal pilota. Il monitoraggio effettuato in tempo reale della velocità e della precisione della risposta data dal sistema consente di individuare eventuali rallentamenti o disallineamenti sintomo di un possibile degrado delle componenti meccaniche o idrauliche. Gli attuatori utilizzati per questa tipologia di comando sono, infatti, lineari e quindi soggetti a movimenti ripetitivi e usuranti. Grazie alla prognostica è possibile identificare una tendenza al degrado basata sulla frequenza e

sull'intensità dei movimenti, sfruttando algoritmi di machine learning che possono stimare in tempo reale la *RUL* del componente.

Gli attuatori di tipo elettrico dei comandi di volo secondari devono, invece, operare a determinati livelli di temperatura e di potenza. Tramite un'analisi predittiva dell'assorbimento di potenza e della variazione di calore è possibile rilevare sovraccarichi o problemi elettrici, consentendo di intervenire prima che il degrado del componente diventi critico.

1.4.6 Tecniche di supporto per prognostica nei comandi di volo

L'applicazione della prognostica per la manutenzione dei servomeccanismi dei comandi di volo richiede una combinazione di tecnologie avanzate di raccolta dati, monitoraggio ed analisi. Per questo motivo le applicazioni reali e già in atto non sono molte, ma tra queste è possibile trovare delle tecnologie di supporto che si basano su sensori IoT (*Internet of Things*), i quali raccolgono dati sui parametri operativi dei servomeccanismi, come le vibrazioni, la temperatura, la corrente e la posizione, inviandoli ad un sistema centralizzato per effettuare le dovute analisi. Strumenti come l'Intelligenza Artificiale (IA) e il machine learning vengono impiegati per identificare dei pattern di degrado ed anomalie tra questi dati raccolti. Ad esempio, le reti neurali possono essere addestrate per riconoscere il degrado meccanico in base a delle frequenze di vibrazioni che le vengono fornite, ma la gestione di grandi volumi di dati crea dei problemi legati alle grosse infrastrutture che vengono utilizzate per tale scopo. Il *cloud computing* permette di archiviare e processare i dati in modo efficiente, supportando così le analisi in tempo reale e la condivisione delle informazioni tra più gruppi.

I modelli di fatica dei materiali e le simulazioni fisiche vengono, invece, utilizzate particolarmente per i comandi di volo primari, con i quali si può predire l'usura dei materiali sottoposti a stress ciclici.

Gli esempi riportati riguardanti il settore aeronautico dimostrano come l'applicazione della prognostica nei servomeccanismi può migliorare la sicurezza del volo, ridurre i costi di manutenzione e ottimizzare la gestione delle risorse. La combinazione di tecniche di monitoraggio, modelli di machine learning e simulazioni fisiche permette di ottenere delle stime accurate della vita utile residua dei componenti e di identificare i segnali di degrado precoce. Tutto ciò però deve fare fronte a delle grandi sfide che gli si pongono di fronte e che negli anni futuri andranno affrontate: l'alta complessità dei modelli di degrado; la raccolta e gestione dei dati che richiede sensori robusti e infrastrutture adeguate; la formazione del personale, in quanto l'implementazione di tali tecniche manutentive richiede competenze specifiche e complesse.

Grazie all'integrazione di questi strumenti nei sistemi di manutenzione predittiva, in un futuro le compagnie aeree potranno affrontare al meglio le sfide attuali e garantire prestazioni affidabili per i loro aeromobili.

2 STATO DELL'ARTE DELLA PROGNOSTICA

In questo capitolo verranno studiati nel dettaglio i diversi approcci che vengono sfruttati per la prognostica ed analizzati attentamente gli algoritmi che li guidano e le sfide associate alla loro applicazione pratica.

Attualmente esistono tre tipologie diverse di analisi dei dati nel settore della prognostica: modelli fisici, modelli basati sui dati e modelli basati sulla conoscenza [7]. I modelli fisici richiedono una accurata e precisa rappresentazione del comportamento del sistema fisico, sia nel suo stato di salute sano, sia in quello difettoso. Per individuare un'anomalia nello stato di salute del sistema è possibile comparare i dati ottenuti dai sensori con dei modelli predittivi. I modelli basati sui dati utilizzano, invece, una serie di dati storici registrati su componenti precedenti per valutare l'attuale stato di salute del sistema e prevederne la *RUL*, basandosi su dati *run-to-failure*. Questa metodologia viene frequentemente impiegata ai fini di stima, riferendosi ad un valore di soglia per il guasto prestabilito, ma richiede una comprensione generale di tutte le caratteristiche fisiche del sistema.

2.1 Modelli fisici (Physics-Based Models)

Il presupposto fondamentale per l'utilizzo dei modelli fisici è la presenza di un quadro fondamentale che chiarisca il processo di usura del componente. Per questo motivo, spesso il modello fisico viene indicato come modello di degradazione, mentre la prognostica basata sulla fisica è anche chiamata prognostica basata sul modello. Se esiste un modello fisico che rappresenta fedelmente il deterioramento di un componente nel tempo, allora la prognostica può essere applicata in maniera efficace.

Il modello di degradazione viene formulato come una funzione matematica, che dipende dalle circostanze di utilizzo o di carico, indicate dal parametro L , dal tempo di utilizzo t e dai parametri del modello, indicati da θ . Le condizioni di utilizzo sono la prima causa di incertezza, poiché sono caratterizzate da usi futuri sconosciuti, ma solitamente queste insieme al tempo di utilizzo vengono predeterminate in fase di sviluppo del modello. Stabilito ciò, l'obiettivo principale dei modelli fisici è quello di identificare i parametri del modello e di predire di pattern futuri di degradazione. In Figura 4 viene rappresentato uno schema del processo a cui fa riferimento la prognostica basata sui modelli fisici.

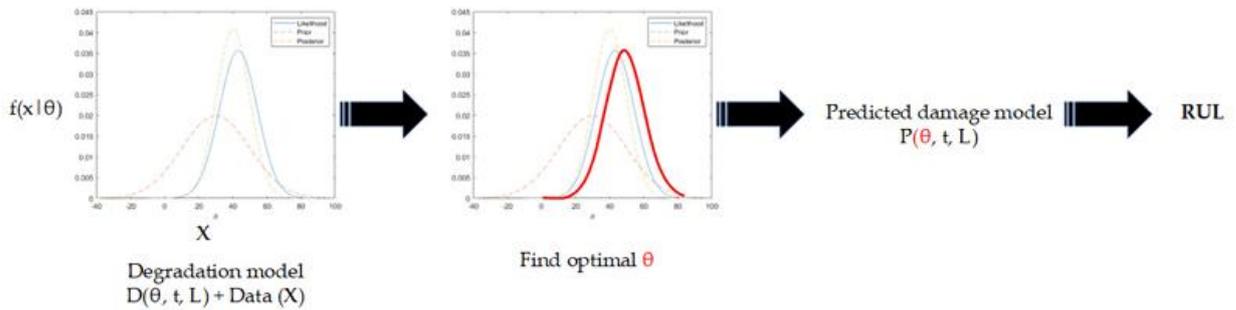


Figura 4 : Processo Physics-based model [7]

Per ottenere una stima della *RUL* la più accurata possibile è essenziale ridurre il grado di incertezza legato ai parametri del modello, per migliorare la precisione della previsione e, di conseguenza, la durata della manutenzione. Una volta che tali parametri sono stati identificati, è dunque possibile determinare il comportamento futuro del degrado, estendendo l'utilizzo del modello a periodi successivi. Per far ciò, è necessario sostituire i parametri identificati, il tempo e i carichi all'interno del modello di degradazione, propagando in modo continuo la condizione di degrado finché non raggiunge un valore di soglia prestabilito, ottenendo la previsione della *RUL*.

Le tecniche per l'individuazione dei parametri sono diverse e fungono da strumento per categorizzare i modelli fisici in differenti aree. Tra queste troviamo i minimi quadrati non lineari (*NLS*), il metodo *Bayesian (BM)* ed approcci basati sui filtri multipli come il filtro *Kalman (KF)* e il filtro particellare (*PF*). Questi ultimi due, in particolare, adottano un approccio basato sul filtraggio che fa variare iterativamente i parametri includendo i dati di misurazioni individuali. L'efficacia dell'algoritmo *KF* è fortemente influenzata dalle condizioni iniziali e dalla varianza dei parametri in gioco, oltre che dall'accuratezza dell'approssimazione della linearizzazione. L'algoritmo *PF* al contrario non è soggetto ad alcuna limitazione per quanto riguarda il tipo di sistema e di rumore.

La scelta della tecnica per l'individuazione dei parametri viene basata su diversi fattori, che possono essere la tipologia di dati utilizzati, l'incertezza, il rumore presente e la dimensione dei dati, ma è importante sottolineare che nessun algoritmo è in grado di soddisfare i requisiti di ogni sistema. L'influenza di questi parametri sul comportamento del modello è, infatti, molto importante e spesso non viene compresa appieno, per questo la loro individuazione viene considerata una parte integrante del processo prognostico. Sono presenti diversi lavori in letteratura in cui questo argomento viene studiato per trovare delle soluzioni innovative ai problemi che questa metodologia di prognostica porta con sé, come in "Yu, Y.; Yao, H.; Liu, Y. *Aircraft Dynamics Simulation Using a Novel Physics-Based Learning Method. Aersp. Sci. Technol.* 2019, 87, 254–264." [8]. Lo studio introduce un nuovo approccio basato sui principi del modello fisico, definito come "*Model Order Reduction method*" (*MOR*), per simulare la dinamica di un velivolo. Il metodo prevede l'integrazione dei principi fondamentali dei sistemi dinamici di un aeromobile all'interno di modelli di machine learning, con l'obiettivo di minimizzare le spese di apprendimento ed aumentare la

capacità di simulazione. I risultati dimostrano come l'approccio di apprendimento basato sui modelli fisici ha una maggiore efficienza computazionale rispetto ai metodi numerici tradizionali. Questo perché questa tipologia di algoritmi sfruttano degli step temporali maggiori, che violerebbero il vincolo della stabilità numerica, mantenendo al contempo uno schema di integrazione temporale esplicito.

2.1.1 Sfide dei modelli fisici

A differenza della seconda tipologia di modello prognostico, i modelli basati sui dati possiedono svariati vantaggi, in quanto hanno la capacità di generare delle stime a lungo termine. Infatti, una volta che i parametri del modello sono stati accuratamente stimati, la previsione della *RUL* viene effettuata propagando semplicemente il modello di degradazione finché non raggiunge il valore soglia prestabilito. Un ulteriore vantaggio riguarda il numero di dati necessari per fare questa stima, i quali, teoricamente, dovrebbero essere equivalenti al solo numero dei parametri indefiniti del modello, ma che nella realtà sono di un numero superiore a causa della presenza di rumore all'interno dei dati e dell'aumento della degradazione dei parametri stessi.

Tuttavia, sono presenti anche alcuni problemi che ostacolano l'utilizzo pratico di questa metodologia, tra i quali troviamo l'accuratezza del modello, la stima dei parametri e l'affidabilità dei dati.

L'accuratezza del modello è un fattore fondamentale, in quanto l'adeguatezza di questo rappresenta la capacità di previsione del comportamento futuro del degrado. Inoltre, una particolare attenzione viene posta sull'analisi di diversi *data point* e, in particolare, sugli errori associati all'estrapolazione. Prima di effettuare qualsiasi analisi, è necessario effettuare come primo passo una validazione del modello tramite metodi statistici, poiché la maggior parte dei modelli sono basati su assunzioni ed approssimazioni. Il problema dell'adeguatezza di un modello molto complesso può essere parzialmente risolto sfruttando dei parametri simili presi da un modello più semplice. In letteratura sono presenti alcuni esempi di questa applicazione e vengono riportati i vantaggi che ne porta come in "Coppe, A.; Pais, M.J.; Haftka, R.T.; Kim, N.H. Equivalent Damage Growth Parameters Using a Simplified Model. In *Proceedings of the ASME Design Engineering Technical Conference, Washington, DC, USA, 28–31 August 2011; Volume 2, pp. 1317–1328.*" [9], dove la stima della formazione di una frattura all'interno di una geometria complessa viene effettuata basandosi sull'utilizzo un modello di *Paris* semplificato, nel quale è stata impiegata una componente di stress assunta ed in cui i parametri sono stati modificati a causa dell'imprecisione dovuta al fattore di stress.

La stima dei parametri assume un ruolo importantissimo nel processo prognostico, poiché consente la previsione della *RUL*. Anche questo passo presenta però diverse difficoltà, poiché l'accuratezza della stima dei parametri è influenzata dalla tipologia di approccio che viene seguito per effettuarla, oltre al fatto che la presenza di correlazioni tra i diversi parametri del modello porta ulteriori difficoltà per la stima di questi. Nonostante queste difficoltà possano causare delle problematiche nell'identificazione dei componenti, è

comunque possibile ottenere una precisa previsione del grado di deterioramento e della *RUL*.

Infine, un'ulteriore sfida di cui bisogna tener conto nel processo prognostico dei modelli fisici è l'affidabilità dei dati. Quando questi vengono registrati dai sensori posti nei punti di interesse sono presenti notevoli livelli di rumore e *bias* a causa di diversi fattori. Il rumore è il termine che descrive le fluttuazioni casuali presenti all'interno dei segnali, dovute ad interferenze o campi elettromagnetici indesiderati presenti nei componenti elettronici. La presenza del rumore genera dei problemi nella distinzione dei segnali legati al degrado. Il *bias* è causato dalla continua deviazione del segnale dal suo valore reale, a causa di problemi legati spesso alla calibrazione e la sua presenza porta a degli errori nella previsione dei risultati. La presenza di questi due fattori all'interno dei dati di registrazione rappresenta un grosso ostacolo nell'identificazione delle caratteristiche di degrado e per le capacità prognostiche dei modelli fisici. La mitigazione di questi fattori, in particolare del rumore, viene spesso effettuata con delle tecniche di "*de-noising*".

2.2 Modelli basati sui dati (Data-Driven Models)

I modelli basati sui dati sfruttano l'analisi e l'interpretazione dei dati ottenuti tramite diversi test sul componente d'interesse per conoscere il comportamento di un sistema. La filosofia di questa metodologia si basa sull'impiego di un approccio neutrale ed implicito per cercare di comprendere al meglio un sistema basandosi sui dati grezzi raccolti dai test fisici. Grazie a questi modelli è possibile effettuare delle previsioni anche in assenza di esperienza reale.

Secondo la teoria dei sistemi comportamentali, un sistema viene definito come un insieme di traiettorie che rappresentano un modello di comportamento. Tramite la metodologia dei modelli a base di dati, invece, il sistema è privo di una qualsiasi rappresentazione specifica e non viene visto come una serie di traiettorie. Per questo motivo l'utilizzo di algoritmi utili all'identificazione di pattern all'interno dei dati e alla previsione riguardante nuovi dati sono fondamentali per questo tipo di approccio prognostico.

La modellazione basata sui dati sfrutta l'applicazione di misure empiriche e tecniche di machine learning per sviluppare dei modelli che possano predire dei guasti prima che avvengano e la loro complessità è inferiore a quella dei modelli fisici, a discapito di un'affidabilità inferiore. Il processo seguito per questa tipologia di modelli viene riportato in Figura 5.

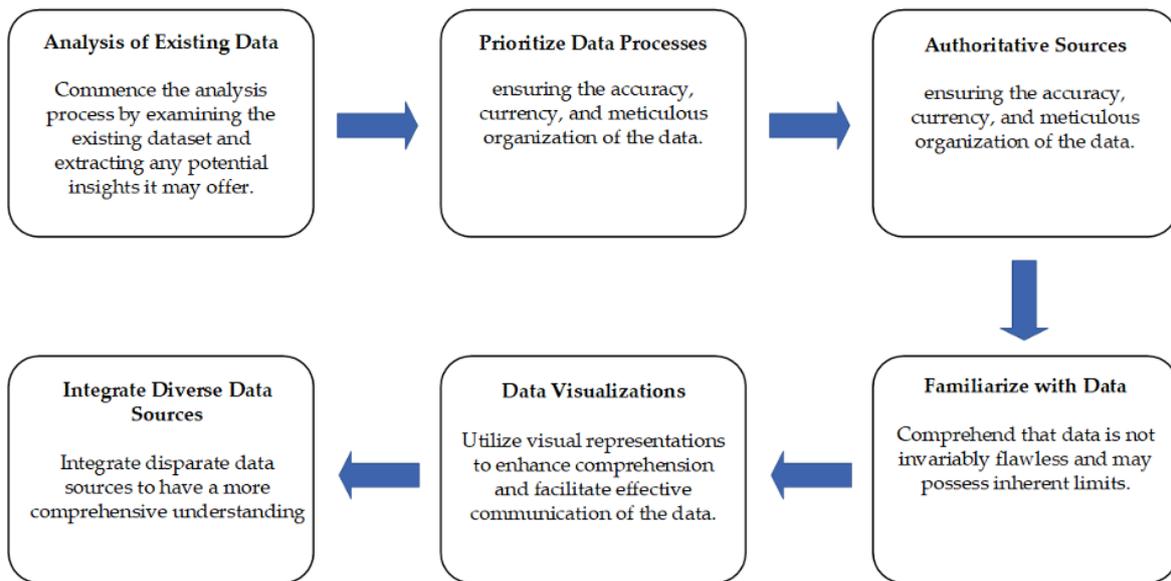


Figura 5 : Processo Data-driven Model [7]

I modelli basati sui dati hanno l'obiettivo principale di acquisire conoscenza dai dati a loro disposizione, considerando una distribuzione di input sconosciuta. Attualmente sono in fase di sviluppo diverse tecniche computazionali e statistiche utili per la progettazione di algoritmi guidati dai dati, i cui obiettivi corrispondono a quelli degli algoritmi in grado di migliorare le proprie prestazioni.

Vengono ora riportati e descritti alcuni strumenti utili per la modellazione basata sui dati usati maggiormente nel contesto attuale.

2.2.1 Metodi di apprendimento e Neural Networks

I Neural Networks Artificiali (ANNs) furono originalmente sviluppati da McCulloch e Pitts nel 1943, con l'obiettivo di simulare i processi cognitivi legati all'apprendimento del cervello umano. I Neural Networks (NNs) sono dei modelli computazionali derivati dagli ANNs e, a differenza del cervello umano, lavorano sfruttando una rete di unità interconnesse pre-elaborate per poter processare dei dati. I Neural Networks sono anche conosciuti come scatole nere o "black box", poiché se gli vengono forniti una serie di esempi da risolvere, essi genereranno una proposta di azioni da compiere, che non possono essere derivate da nessuna funzione matematica conosciuta. I NNs consistono, infatti, in una serie di neuroni artificiali accoppiati per generare una rete connessa ed in grado di modificare la loro struttura in risposta a degli input interni o esterni. Questa tipologia di strumento non viene progettato per risolvere un determinato problema, ma acquisisce la capacità di affrontare ogni tipologia di problema sfruttando una tipologia di addestramento ed apprendimento tramite l'utilizzo di esempi. La serie di dati che viene sfruttata come set di addestramento è composta da valori di input accoppiati con i rispettivi valori di output. Gli algoritmi utilizzati

per l'addestramento dei NNs sono diversi e a titolo di esempio viene riportata la struttura di un processo ANNs *feedforward* in Figura 6.

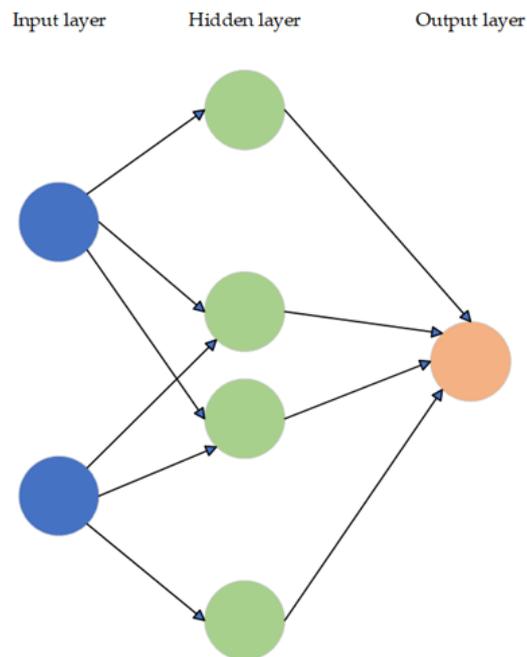


Figura 6 : Processo semplificato di funzionamento di un ANNs feedforward [7]

Il processo completo prevede che i dati di addestramento vengano forniti come input nell'*Input Layer* e propagati nell'*Hidden Layer*. Qui viene effettuato un passaggio in avanti nella fase di addestramento generando una predizione H che viene fornita all'*Output Layer*, il quale calcola la perdita O confrontando la predizione ricevuta con quella obbiettivo dell'addestramento. A questo punto l'*Outer Layer* trasmette questo risultato al *Layer* precedente e il processo rinizia. Questa sequenza di movimenti di dati all'interno di una rete neurale viene rappresentata nel dettaglio in Figura 7.

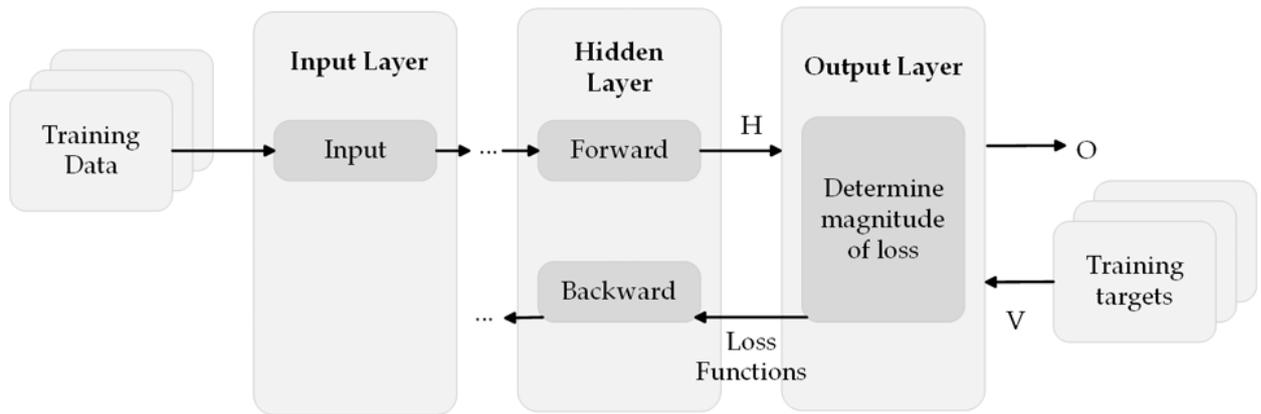


Figura 7: Processo dei movimenti dei dati all'interno di una rete neurale [7]

Durante la fase di test vengono valutate le capacità predittive della rete neurale modificando i pesi delle sue connessioni, in modo da determinare la sua accuratezza. I dati di addestramento vengono generati utilizzando dei set di dati storici e l'architettura della rete viene scelta in base alla sua adeguatezza. Il passo successivo, cruciale nell'implementazione delle reti neurali, riguarda il processo di apprendimento.

Durante il processo di apprendimento, la rete neurale riceve dei percorsi di informazioni attraverso le unità di input (*Input Layer*), le quali attivano le unità nascoste (*Hidden Layer*) che propagano i segnali alle unità di output (*Output Layer*). Le NN prevedono, infatti, che ogni unità riceva un input da un'altra unità posizionata alla sua sinistra. Tale segnale viene moltiplicato per il peso associato alla connessione attraverso cui passa ed una volta che la somma di tutti questi segnali accumulati nella rete supera un valore di soglia, allora l'unità viene attivata insieme a tutte le altre unità a cui è collegata.

Le reti neurali sono in grado di apprendere informazioni grazie ad un meccanismo di retroazione conosciuto come "*backpropagation*", un processo che prevede il confronto tra l'output che è stato generato dalla rete e l'output richiesto. La differenza tra questi due valori viene fornita al Layer precedente, seguendo un processo a ritroso dall'unità di output a quella di input, in modo da aggiornare i pesi delle connessioni tra le due unità all'interno della rete. Il processo di apprendimento viene quindi migliorato nel tempo arrivando, infine, ad uno stato dove la differenza è nulla e la rete raggiunge una condizione ottimale per comprendere i diversi percorsi e le relazioni sottostanti.

Una delle principali suddivisioni riguardanti le reti neurali è legata alla direzione in cui viene propagata l'informazione. Questa può seguire un comportamento "*feedforward*" oppure "*feedback*". La prima tipologia segue un'architettura per cui il segnale viene propagato in maniera unidirezionale dall'*Input Layer* all'*Output Layer*. Le reti neurali "*feedforward*" sono composte da un'unica unità di input e di output, mentre le unità nascoste possono essere molte oppure nessuna. Il flusso di informazioni segue una fase iniziale di apprendimento, che avviene quando la rete viene istruita e una fase di funzionamento operativo, che avviene dopo che il processo di addestramento è stato concluso. Le reti neurali "*feedback*"

utilizzano, invece, una memoria per elaborare in modo efficace le sequenze di input e dei cicli presenti all'interno della rete facilitano la trasmissione dei segnali in entrambe le direzioni.

Esistono tre tipologie in cui i metodi di apprendimento possono essere classificati: l'apprendimento supervisionato, l'apprendimento semi-supervisionato e l'apprendimento non supervisionato.

- L'apprendimento supervisionato utilizza una tecnica di apprendimento basata sull'analisi di esempi noti derivati da dati passati, in modo da permettere alla rete neurale di stabilire delle connessioni tra gli input e gli output. Questa tecnica sfrutta dei dataset per addestrare degli algoritmi il cui obiettivo è classificare attentamente i dati e prevedere i risultati. Le coppie di input e output vengono fornite all'algoritmo con lo scopo di fargli apprendere una regola generale tramite cui esso può stabilire una connessione tra input e output forniti. L'apprendimento supervisionato viene utilizzato in diversi campi tra cui quello aerospaziale ed in particolare nella manutenzione predittiva dei sistemi dei velivoli.
- L'apprendimento semi-supervisionato è una metodologia molto utilizzata nel campo del machine learning, poiché permette di utilizzare una quantità limitata di dati etichettati manualmente durante le fasi più dispendiose dell'apprendimento supervisionato. Nelle fasi successive vengono invece utilizzati numerosi dati non etichettati, i quali vengono sfruttati durante l'apprendimento non supervisionato. Tale vantaggio risulta molto utile a causa della comparsa di modelli linguistici di dimensioni sempre maggiori, che richiedono una grande quantità di dati per il loro addestramento. Questa metodologia di apprendimento si pone in una posizione intermedia tra l'apprendimento supervisionato e l'apprendimento non supervisionato, in quanto il suo obiettivo è sfruttare dati non etichettati per migliorare l'efficacia di un modello addestrato su dati etichettati.
- L'apprendimento non supervisionato è una tipologia di apprendimento per cui solo i dati di input vengono forniti, senza ulteriori informazioni. L'approccio è basato su degli studi che hanno osservato come delle simulazioni simili tendono ad attivare dei neuroni vicini tra loro, mentre delle simulazioni diverse tra loro attivano dei neuroni distanti tra loro. Questi algoritmi vengono utilizzati principalmente per l'analisi dei dati, permettendo di suddividerli in categorie diverse in base alle loro caratteristiche comuni o alle loro differenze.

L'efficienza dei modelli di machine learning, che essi siano ad apprendimento supervisionato, semi-supervisionato o non supervisionato, dipende da alcuni fattori come la qualità e la quantità dei dati utilizzati per addestrare il modello, la scelta della tipologia di algoritmo e la complessità del problema. L'accuratezza degli algoritmi ad apprendimento supervisionato è sicuramente migliore rispetto a quella degli algoritmi ad apprendimento non supervisionato, poiché utilizzano dati etichettati per allenare il modello che consentono di avere una conoscenza migliore e migliorare la precisione delle previsioni. D'altra parte, l'accuratezza di questa tipologia di algoritmi può essere limitata dal numero di dati

etichettati accessibili. Gli algoritmi ad apprendimento non supervisionato si basano, come detto, sull'identificazione di pattern e correlazioni intrinseche all'interno dei dati e perciò la loro accuratezza può variare in base alla complessità del compito e alla qualità dei dati. Gli algoritmi ad apprendimento semi-supervisionato, invece, utilizzando una piccola quantità di dati etichettati per aumentare l'efficienza dei modelli allenati con dati non etichettati, hanno un'accuratezza che supera quella degli algoritmi ad apprendimento non supervisionato, anche se il risultato dipende anche in questo caso dalla disponibilità e dalla quantità di dati etichettati. Per tali motivi, non è, dunque, possibile definire un unico metodo generale per trovare la tipologia di algoritmo più precisa.

2.2.2 K-means

L'algoritmo *k-means* si basa sulla metodologia di apprendimento non supervisionato e prevede una suddivisione dei dataset che gli vengono forniti in k gruppi diversi, prendendo in considerazione la similarità e le metriche di distanza tra i punti dati, i quali vengono inseriti all'interno del raggruppamento con il valor medio più vicino a questi. L'obbiettivo dell'algoritmo è minimizzare la somma totale delle distanze quadratiche tra i punti e il centroide del raggruppamento, con quest'ultimo che viene calcolato facendo la media aritmetica dei punti presenti all'interno del gruppo. Il metodo *k-means* consiste, quindi, nel riassegnare i punti nei vari gruppi finché la condizione di convergenza non viene soddisfatta.

Esistono diverse tecniche per stabilire quale sia il numero ottimale di raggruppamenti nel metodo *k-means* tra i quali il metodo del "gomito", che consiste nel calcolare la somma del quadrato delle distanze per diversi valori di k e nel selezionare il valore di k per cui la riduzione della somma delle distanze quadratiche inizia a stabilizzarsi. Un'altra tecnica che permette di valutare il grado di separazione tra dei gruppi generati da un algoritmo *k-means* è l'analisi "Silhouette", la quale consente anche di determinare dei parametri come il numero ottimale di raggruppamenti.

2.2.3 Fuzzy Logic

La *Fuzzy Logic* è spesso utilizzata come soluzione per tutti quei problemi in cui i metodi matematici standard non possono essere implementati a causa della complessità numerica; infatti, facilitano la caratterizzazione e il controllo di quei sistemi che mancano di un preciso e riconoscibile modello definito. Questi algoritmi sono un'estensione della logica booleana, ampliando l'interpretazione della teoria degli insiemi classica e introducendo il concetto di grado nella verifica delle condizioni. La *Fuzzy Logic* offre, quindi, una maggiore flessibilità nel ragionamento, permettendo l'esistenza intermedia di condizioni tra vero e falso e consentendo una migliore gestione degli errori e delle incertezze all'interno dei processi decisionali. Non è, dunque, sempre necessaria una giustificazione probabilistica ed un approccio spesso utilizzato consiste nel generalizzare i concetti della logica multivalore, cercando di mantenere parte della struttura algebrica originale.

La *Fuzzy Logic* viene attualmente sfruttata per componenti elettronici che richiedono un sistema di controllo efficace, ma in cui il controllo ottimale non è una priorità necessaria.

2.2.4 Alberi decisionali

Un albero decisionale è uno strumento che rappresenta un modello sotto forma di “*albero*” per illustrare diverse opzioni e le loro corrispondenti conseguenze e viene utilizzato per determinare l’esito di eventi caratterizzati da incertezza. La sua struttura è composta da dei nodi interni che svolgono il compito di valutare gli esiti di alcuni test, dai “*rami*” dell’albero che rappresentano i possibili esiti dei test e dai nodi “*foglia*” che simboleggiano le etichette di classe, cioè le decisioni prese dopo aver valutato tutte le caratteristiche. I percorsi che vanno dalle radici ai nodi “*foglia*” sono rappresentazioni dei criteri di categorizzazione.

Gli algoritmi di apprendimento basati sugli alberi decisionali sono spesso considerati come una tipologia di algoritmo molto potente e vengono utilizzati per apprendimenti supervisionati. In particolare, vengono sfruttati in situazioni di classificazione quando esiste una variabile target predefinita. Essi hanno una struttura intrinseca caratterizzata da una tipologia “*if...then...else*”, che li rende molto compatibili con le strutture programmate. Inoltre, risultano molto idonei per compiti di classificazione quando le “*features*” vengono esaminate metodicamente per determinare una categoria definitiva.

Per questa tipologia di algoritmi vengono spesso utilizzati una serie di termini specifici legati al concetto di albero fisico che fanno riferimento a concetti matematici. Tra questi si trovano: il “*nodo radice*” che funge da rappresentazione dell’intero campione e viene successivamente suddiviso in due o più sottoinsiemi che mostrano caratteristiche simili tramite lo “*splitting*”, termine utilizzato per rappresentare proprio il processo di suddivisione di un nodo in più sotto-nodi; il “*nodo decisionale*” è definito come un sotto-nodo che si biforca in altri sotto-nodi; i “*nodi foglia*” o “*terminali*” sono, invece, dei nodi che non subiscono alcuna ulteriore suddivisione; la “*potatura*” si riferisce alla rimozione dei sotto-nodi da un “*nodo decisionale*”; un “*ramo*” o “*sotto-albero*” rappresenta un componente o una suddivisione specifica all’interno di una più ampia struttura. Infine, un concetto fondamentale è legato a quello di “*nodo padre*” e “*nodo figlio*”. In una struttura gerarchica come quella di un albero decisionale il “*nodo padre*” è un nodo suddiviso e connesso direttamente con dei nodi più piccoli, chiamati “*nodi figli*”.

2.2.5 Support Vector Machine (SVM)

Le *Support Vector Machine* (SVM) sono una tipologia di algoritmi di apprendimento supervisionato utilizzati per la classificazione, regressione e identificazione. I *support vector* sono le coordinate specifiche delle osservazioni individuali ed ogni punto all’interno dell’algoritmo è rappresentato in uno spazio n -dimensionale, dove n è il numero di *features*. La rappresentazione numerica di ogni caratteristica corrisponde al valore di una certa coordinata.

Questi algoritmi risultano molto efficienti in ambienti ad alta dimensione, in cui il numero di dimensioni supera anche il numero di campioni. D’altra parte, quando il numero di *features* è molto più elevato al numero di campioni c’è il rischio di *overfitting*. Un ulteriore svantaggio è legato al fatto che le SVM non forniscono direttamente stime e valori di probabilità, ma

questi valori devono essere calcolati utilizzando la validazione incrociata a cinque pieghe, un metodo computazionale molto costoso.

Le reti neurali artificiali, gli algoritmi *k-means*, la *Fuzzy Logic*, gli alberi decisionali e le SVM sono tutte tecniche di machine learning utilizzate per compiti di classificazione ed ognuna di queste è caratterizzata dal proprio meccanismo di funzionamento. La scelta della tecnica adeguata può essere complessa e deve tenere in considerazione le dimensioni e le proprietà dei dataset, la complessità del problema trattato e le prestazioni, l'interpretabilità e la sostenibilità del modello. Un metodo per effettuare questa scelta può essere quello di utilizzare diversi algoritmi e valutare la loro prestazioni sui dataset forniti. La tecnica della validazione incrociata può, inoltre, essere sfruttata per valutare l'efficacia di ciascun metodo. La complessità del modello è un ulteriore fattore da tenere in considerazione. Gli algoritmi più sofisticati permettono di individuare dei pattern più complessi all'interno dei dati, al costo di maggiori difficoltà in termini di manutenzione ed interpretabilità.

La scelta del modello più adatto dipende dunque dalle esigenze specifiche e dagli obiettivi da raggiungere.

2.2.6 Algoritmi di rilevamento delle anomalie

Gli algoritmi di rilevamento delle anomalie sono delle tecniche utili per rilevare e identificare delle deviazioni da quelli che sono i comportamenti attesi. Essi fanno affidamento sull'intelligenza artificiale (AI), la quale viene applicata per rilevare tentativi di intrusioni identificando pattern insoliti o per stimare lo stato di salute di un sistema. Nel campo aerospaziale viene sempre più utilizzata per identificare guasti all'interno degli ambienti operativi.

Esistono tre principali metodi per il rilevamento delle anomalie: il rilevamento delle anomalie basato sulla densità, il rilevamento dell'anomalia in base ai raggruppamenti e il rilevamento dell'anomalia basato sull'algoritmo SVM.

Il rilevamento dell'anomalia basato sulla densità fa riferimento ad un metodo utilizzato nell'analisi dei dati per identificare delle irregolarità in un dataset in base alla distribuzione di densità dei dati. Questa tecnica si basa sull'utilizzo dell'algoritmo *k-Nearest Neighbors* (kNN), secondo cui i dati tipici sono tutti concentrati in prossimità l'uno dell'altro, mentre le anomalie si trovano ad una notevole distanza. La prossimità tra un insieme di punti dati viene valutata in base ad una distanza Euclidea o ad una metrica simile, che dipende dalla natura dei dati analizzati. L'algoritmo kNN è un approccio semplice e non parametrico che viene comunemente utilizzato per la classificazione dei dati e per identificare somiglianze tra i punti dati, i quali possono anche essere classificati in base alla loro densità relativa.

Il rilevamento di anomalie in base ai raggruppamenti si basa sulla premessa fondamentale che i dati simili tra loro tendono a far parte dello stesso gruppo e ciò viene determinato in base alla loro prossimità ai centroidi locali. La tecnica *k-means* viene utilizzata per questa tipologia di rilevamento, infatti, essa genera *k* gruppi di dati con somiglianze tra loro e le anomalie possono essere identificate in quei dati che non appartengono a questi gruppi.

Il rilevamento di anomalie basato sull'algoritmo SVM fa riferimento, ovviamente, a quest'ultimo, che viene utilizzato principalmente per gli apprendimenti supervisionati. Esistono, però, alcune varianti come la *Core Vector Machine* (CVM), che permettono l'identificazione di anomalie anche in processi di apprendimento non supervisionati, in cui i dataset di allenamento non contengono alcun dato etichettato. Questo metodo utilizza il dataset di addestramento per imparare a classificare i campioni di dati normali e, successivamente, adatta i suoi parametri in base al tipo di test specifico per individuare eventuali anomalie che si trovano al di fuori dell'intervallo stabilito.

2.2.7 Tecniche numeriche convenzionali

Esistono diverse tecniche numeriche che possono essere utilizzate per i modelli basati sui dati come il filtro di *Kalman*, il filtro di *Kalman* esteso, il filtro a particelle e la regressione, ma risultano molto complesse ed esulano dall'obiettivo di questa tesi, per cui verrà fatta solo una breve descrizione di una di queste tecniche ossia il filtro *Kalman*.

Il filtro di *Kalman* (KF), proposto da *Kalman* nel 1960, è stato negli anni utilizzato in diversi campi pratici, in particolare in quello aeronautico e quello aerospaziale. Questo metodo sfrutta una serie di misurazioni nel tempo, comprensive di rumore statistico e altre fonti di errore, per fornire delle previsioni su delle variabili sconosciute in modo da avere maggiore accuratezza rispetto ad una stima generata da una singola osservazione. Il funzionamento del filtro KF è suddiviso in due fasi: una fase di previsione e una di aggiornamento. Durante la prima fase il filtro genera delle stime sulle variabili di stato correnti e sulle loro incertezze. Valutate queste stime, esse vengono modificate utilizzando una media ponderata, in modo che le stime con livelli di incertezza maggiori vengano subito individuate.

Il filtro radice quadrata (SRF) è considerato un'alternativa più affidabile al filtro di *Kalman*, in quanto utilizzando delle trasformazioni ortogonali numericamente stabili risulta più preciso nelle stime. Il filtro di *Kalman* ha, infatti, problemi con la stabilità numerica e diverse strategie sono state adottate per affrontare questo problema: aumentare il livello di precisione matematica; utilizzare la tecnica SRF; inizializzare correttamente la covarianza per ridurre le fluttuazioni; impiegare un filtro con memoria decrescente; utilizzare del rumore ipotetico. [11]

Alcuni di questi problemi legati al filtro di *Kalman* sono stati risolti nel tempo grazie all'aumento del costo computazionale impiegabile, ma rimangono comunque diverse limitazioni legate a questo metodo come l'assunzione di linearità del sistema. Per questo motivo, vengono spesso proposte delle versioni alternative del KF, come il filtro di *Kalman* esteso, utilizzato per i sistemi non lineari, per superare queste sfide numeriche. Ovviamente, l'implementazione di questi metodi richiede un livello più complesso di competenza.

2.3 Modelli ibridi

La prognostica basata sui modelli ibridi è una combinazione di diversi approcci manutentivi impiegati per prevedere lo stato di salute futuro di un determinato sistema. L'impiego di questa tecnica permette di aumentare la precisione nella previsione dei guasti e migliorare l'affidabilità e la sicurezza dei sistemi. Infatti, l'integrazione di più metodologie consente di valutare in modo completo lo stato di salute di un componente oltre ad ottimizzare la manutenzione predittiva. Le tecniche impiegate nella prognosi ibrida sono quelle basate sui dati, sui modelli fisici e sulla conoscenza.

Gli approcci prognostici ibridi differiscono notevolmente dalle metodologie tradizionali di prognostica, in quanto utilizzano diverse strategie per predire la condizione futura di un sistema, al contrario delle tecniche classiche che spesso fanno riferimento solo ad una singola strategia per fare una previsione. Ad esempio, l'approccio basato sui dati richiede un'estrazione manuale delle *features* da dei dati grezzi di un sensore, processo spesso laborioso e che non assicura l'identificazione delle caratteristiche ottimali per ogni misurazione. Invece, l'approccio ibrido permette di utilizzare una profonda struttura di rete neurale ibrida che è in grado di estrarre le *features* direttamente dai dati grezzi durante il periodo di allenamento.

L'integrazione dei modelli fisici con i modelli basati sui dati viene spesso impiegata ed esistono diverse tecniche con cui i due algoritmi possono lavorare insieme. Un primo approccio consiste nell'utilizzare un modello fisico per estrarre delle *features* che possono poi essere impiegate in un modello basato sui dati, con quest'ultimo che, attraverso l'apprendimento dei dati, è in grado di generare delle previsioni. Un secondo approccio può prevedere l'impiego di un modello fisico per effettuare una prima stima del comportamento di un sistema, la quale viene poi affinata grazie all'utilizzo di un modello basato sui dati, addestrato con i dati reali del sistema stesso. Entrambi gli approcci sfruttano un modello fisico per incorporare delle conoscenze pregresse sul caso di studio, in modo da migliorare il processo di apprendimento del modello basato sui dati. Grazie a questa combinazione è possibile aumentare l'accuratezza delle previsioni rispetto all'utilizzo singolo di una sola strategia.

La scelta della tipologia dei modelli da utilizzare ed integrare all'interno di un modello ibrido viene riportata nel *flowchart* rappresentato in Figura 8 e basato su uno studio condotto da "Liao, L.; Köttig, F. Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction. *IEEE Trans. Reliab.* 2014, 63, 191–207" [\[10\]](#).

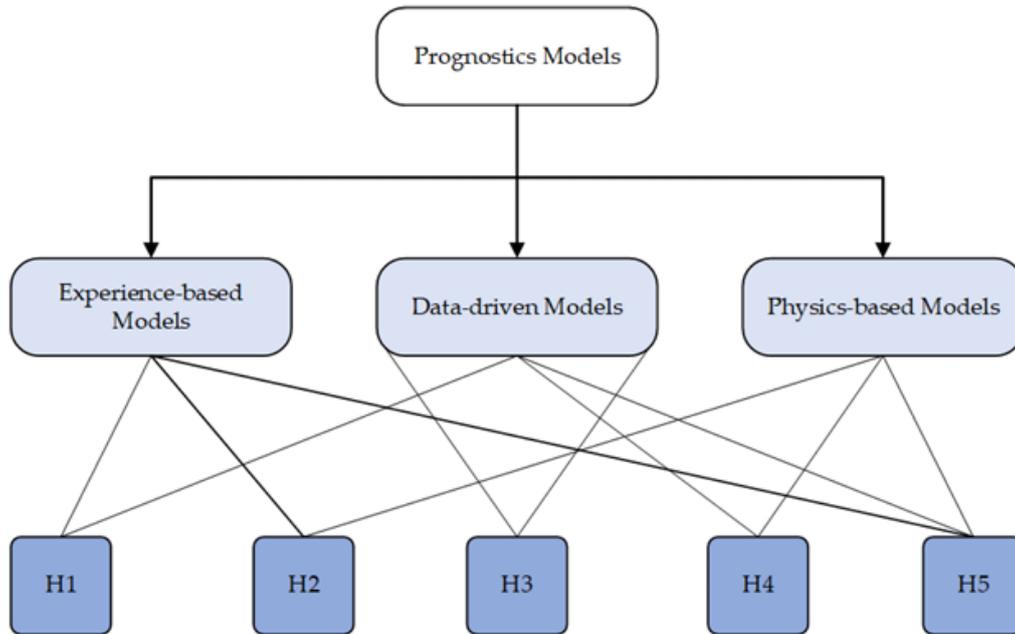


Figura 8 : Flowchart dei modelli prognostici ibridi [7]

Lo schema riportato presenta i tre modelli utilizzabili e le possibili combinazioni tra questi. La prima di queste prevede l'integrazione di un modello basato sull'esperienza e uno basato sui dati, mentre la seconda un modello basato sull'esperienza ed un modello fisico. La terza combinazione prevede, invece, la combinazione di un modello basato sui dati con un altro modello basato sui dati, cosa che può apportare significativi benefici. La quarta integrazione prevede un modello basato sui dati ed un modello fisico e, infine, la quinta prevede una combinazione di tutti i modelli insieme, che, secondo lo studio, genera dei risultati molto favorevoli.

In particolare, tra tutte queste combinazioni, l'implementazione di approcci basati sui dati e sui modelli fisici risulta particolarmente efficace nella stima della *RUL*. I modelli fisici, infatti, si basano sulla comprensione dei principi fisici per generare delle stime accurate e, nel caso di uno studio del degrado di un componente, esso utilizza un'equazione analitica nota come modello di degradazione, che dovrebbe rappresentare accuratamente il processo di degradazione. L'implementazione dei modelli basati sui dati, che sfruttano dati passati registrati sul sistema di studio, permette di migliorare l'accuratezza della previsione e ridurre l'incertezza. Una rappresentazione schematica dell'implementazione di questi due singoli metodi all'interno di un modello ibrido viene riportata in Figura 9.

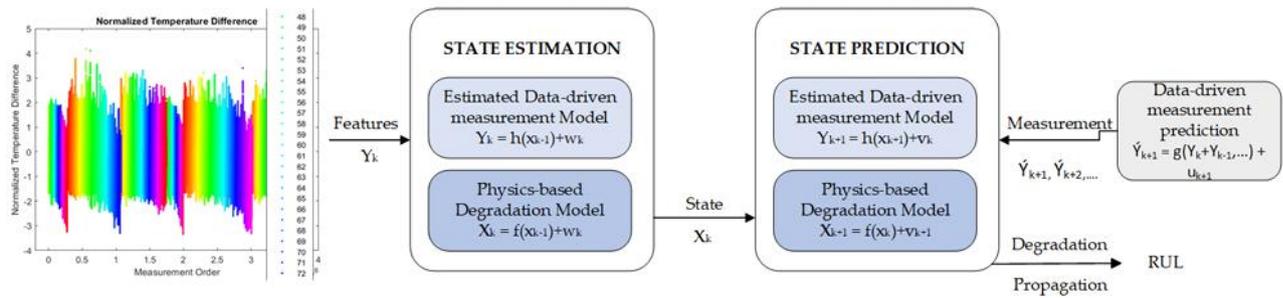


Figura 9 : Modello ibrido con implementazione di modello fisico e modello basato sui dati [12]

Lo schema rappresenta una visione preliminare dell'intricata relazione che esiste nell'implementazione delle due metodologie di algoritmi. Il processo prognostico prevede l'utilizzo di due tecniche basate sui dati, una per stimare il modello di misurazione e una per prevedere misurazioni future in scenari a lungo termine, e di una tecnica tradizionale di modello fisico, necessaria per stimare indirettamente lo stato interno del sistema.

Le metodologie prognostiche ibride integrano i vantaggi dei singoli metodi per raggiungere una maggior precisione ed affidabilità nel rilevamento di un guasto, integrando la robustezza di ogni algoritmo utilizzato. Tuttavia, bisogna sottolineare come alcune delle limitazioni presenti nei singoli approcci siano comunque presenti nei modelli ibridi. Inoltre, un'ulteriore limitazione è legata proprio alla sfida di sviluppare dei modelli ibridi che integrino efficacemente i benefici di entrambe le tecniche. La complessità di questi modelli richiede, infatti, un elevato livello di competenza per la loro creazione e manutenzione continua, oltre a necessitare di una grande quantità di dati per l'addestramento e la validazione. Per affrontare queste sfide, diversi studi su metodologie innovative sono stati portati avanti in questi anni. Nel settore aeronautico, ad esempio, viene sfruttata l'implementazione di funzioni di densità di probabilità (PDF), le quali possono essere utilizzate per valutare la probabilità che un sistema si trovi in uno stato sano o di degrado. Tale stima viene generata considerando gli indicatori di salute (HI) associati al sistema, che permettono ai modelli ibridi di prevedere e prevenire guasti con una maggiore precisione.

2.4 Sfide nella Prognostica

La prognostica, come si è visto, è un campo di studio che offre numerosi vantaggi e benefici nel campo manutentivo, ma essendo una tecnologia ancora relativamente "giovane", presenta ancora svariati ostacoli che deve affrontare e superare.

Uno tra tutti è la selezione e la corretta localizzazione dei sensori con cui vengono registrati i dati, poiché il processo di acquisizione dei dati è una parte fondamentale del processo prognostico. Questo, infatti, può essere compromesso da un'inaccurata lettura dei risultati, a causa di un'errata selezione o un errato dislocamento dei sensori utilizzati. Essendo questi

dei componenti fondamentali nell'individuazione di guasti all'interno dei sistemi, sono stati oggetto di diversi studi per aumentare la loro accuratezza ed affidabilità. Delle soluzioni proposte prevedono l'utilizzo di sensori ridondanti e l'implementazione di tecniche di validazione. Anche l'estrazione delle *features* è un passaggio di particolare importanza, che deve esser effettuato in maniera molto accurata, in quanto consente di raccogliere delle informazioni collegate direttamente all'insorgere del danno.

L'incertezza e la valutazione dell'accuratezza legate alla prognostica sono delle importanti sfide che vanno analizzate attentamente. Un grosso ostacolo presente in questo campo è, infatti, legato alla formulazione di approcci che siano in grado di affrontare efficacemente le incertezze incontrate in delle situazioni pratiche. Le fonti di incertezza a cui è collegata la prognostica vengono riportate in Figura 10.

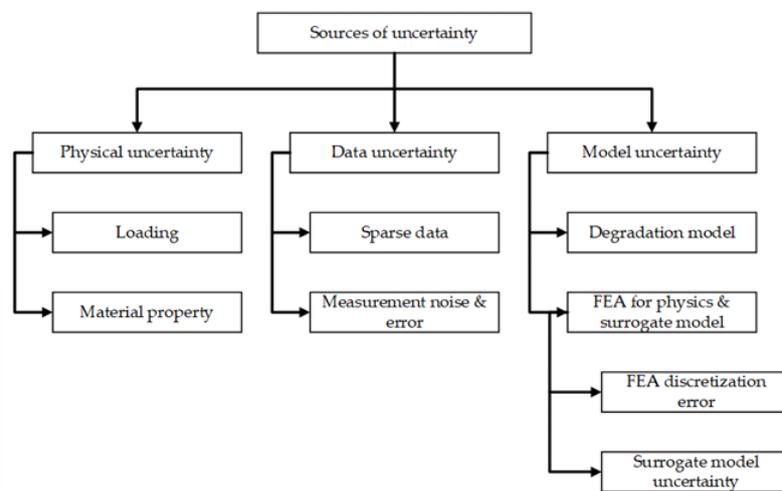


Figura 10 : Classificazione delle incertezze presenti nella prognostica [7]

La presenza di incertezze all'interno del processo può portare a delle discrepanze tra i risultati generati e la condizione reale; pertanto, lo sviluppo di approcci che possano descrivere in modo dettagliato questi limiti è di fondamentale importanza. Questi risultano essenziali per determinare e misurare l'accuratezza di un sistema di prognosi, ma al giorno d'oggi non c'è ancora un accordo generale su quali metodologie siano più adatte per valutare le performance prognostiche.

Per affrontare queste sfide nel migliore dei modi, è necessaria la collaborazione tra tutti i settori chiamati in causa, che nell'ambito aeronautico vanno dai produttori di aeromobili, ai manutentori, agli enti regolatori e agli sviluppatori tecnologici.

3 MODELLO ATTUATORE ELETTROMECCANICO

3.1 Gli attuatori elettromeccanici in generale

Gli attuatori elettromeccanici (EMA) rappresentano una soluzione tecnologicamente avanzata per la generazione e il controllo di movimento in diversi settori ingegneristici ed in particolare in quello aerospaziale. A differenza degli attuatori idraulici e pneumatici, infatti, gli EMA utilizzano esclusivamente componenti elettrici ed elettronici per convertire energia elettrica in meccanica, in quanto seguono la filosofia “*more electric*”, consentendo di eliminare i vari sistemi complessi di tubazioni e pompe presenti negli attuatori più classici. Gli EMA vengono impiegati in numerosi sistemi aeronautici, tra i quali le superfici di controllo, i carrelli di atterraggio e i meccanismi di regolazione della spinta, offrendo una soluzione avanzata per il movimento di componenti critici.

All'interno degli attuatori elettromeccanici la conversione di energia elettrica in energia meccanica avviene attraverso un motore elettrico, il quale, accoppiato con un sistema di trasmissione meccanico, consente il movimento lineare o rotativo del sistema. Il motore elettrico, che funziona da alimentatore per l'intero componente, è generalmente di tipo *brushless*, i quali sono particolarmente utilizzati in ambito aeronautico, grazie alla loro maggiore efficienza e durata e alla loro ridotta necessità di manutenzione. Questi motori elettrici funzionano grazie all'interazione tra un campo magnetico rotante generato dagli avvolgimenti dello statore esterno e i magneti permanenti del rotore interno. Il passaggio di corrente attraverso gli avvolgimenti, noto come commutazione elettronica, permette di controllare con precisione la velocità e la coppia erogata dal motore, rendendolo stabile ed efficiente. A differenza dei motori a spazzole, nei motori brushless la commutazione viene gestita da un controller elettronico, il quale regola il flusso di corrente negli avvolgimenti per generare il campo magnetico rotante. In questo modo viene eliminato il problema legato all'usura delle spazzole e alle perdite di attrito di motori tradizionali, migliorando quindi l'affidabilità e riducendo la necessità di manutenzione. Grazie a questi motori elettrici, inoltre, è possibile generare una densità di potenza molto superiore, rendendoli ideali per il settore aeronautico, dove peso e dimensioni sono fondamentali.

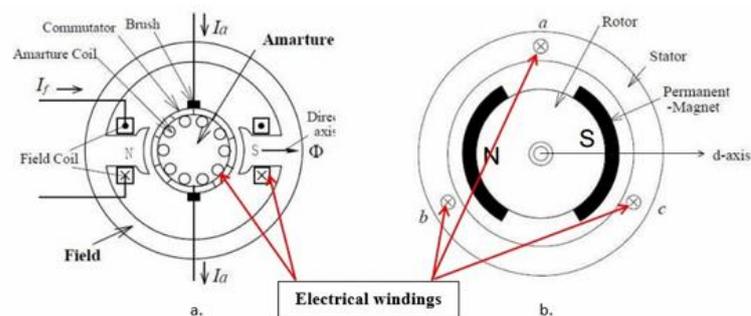


Figura 11 : (a) Motore spazzolato DC, (b) Motore brushless [4]

Il motore elettrico genera un movimento rotativo, che deve essere linearizzato tramite l'utilizzo di un riduttore meccanico, il quale ha il compito di adattare la velocità e la coppia trasmessa al carico. Un ulteriore componente di grande importanza è la vite a ricircolo di sfere, un meccanismo che consente di trasformare il moto rotante del motore elettrico in un moto lineare, riducendo l'attrito e aumentando l'efficienza del sistema. Il suo funzionamento, come riportato in Figura 12, prevede un albero filettato esternamente, corrispondente alla vite, una madrevite filettata internamente, chiamata "chiocciola" e le sfere d'acciaio, le quali sono l'elemento fondamentale del sistema in quanto permettono l'effettiva trasformazione del moto da rotazionale a lineare. Esse minimizzano l'usura, migliorando la trasmissione della forza e si interpongono tra vite e madrevite, percorrendo il percorso elicoidale e quello di ricircolo. Quest'ultimo viene percorso una volta che le sfere raggiungono la fine del tratto elicoidale, in modo da poter ritornare al punto di partenza della vite. Rispetto ai riduttori a vite trapezoidale, la vite a ricircolo di sfere offre una migliore efficienza, consentendo un utilizzo più efficace dell'energia a disposizione e una maggiore durata della vita operativa del componente. Inoltre, questa tipologia di sistema consente di operare con maggiore precisione e ripetibilità, elemento essenziale nel campo aeronautico, dove la precisione del controllo delle superfici mobili è necessario per garantire la sicurezza del volo.

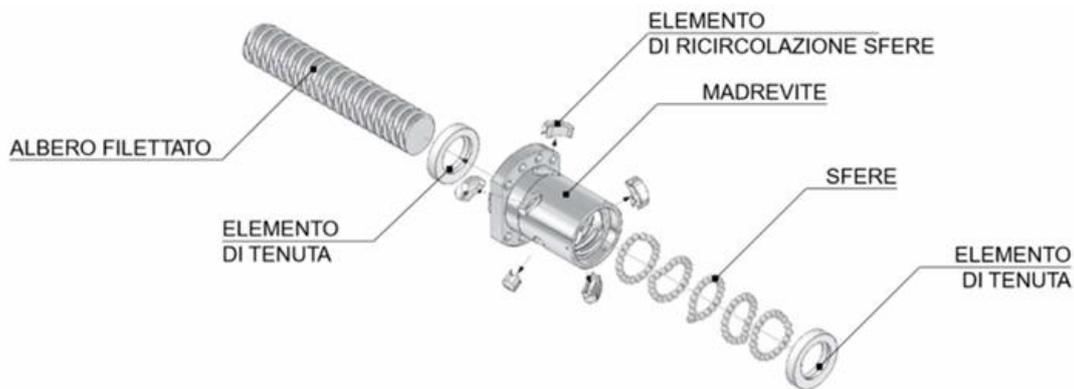


Figura 12 : Vite a ricircolo di sfere [13]

Gli attuatori elettromeccanici hanno numerosi vantaggi rispetto agli attuatori tradizionali, tra i quali la riduzione del peso e della complessità del sistema. A differenza degli attuatori idraulici, infatti, che richiedono la presenza di pompe, serbatoi e tubazioni per il fluido, gli attuatori elettromeccanici non necessitano della presenza di tutti questi componenti, semplificando l'integrazione all'interno del velivolo, le dimensioni, il peso e i costi. Un ulteriore vantaggio rispetto agli attuatori classici è legato alla minore manutenzione richiesta, poiché non ci sono problemi legati alle perdite di fluido o alla sua contaminazione, che possono compromettere il funzionamento dei sistemi idraulici, e poiché l'assenza di tubazioni evita problematiche legate alle perdite o alla rottura di queste.

Gli EMA risultano più efficienti anche dal punto di vista energetico. Infatti, essi assorbono energia solamente durante il movimento e non devono mantenere costantemente in pressione il fluido. Per questo risultano particolarmente adatti per quelle applicazioni in cui il consumo energetico è fondamentale, come nei velivoli elettrici, oltre al fatto che i loro minori consumi si traducono in un minore impatto ambientale, fattore che attualmente sta assumendo un'importanza sempre maggiore nell'ambito aeronautico.

Dal punto di vista del controllo, gli attuatori elettromeccanici favoriscono una maggiore precisione e flessibilità dovuti all'utilizzo di sensori di posizione e di forza, che consentono di regolare l'attuatore con estrema accuratezza. Inoltre, l'integrazione di sistemi di controllo digitale permettono di effettuare una diagnostica avanzata sugli EMA, in modo da monitorare in tempo reale lo stato di salute dell'attuatore e prevedere eventuali guasti attraverso algoritmi di manutenzione predittiva. La capacità di raccogliere dati operativi in tempo reale, infatti, consente di implementare strategie di *Condition-Based Maintenance* (CBM), minimizzando i tempi di fermo e migliorando la sicurezza e l'affidabilità del sistema.

Nonostante presentino numerosi vantaggi, gli attuatori elettromeccanici hanno a che fare anche con alcune sfide tecnologiche. Le principali sono la gestione delle sollecitazioni elettromeccaniche e la dissipazione del calore generato dal motore elettrico e dai componenti elettronici. I motori *brushless*, infatti, pur essendo molto efficienti generano comunque del calore che deve essere dissipato per evitare il surriscaldamento del componente. Inoltre, essendo le viti a ricircolo di sfere dei riduttori molto complessi meccanicamente, bisogna imporre dei requisiti stringenti sulla progettazione e sulla scelta dei materiali. L'impiego di materiali avanzati, come leghe leggere ad alta resistenza risulta necessario per migliorare le prestazioni e la durata della vita operativa di questa tipologia di attuatori.

Gli EMA rappresentano, dunque, una soluzione innovativa nell'ambito aerospaziale, con svariati vantaggi in termini di efficienza, affidabilità e controllo. Con l'avanzare dell'evoluzione di tecnologie legate allo sviluppo di motori elettrici e riduttori, le prestazioni di questi attuatori miglioreranno ulteriormente, favorendo una transazione verso sistemi sempre più elettrificati ed intelligenti.

In questa prima sezione sono stati introdotti brevemente gli attuatori elettromeccanici e i loro vantaggi, mentre nel paragrafo successivo verrà riportato il modello EMA utilizzato per ottenere i risultati di questo lavoro, con una sua dettagliata descrizione .

3.2 Modello EMA proposto

Nello svolgere il lavoro, che verrà presentato nel capitolo successivo, è stato utilizzato un modello Simulink di un attuatore elettromeccanico basato su quello proposto da Schettini F., Denti E., Di Rito G. e Galatolo R., nel documento “Simulation of an All-Electric Flight Control System for the Evaluation of Power Consumption” [1]. Il modello proposto in questo studio è stato sviluppato facendo riferimento ad un sistema di controllo di volo (FCS) composto dalle seguenti superfici:

- Comandi di volo primari:
 - Alettoni e spoiler tradizionali per il controllo del rollio;
 - Equilibratori per il controllo del beccheggio;
 - Timone per il controllo dell’imbardata;
- Comandi di volo secondari:
 - Flap interni ed esterni;

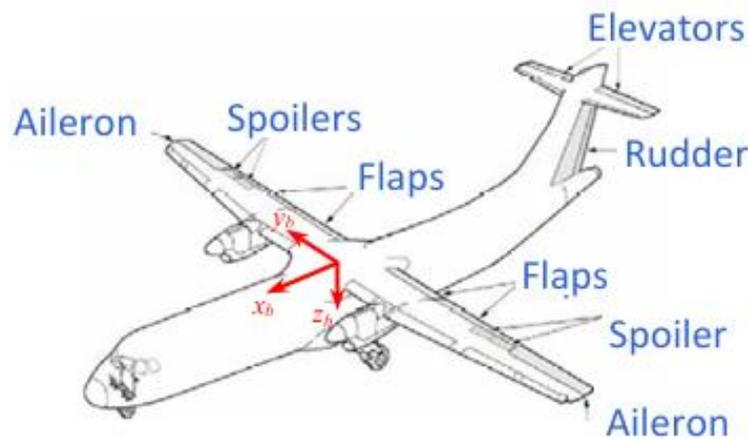


Figura 13 : Layout del FCS [1]

Il modello proposto in questo studio è stato sviluppato utilizzando per ogni superficie di controllo del *Flight Control System* lo schema Matlab-Simulink rappresentato in Figura 14.

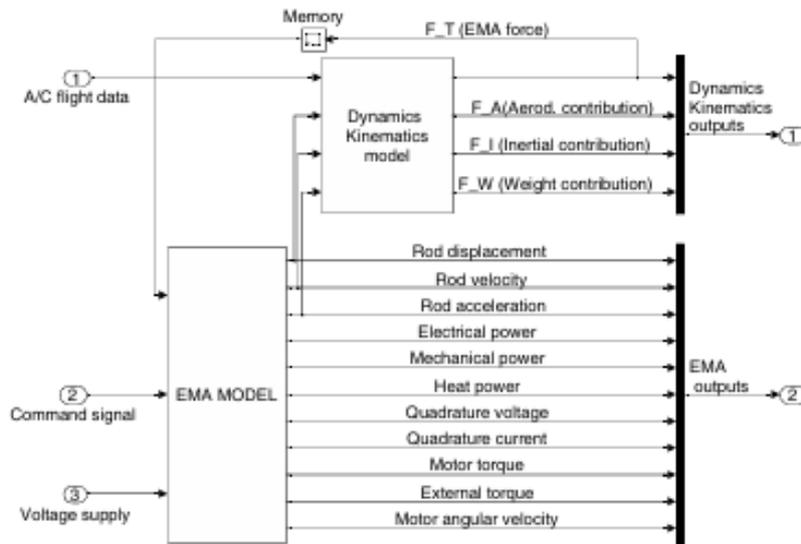


Figura 14 : Modello Matlab-Simulink proposto dallo studio [1]

Lo schema raffigurato è composto da due modelli principali, un modello EMA e un modello dinamico e cinematico per il controllo delle superfici. Essi sono strettamente collegati tra loro, poiché il modello EMA valuta la dinamica dell'asta dell'attuatore, la cui accelerazione, velocità e posizione sono necessarie al modello dinamico e cinematico per fornire la forza agente sul modello EMA stesso. Tale interazione causa un loop algebrico all'interno del modello Simulink, la quale è stata risolta applicando un ritardo nello step di integrazione sul segnale della forza.

3.3 Modello EMA utilizzato

Lo schema precedentemente presentato è stato modificato e semplificato per soddisfare le necessità richieste all'interno del nostro studio. La prima grande modifica è stata l'eliminazione del modello dinamico e cinematico, in quanto le grandezze di nostro interesse venivano fornite solamente dal modello EMA e per sopperire a questa mancanza, la forza fornita a questo modello è stata imposta ad un valore pari a costante. Inoltre, sono state apportate ulteriori modifiche all'interno del modello EMA stesso, che verranno analizzate nel dettaglio con la descrizione del modello.

Nello sviluppo del modello EMA sono state fatte tre assunzioni, come riportate nello studio: il motore elettrico modellato è un motore *brushless* trifase e la prima assunzione riguarda la sua dinamica, la quale è descritta nel riferimento $dq0$, senza l'uso delle trasformate di *Park* e trascurando l'asse omopolare; la seconda assunzione prevede che la potenza elettronica è ideale, senza nessuna caduta di tensione su un interruttore chiuso, nessuna corrente attraverso un interruttore aperto ed un perfetto inseguimento della tensione; la terza ed ultima assunzione prevede, invece, che la trasmissione meccanica sia considerata perfettamente rigida.

3.3.1 Modello matematico

Il modello matematico, su cui si basa il modello fisico, è composto da diverse equazioni che descrivono la componente elettrica e la componente meccanica [1].

Le equazioni elettriche lungo l'asse diretto e di quadratura del motore vengono riportate di seguito:

$$V_d = R i_d + L_d \frac{di_d}{dt} - L_q p_m i_q \omega_m \quad (\text{Eq. 1})$$

$$V_q = R i_q + L_q \frac{di_q}{dt} + L_d p_m i_d \omega_m + K_b \omega_m \quad (\text{Eq. 2})$$

I termini presenti all'interno di queste prime due equazioni rappresentano la tensione diretta (V_d), la tensione di quadratura (V_q), la corrente diretta (i_d) e la corrente di quadratura (i_q), che rappresentano rispettivamente le tensioni e le correnti dirette lungo l'asse diretto e l'asse di quadratura. La corrente diretta e la corrente di quadratura sono le due componenti di corrente di un motore *brushless* trifase. La corrente diretta è allineata con il flusso magnetico del rotore e, infatti, è la principale responsabile della produzione del flusso magnetico stesso. La corrente di quadratura, invece, è perpendicolare al flusso magnetico del rotore e svolge il compito principale di produrre coppia all'interno del motore.

Inoltre, nelle equazioni sopra riportate, sono presenti la resistenza del motore R , l'induttanza diretta L_d e l'induttanza di quadratura L_q , il numero di coppie di poli magnetici p_m , la velocità angolare dell'albero motore ω_m ed il coefficiente di forza controlettromotrice del motore K_b .

Le equazioni meccaniche su cui si basa il modello sono solamente due, l'equazione della coppia motore e l'equazione del moto riferita all'albero motore, utilizzata per descrivere la dinamica del sistema.

$$T_m = [(L_d - L_q)p_m i_d + K_b] i_q \quad (\text{Eq. 3})$$

$$J_{tot} \omega'_m = T_m - T_{ext} - B_{tot} \omega_m - T_{f\ tot} \text{sgn}(\omega_m) \quad (\text{Eq. 4})$$

I termini presenti nell'equazione 4 rappresentano la coppia motore T_m , la coppia esterna T_{ext} , il coefficiente di smorzamento viscoso B_{tot} , che descrive la resistenza al moto dovuta alle forze viscosi e il termine $T_{f\ tot}$, che tiene in conto degli effetti di attrito radente. Il termine ω'_m rappresenta, invece, l'accelerazione angolare dell'albero motore e viene moltiplicata per l'inerzia totale del sistema J_{tot} . Quest termine e la coppia esterna vengono ricavati dalle seguenti equazioni:

$$J_{tot} = J_m + \frac{J_g}{\tau_g^2} \quad (\text{Eq. 5})$$

$$T_{ext} = \frac{l_s}{\tau_g} F_{ext} \quad (\text{Eq. 6})$$

Nell'equazione 5 sono presenti i termini rappresentati l'inerzia dell'albero motore J_m e l'inerzia degli ingranaggi J_g , la quale viene divisa per il quadrato del rapporto di trasmissione del riduttore τ_g^2 . Nell'equazione 6, invece, è presente il rapporto tra il passo della vite l_s e il rapporto di trasmissione τ_g , il quale viene moltiplicato per la forza esterna F_{ext} , che viene fornita in input.

Le equazioni di controllo del modello EMA sono quattro e vengono utilizzate tramite quattro loop proporzionali per posizione, velocità del motore e corrente diretta e di quadratura.

$$V_d = -K_i i_d \quad (\text{Eq. 7})$$

$$V_q = K_i (i_{qi} - i_q) \quad (\text{Eq. 8})$$

$$i_{qi} = K_\omega (\omega_{mi} - \omega_m) \quad (\text{Eq. 9})$$

$$\omega_{mi} = K_x (x_{ai} - x_a) \quad (\text{Eq. 10})$$

I termini K_i , K_ω e K_x sono i guadagni di controllo rispettivamente dei loop di corrente, velocità angolare e posizione. Nell'equazione 10 è presente il termine x_a , che rappresenta la posizione dell'attuatore che andrebbe confrontata con la posizione comandata dal pilota per verificare il corretto funzionamento della superficie di controllo. Un'ulteriore equazione di cui si può tenere in conto è, quindi, la velocità lineare di movimento dell'attuatore:

$$x'_a = \frac{l_s}{\tau_g} \omega_m \quad (\text{Eq. 11})$$

All'interno dei loop di controllo, come si vedrà successivamente, i valori di tensione e corrente verranno saturati, in quanto le richieste di questi sono limitate.

Infine, viene riportata l'equazione legata al bilancio di potenza del modello:

$$P_e^{in} = P_e^{stored} + P_e^{loss} + P_m^{stored} + P_m^{loss} + P_m^{out} \quad (\text{Eq. 12})$$

L'equazione 12 prevede che la potenza assorbita dal motore sia pari alla somma della potenza totale persa, della potenza totale immagazzinata e della potenza meccanica in

uscita. I termini riportati rappresentano, infatti, la potenza elettrica in ingresso P_e^{in} , la potenza elettrica immagazzinata P_e^{stored} , la quale fa riferimento solamente alle induttanze in quanto non vengono considerati i capacitori, la potenza meccanica immagazzinata P_m^{stored} , la potenza elettrica persa nel circuito P_e^{loss} , la potenza meccanica persa P_m^{loss} e, infine, la potenza meccanica in uscita P_m^{out} .

Tutte queste potenze si ricavano tramite le seguenti equazioni matematiche:

$$P_e^{in} = V_d i_d + V_q i_q \quad (\text{Eq. 13})$$

$$P_e^{stored} = L_d \left(\frac{di_d}{dt} \right) i_d + L_q \left(\frac{di_q}{dt} \right) i_q \quad (\text{Eq. 14})$$

$$P_e^{loss} = R(i_d^2 + i_q^2) \quad (\text{Eq. 15})$$

$$P_m^{stored} = J_{tot} \omega_m' \omega_m \quad (\text{Eq. 16})$$

$$P_m^{loss} = B_{tot} \omega_m^2 + T_{f\ tot} \text{sgn}(\omega_m) \omega_m \quad (\text{Eq. 17})$$

$$P_m^{out} = T_{ext} \omega_m \quad (\text{Eq. 18})$$

3.3.2 Modello Matlab-Simulink

In questa sezione verrà presentato e descritto nel dettaglio il modello fisico creato grazie al software Matlab-Simulink, il quale si basa sul modello fornito dallo studio di riferimento, ma al quale sono state apportate le dovute modifiche del caso.

Si è voluto simulare un comando in ingresso fornito all'attuatore elettromeccanico collegato alla superficie di controllo riferita agli equilibratori. Il segnale fornito in ingresso viene generato tramite un comando a gradino utilizzando il blocco Simulink "Pulse Generator", il quale genera impulsi a onda quadra ad intervalli regolari. I parametri forniti al blocco sono i seguenti:

- Tipo di impulso: viene scelta l'opzione "Time-based", in quanto è consigliata per l'uso con un solutore a passo variabile
- Tempo (t): viene utilizzato il tempo di simulazione pari a $t = 4$ s
- Ampiezza: 0.04
- Periodo: $T = 2$ s
- Larghezza dell'impulso (larghezza del gradino): 50 % del periodo, ossia 1 s
- Ritardo di fase (tempo dopo cui parte l'impulso): 0.25 s

Al blocco "Pulse Generator" viene sommata una costante pari a 0.02, in modo che la posizione dell'attuatore, data dal comando, vari tra 0.02 e 0.06. Questi valori sono stati scelti in accordo con quelli presentati dallo studio di riferimento. Il segnale ottenuto corrisponde

alla posizione comandata all'attuatore dell'equilibratore e viene fornito come variabile di ingresso al sottosistema di riferimento nominato EMA.

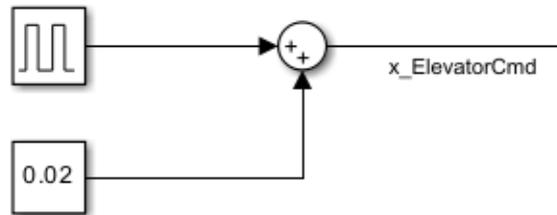


Figura 15 : Segnale di comando di ingresso

All'interno del modello sono presenti altri blocchi per la generazione del comando di ingresso, i quali nel caso in analisi non sono stati utilizzati, ma che vengono riportati per completezza. Viene utilizzato un blocco "Sine Wave", per generare un'onda sinusoidale, il quale permette di scegliere la tipologia di sinusoide che si vuole utilizzare tra quella "Time-based" o "Sample-Based", che viene consigliata per lunghi tempi di simulazione. Il secondo blocco è, invece, una costante e tra questi è interposto un "Clock", che fornisce il tempo corrente della simulazione. I tre blocchi sono collegati ad uno "Switch", per cui se l'input 1 (ossia il blocco "Sine Wave") soddisfa il criterio presente all'interno dello "Switch" stesso, allora il segnale di comando corrisponderà ad esso, altrimenti passerà l'input 3, pari al blocco "Constant".

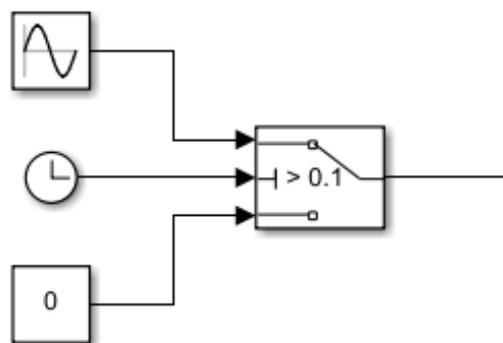


Figura 16 : Seconda tipologia di comando di ingresso

La scelta tra le due tipologie di comando può essere effettuata tramite l'utilizzo di un "Manual Switch".

Una volta generato il segnale di comando di ingresso, questo viene fornito, come detto, in input al sottosistema di riferimento EMA. Come altra variabile di input è presente anche la forza di carico esterna. Tale valore si ricava in maniera molto elementare, moltiplicando il segnale di posizione per un blocco di guadagno “Gain” di valore pari a $K = 20000$. In questo modo, si ottiene che la forza esterna F_{ext} , vari tra un valore di 400 N e 1200 N. La costante K è stata assunta per ottenere dei carichi applicati plausibili.

Come variabili di output dal sottosistema EMA, si ottengono la posizione attuale dell’attuatore, che va confrontata con la posizione comandata, le correnti e le tensioni dirette e di quadratura. Come si vedrà successivamente le variabili di uscita saranno molte di più, ma a livello esterno del modello vengono riportate solo queste.

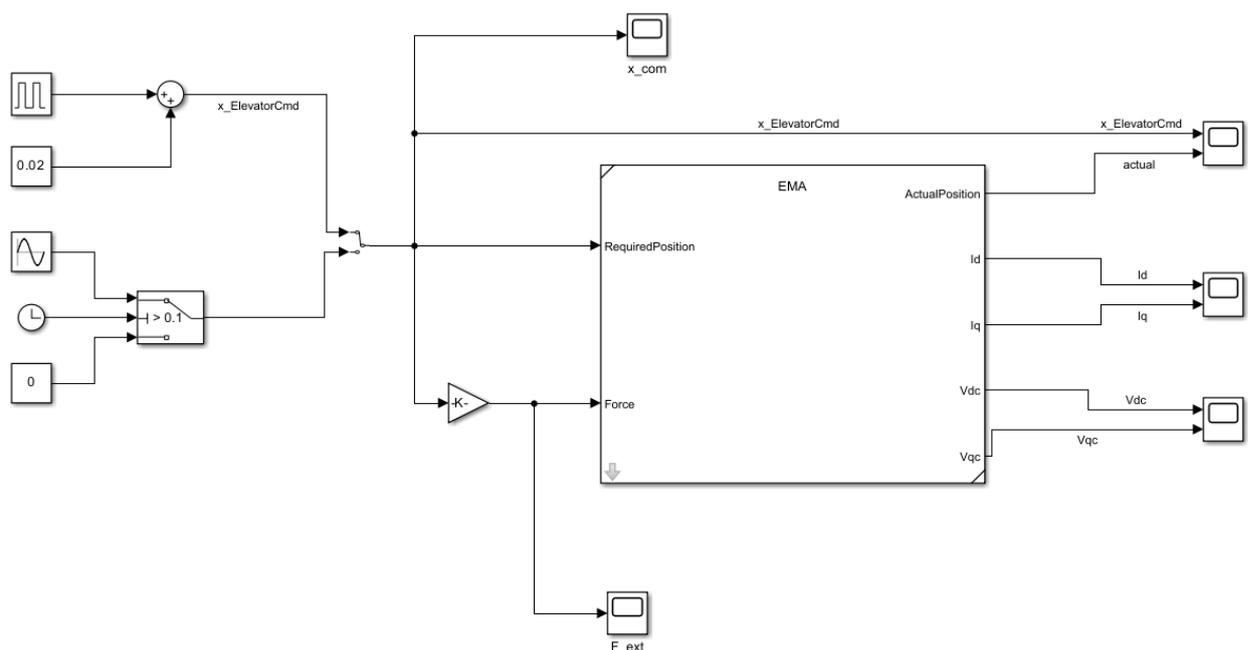


Figura 17: Modello Matlab-Simulink

Viene ora descritto il sottosistema di riferimento EMA. Esso è composto da ulteriori due sottosistemi: il *Plant* e il *Controller*. Innanzitutto, è necessario fornire i valori dei parametri costanti all’interno del modello, che sono stati scelti facendo riferimento al documento di studio.

I parametri da fornire al *Plant* sono i seguenti:

Riduttore e Vite a ricircolo di sfere	
<i>Off Position</i>	0.6
<i>Posizione iniziale</i>	0.02
<i>Passo della vite l_s</i>	$8 \cdot 10^{-4}$ [m / rad]
<i>Rapporto di trasmissione del riduttore τ_g</i>	1.1 [-]
<i>Inerzia degli ingranaggi J_g</i>	$4 \cdot 10^{-4}$ [kg*m ²]

Motore elettrico	
<i>Parametro che tiene conto degli effetti dell'attrito radente $T_{f\ tot}$</i>	0.01 [N*m]
<i>Coefficiente di smorzamento viscoso complessivo B_{tot}</i>	$4 \cdot 10^{-3}$ [N*m*s / rad]
<i>Numero di coppie di poli magnetici p_m</i>	1
<i>Coefficiente di forza controelettromotrice del motore K_b</i>	0.4 [N*m / A]
<i>Inerzia dell'albero motore J_m</i>	$4 \cdot 10^{-4}$ [kg*m ²]

I parametri da fornire al Controller sono invece:

<i>Guadagno di controllo per la tensione diretta</i>	-1
<i>Guadagno di controllo per la velocità angolare dell'albero motore</i>	$5 \cdot 10^5$
<i>Guadagno di controllo per la corrente di quadratura</i>	0.2
<i>Guadagno di controllo per la tensione di quadratura</i>	1

Il primo sottosistema presente all'interno dell'EMA è il *Controller*. Questo riceve in ingresso la posizione richiesta fornita in input all'EMA, la posizione attuale, la velocità angolare dell'albero motore, la corrente diretta e la corrente di quadratura che arrivano come uscite del *Plant*. All'interno del *Controller* sono riportate le equazioni di controllo (Eq.7), (Eq.8), (Eq.9) e (Eq.10), le quali vengono rappresentate tramite l'utilizzo di quattro loop proporzionali. I quattro loop di controllo vengono utilizzati per ricavare la tensione diretta, che viene ottenuta da un loop singolo, e la tensione di quadratura, che si ricava, invece, partendo dal loop della velocità angolare, passando poi al loop della corrente di quadratura

e, infine, al loop della tensione di quadratura stessa. Ognuno di questi subisce, inoltre, una saturazione tramite l'utilizzo di un blocco "Switch". Il loop collegato alla corrente diretta prevede una saturazione tra i -270 V e i 270 V, come anche il loop per la tensione di quadratura, con tale valore che viene scelto come tensione massima. Il loop della corrente di quadratura viene, invece, saturato tra i valori di -30 A e 30 A, valore massimo di corrente ammissibili, mentre il loop della velocità angolare subisce una saturazione compresa tra i -800 rad/s e 800 rad/s. Tali valori massimi sono stati scelti basandosi sui termini massimi di tensione, corrente e velocità angolare proposti nel documento di riferimento.

Come valori di output del *Controller* si ottengono quindi i segnali di tensione di quadratura e di tensione diretta.

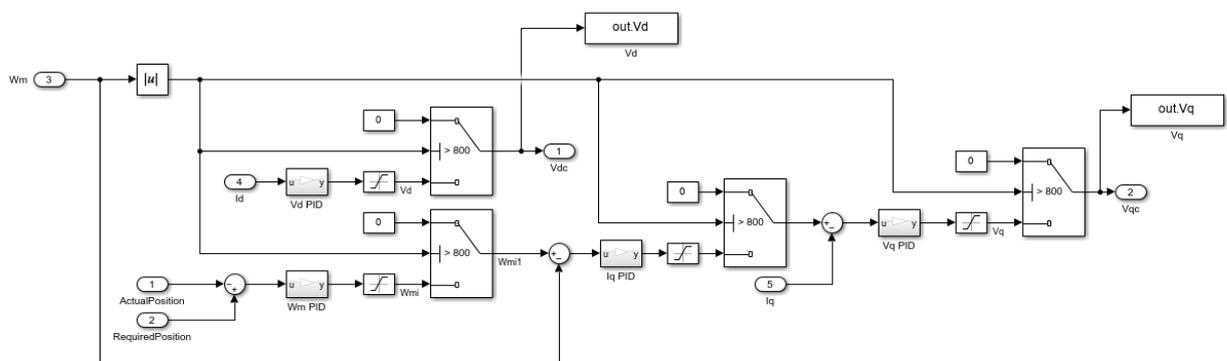


Figura 18 : Schema del Controller

Ricavati i valori di tensione è possibile fornirli come input, insieme alla forza esterna al secondo sottosistema, ossia il *Plant*. Questo risulta molto semplice e prevede l'implementazione dell'equazioni matematiche riportate in precedenza, in modo da potersi ricavare le variabili di interesse. Alcune di queste saranno le variabili che servono come input al *Controller*, rendendo tutto il modello interconnesso.

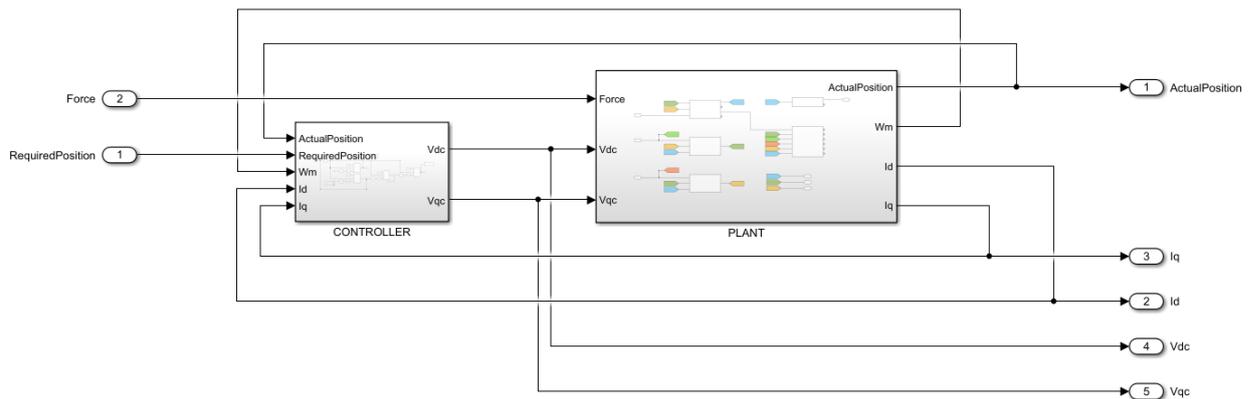


Figura 19 : Schema del sottosistema di riferimento EMA

All'interno del *Plant* sono presenti cinque sottosistemi, ognuno dei quali viene utilizzato per ricavare le variabili di nostro interesse. Questi sono divisi in:

- Blocco di valutazione della velocità angolare dell'albero motore ("*Evaluate_Wm*")
- Blocco di valutazione della corrente diretta ("*Evaluate_Id*")
- Blocco di valutazione della corrente di quadratura ("*Evaluate_Iq*")
- Blocco di valutazione della posizione attuale dell'attuatore ("*Gearbox & Ballscrew*")
- Blocco di valutazione delle potenze del sistema ("*Powers*")

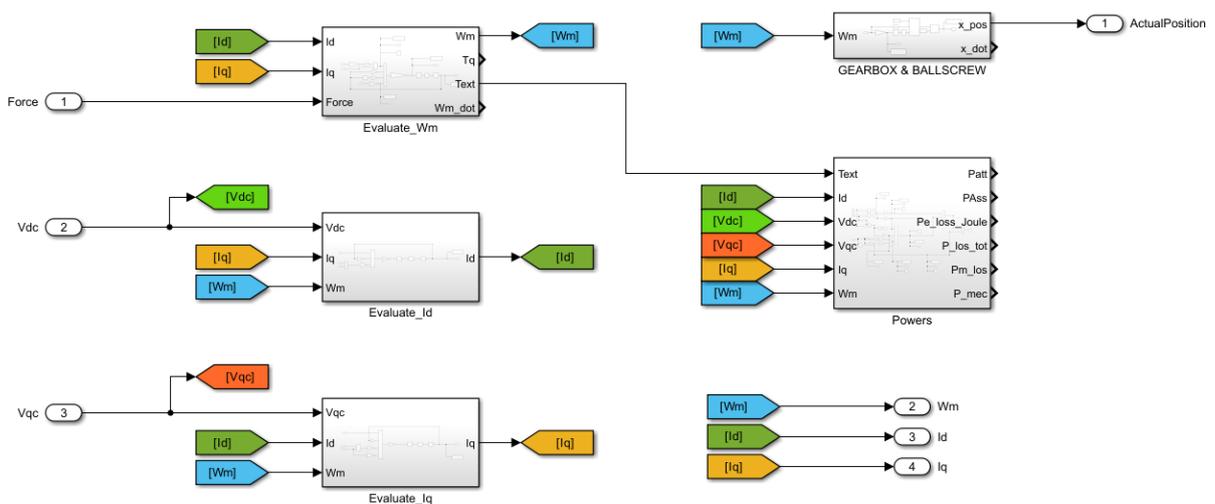


Figura 20 : Schema del Plant

Valutazione della velocità angolare dell'albero motore ω_m :

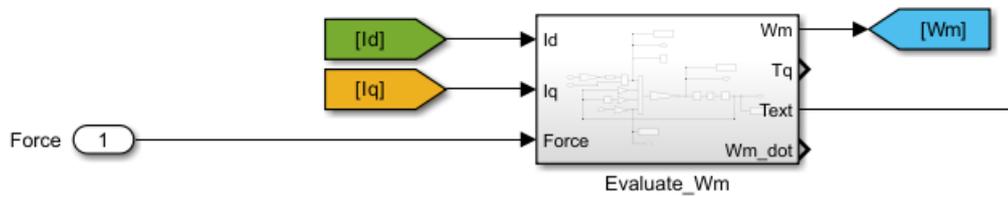


Figura 21 : Blocco di valutazione della velocità angolare ω_m

Il primo sottosistema presente all'interno del *Plant*, in alto a sinistra da Figura 20, riguarda il blocco di valutazione della velocità angolare dell'albero motore. Esso riceve come variabili di ingresso la forza esterna F_{ext} nota, la corrente diretta i_d e la corrente di quadratura i_q , le quali vengono ricavate nei blocchi successivi. Come variabili di output il blocco genera la velocità angolare dell'albero motore ω_m e la coppia esterna T_{ext} , la quale servirà come input per il blocco "Powers". All'interno del blocco vengono implementate le equazioni (Eq.3), (Eq.4) e (Eq.6). Dall'equazione 3 è possibile ricavarsi il parametro di coppia del motore, al quale vanno sommati gli altri parametri per ottenere l'equazione 4. L'equazione 6 viene utilizzata per ricavarsi il parametro di coppia esterna, il quale andrà anch'esso sommato all'interno dell'equazione 4. L'obiettivo è ottenere la velocità angolare ω_m , per cui in (Eq.4) va isolato il termine ω_m' , rappresentante l'accelerazione angolare dell'albero motore. Quest'ultimo viene integrato tramite l'utilizzo del blocco "Discrete-Time Integrator" e saturato con un blocco "Saturation". A questo punto, è possibile ricavarsi la velocità angolare ω_m , la quale, insieme alla coppia esterna T_{ext} , viene portata al di fuori del sistema tramite un "Outport".

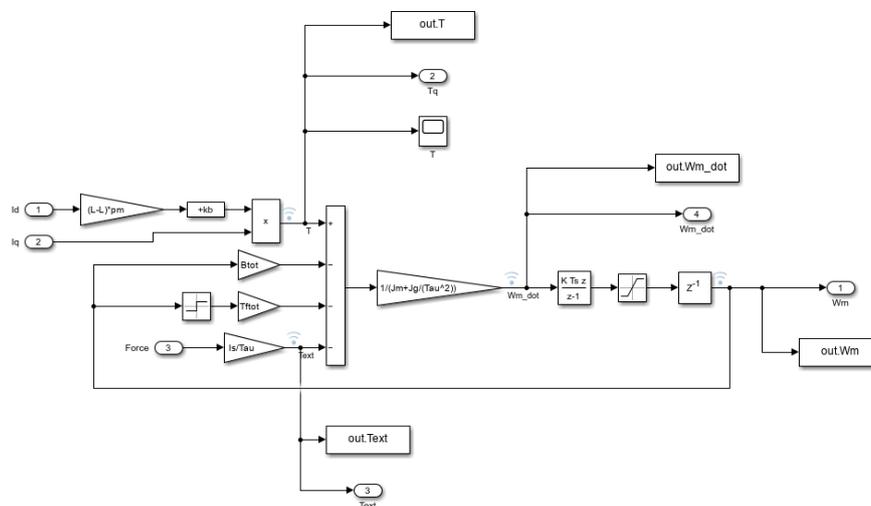


Figura 22 : Schema valutazione ω_m

Valutazione della corrente diretta i_d :

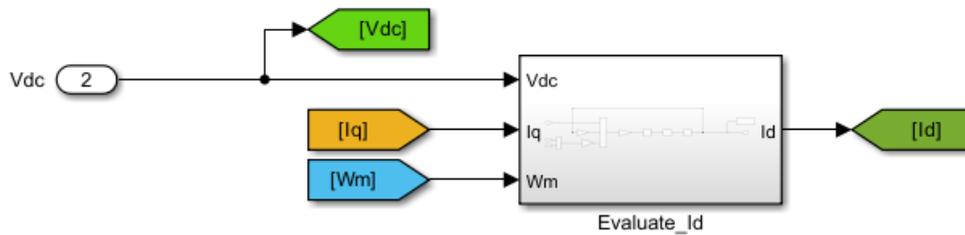


Figura 23 : Blocco di valutazione della corrente diretta i_d

Il secondo blocco sulla sinistra del *Plant*, Figura 20, rappresenta il sottosistema di valutazione della corrente diretta i_d . I parametri che questo riceve in ingresso sono la tensione diretta V_d nota, ricavata all'interno del *Controller*, la corrente di quadratura i_q , la quale viene valutata nel blocco successivo, e la velocità angolare ω_m ricavata nel blocco precedente. L'unica variabile in uscita che il blocco genera è proprio la corrente diretta i_d . All'interno del blocco viene implementata l'equazione matematica (Eq.1), isolando il termine della derivata temporale della corrente diretta. Successivamente viene integrato e saturato con i blocchi "*Discrete-Time Integrator*" e "*Saturation*", in modo da ottenere il parametro i_d . Questo viene portato fuori dal blocco tramite un "*Outport*".

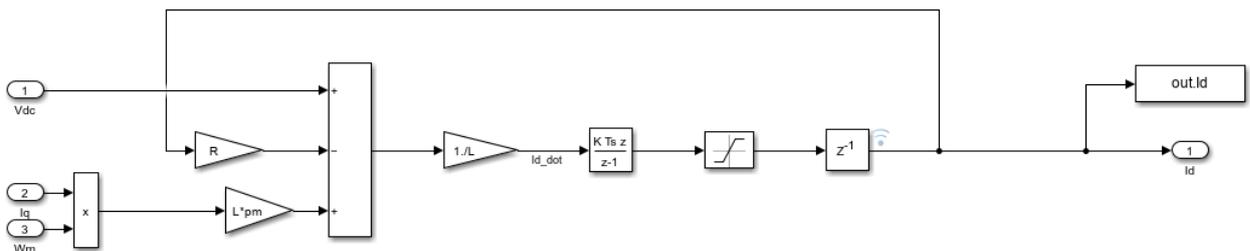


Figura 24 : Schema di valutazione della corrente diretta i_d

Valutazione della corrente di quadratura i_q :

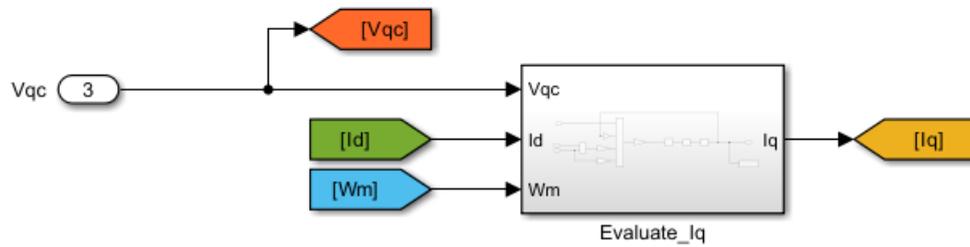


Figura 25 : Blocco di valutazione della corrente di quadratura i_q

Il terzo blocco sulla sinistra in basso della Figura 20 è relativo alla valutazione della corrente di quadratura i_q . I parametri forniti in ingresso al sottosistema sono la tensione di quadratura V_q , ottenuta precedentemente dal *Controller*, la corrente diretta i_d e la velocità angolare dell'albero motore ω_m , valutate nei due blocchi sopra. L'unica variabile di uscita corrisponde proprio alla corrente di quadratura i_q . All'interno del sottosistema viene implementata l'equazione matematica (Eq.2), in cui viene isolata la derivata temporale della corrente i_q . Questa viene integrata e saturata con i blocchi "Discrete-Time Integrator" e "Saturation", ottenendo la variabile di interesse i_q , che viene portata all'esterno dello schema tramite un "Outport".

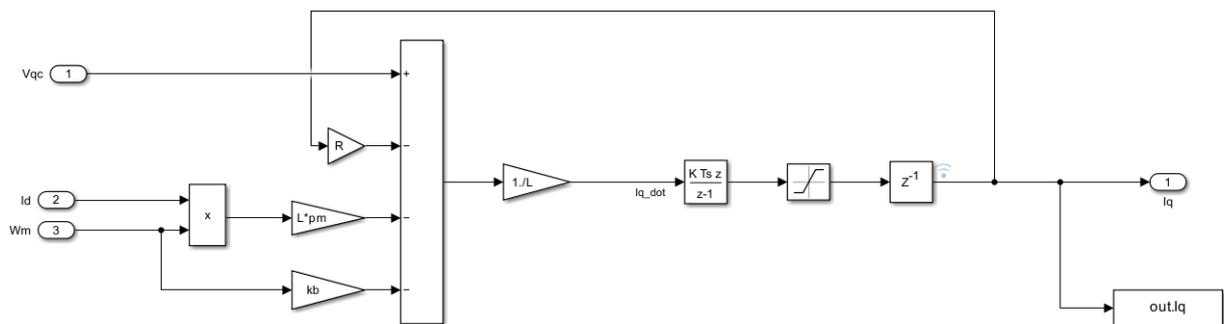


Figura 26 : Schema di valutazione della corrente di quadratura i_q

Valutazione della posizione attuale dell'attuatore x_{pos} :



Figura 27: Blocco di valutazione della posizione attuale dell'attuatore x_{pos}

Il blocco situato in alto a destra nel *Plant*, tiene in conto delle componenti meccaniche del sistema, ossia il riduttore e la vite a ricircolo di sfere. Come variabile di ingresso viene fornita solamente la velocità angolare dell'albero motore ω_m , ricavata inizialmente, mentre come variabile di uscita si ottiene la posizione attuale dell'attuatore, che andrà confrontata con la posizione comandata all'esterno del sottosistema di riferimento EMA. All'interno del blocco viene implementata l'equazione (Eq. 11), che tiene in considerazione il passo della vite l_s e il rapporto di trasmissione del riduttore τ_g , in modo da ricavare la velocità lineare di movimento dell'attuatore x_{pos}' . Questa viene integrata e saturata tramite l'utilizzo dei blocchi "Discrete-Time Integrator" e "Saturation", ma, prima di ottenere il valore finale della variabile, questa deve ancora passare attraverso uno "Switch". Questo serve a valutare che la velocità angolare dell'albero motore rientri nei limiti di velocità concessi per il corretto funzionamento dell'attuatore. Se la velocità angolare rispetta i valori massimi ammissibili allora si ottiene la posizione attuale dell'attuatore x_{pos} , altrimenti, attraverso lo "Switch", passa l'input 1 corrispondente al valore di *Off Position*. In ogni caso la variabile di uscita x_{pos} viene portata fuori dal sottosistema tramite l'utilizzo di un "Outport".

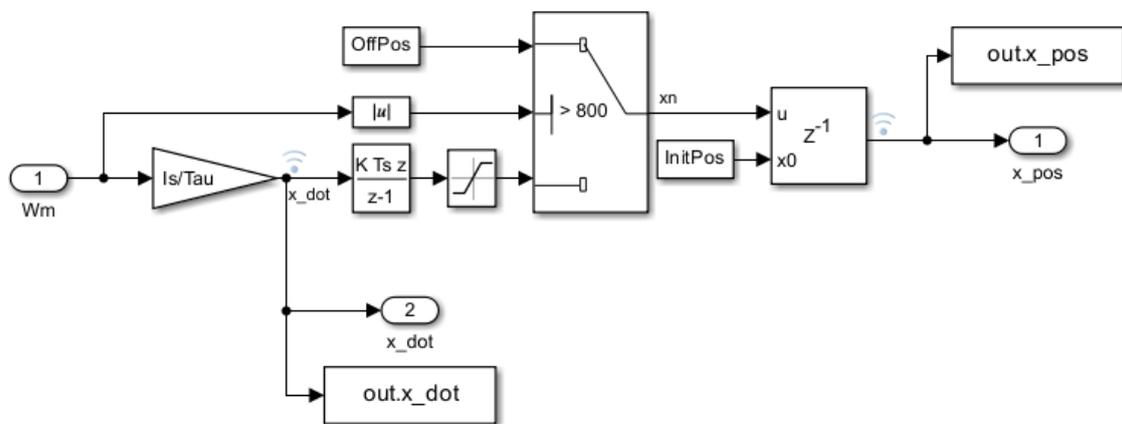


Figura 28: Schema di valutazione della posizione attuale dell'attuatore x_{pos}

Valutazione delle potenze elettriche e meccaniche del sistema:

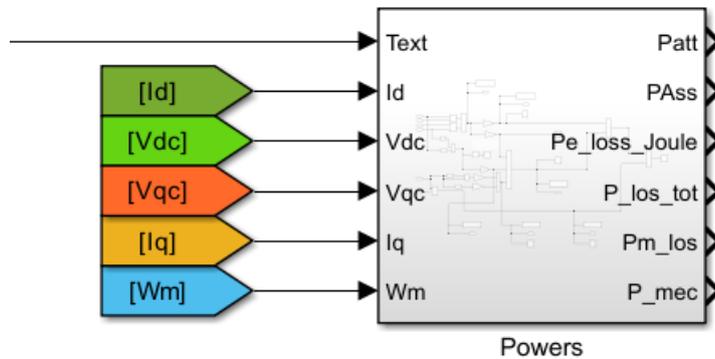


Figura 29 : Blocco di valutazione delle potenze del sistema

L'ultimo blocco presente all'interno del *Plant* riguarda la valutazione delle potenze elettriche e meccaniche presenti all'interno del sistema attuatore. Le variabili che vengono fornite in ingresso sono diverse in questo caso: la coppia esterna T_{ext} , la corrente diretta i_d , la tensione diretta V_d , la tensione di quadratura V_q , la corrente di quadratura i_q e, infine, la velocità angolare dell'albero motore ω_m . Tutte queste variabili a questo punto sono state valutate nei passaggi precedenti. In uscita dal blocco non sono presenti alcune variabili necessarie al corretto funzionamento dell'intero modello, ma sono presenti solamente le varie potenze necessarie per i futuri passaggi del lavoro. Lo schema all'interno risulta leggermente più complesso rispetto a quelli precedenti, poiché vede l'implementazione di un maggior numero di equazioni.

Innanzitutto, viene calcolata la Potenza Attiva P_{att} , tramite l'equazione matematica (Eq.13). Essa rappresenta, infatti, la potenza effettivamente assorbita ed utilizzata dal sistema per compiere lavoro utile, in quanto rappresenta la potenza elettrica che viene convertita in energia meccanica per muovere le superfici di controllo [14]. Dalla Potenza Attiva è possibile ottenere anche la Potenza Apparente P_{app} . Solitamente avviene il contrario, in quanto la formula che lega le due potenze è:

$$P_{att} = P_{app} * \cos(30^\circ)$$

In questo caso, avendo già la Potenza Attiva, possiamo utilizzare la formula inversa per ricavarci la Potenza Apparente come:

$$P_{app} = \frac{P_{att}}{\frac{\sqrt{3}}{2}}$$

Questa rappresenta la potenza attiva che otterremo annullando lo sfasamento tra tensione e corrente.

La Potenza Attiva viene, inoltre, moltiplicata per un fattore pari a 0.18, ottenendo la Potenza Immagazzinata del sistema $P_{el_mec_stored}$. Questo fattore tiene, infatti, conto delle potenze immagazzinate dal motore elettrico e dalla parte meccanica, che nel modello matematico sarebbero rappresentate dalle equazioni (Eq. 14) e (Eq. 16), ma che qui non vengono riportate e racchiuse in un unico fattore.

Le altre potenze che si ricavano all'interno dello schema sono la Potenza Elettrica persa per effetto Joule $P_{el_loss_Joule}$, tramite l'implementazione dell'equazione (Eq. 15), la Potenza Meccanica persa P_{m_loss} , tramite l'equazione (Eq. 17), e la Potenza Meccanica generata in uscita P_{mecc} , implementando l'equazione (Eq. 18).

Infine, viene calcolata la Potenza Totale persa P_{loss_tot} , la quale rappresenta la quantità di calore generato all'interno del sistema a causa delle perdite di potenza derivate dalla resistenza elettrica o dagli attriti meccanici. Essa, infatti, viene calcolata come la somma della Potenza Immagazzinata dal sistema, della Potenza Elettrica persa per effetto Joule e della Potenza Meccanica persa.

Per verificare il corretto calcolo di tutte le potenze considerate, si è effettuato un bilancio di potenza basandosi sull'equazione (Eq. 12), verificando che la differenza tra la Potenza Apparente e la Potenza Totale persa sia uguale alla Potenza Meccanica generata in uscita.

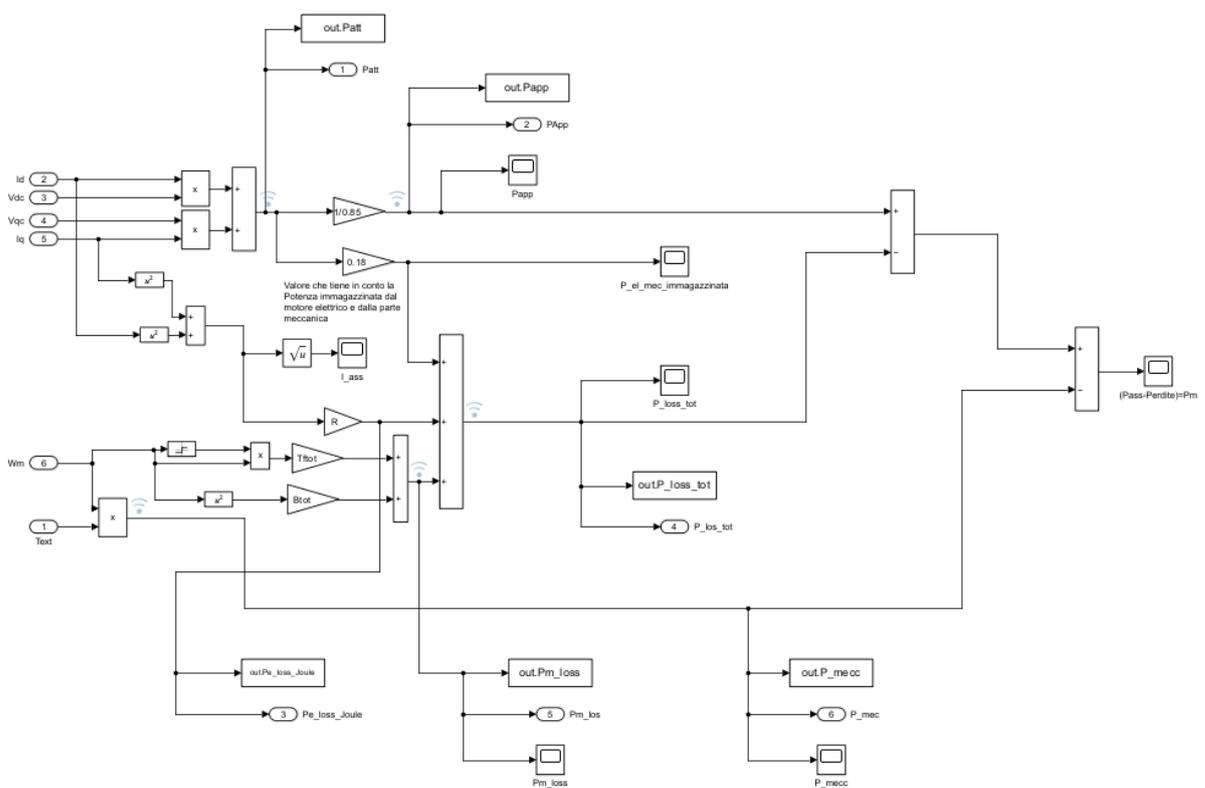


Figura 30 : Schema di valutazione delle Potenze

Il modello Matlab-Simulink sopra descritto risulta molto semplice, ma efficace per descrivere il funzionamento di un attuatore elettromeccanico. La parte di maggiore interesse è stata, infatti, posta sul motore elettrico, mentre la componente meccanica è stata approssimata al solo blocco “*Gearbox & Ballscrew*”, dove la velocità lineare del servomeccanismo è stata ottenuta con una trasformazione della velocità angolare del motore elettrico e la posizione attuale tramite un’integrazione del valore ricavato.

L’obiettivo non era tanto quello di ricreare un modello rappresentante un attuatore elettromeccanico il più accurato possibile, ma semplicemente avere uno strumento con cui potersi ricavare delle variabili di interesse per lo sviluppo successivo del lavoro.

Una volta che la costruzione del modello è stata effettuata si è potuto procedere allo sviluppo dell’algoritmo su cui si basa il lavoro di tesi.

4 SVILUPPO DELL'ALGORITMO PER LA MANUTENZIONE PREDITTIVA

In questo capitolo verrà analizzato nel dettaglio l'algoritmo sviluppato tramite il software Matlab per la stima della *Remaining Useful Life (RUL)* del modello Matlab-Simulink sopra descritto. Tramite questo algoritmo si è voluto definire un processo metodologico da compiere per arrivare alla stima della *RUL* di un componente tramite l'utilizzo di alcuni Toolbox presenti nel software, come il *Diagnostic Feature Designer* e l'*Health Indicator Designer*. Lo studio del degrado si è concentrato sulla parte del motore elettrico dell'attuatore elettromeccanico e, in particolare, sull'usura della resistenza e dell'induttanza, parametri chiave nel monitoraggio delle condizioni operative. Data la mancanza di dati reali e di modelli empirici consolidati che descrivessero l'andamento del degrado di questi parametri, si è proceduto alla generazione di dati sintetici basandosi su ipotesi fondate prese dalla letteratura esistente, come verrà illustrato successivamente.

Si procede ora con un'introduzione alla Manutenzione Predittiva e come questa può essere eseguita sfruttando gli strumenti del software Matlab.

4.1 Introduzione alla Manutenzione Predittiva

La Manutenzione Predittiva è un approccio alla manutenzione di componenti e sistemi tramite l'utilizzo di algoritmi predittivi, i quali utilizzano i dati forniti dai sensori e altre informazioni utili per rilevare anomalie, monitorare lo stato operativo dei componenti e stimare la vita utile residua (*RUL*). In questo modo, è possibile programmare la manutenzione nel momento corretto risparmiando tempi e costi [15].

Al giorno d'oggi le due tipologie di manutenzioni maggiormente praticate sono la Manutenzione Reattiva e la Manutenzione Preventiva. Il primo approccio prevede di effettuare una manutenzione solo dopo che il guasto sia avvenuto e per questo motivo non può essere applicato a certi settori industriali come quello aeronautico. Inoltre, i guasti e i tempi di fermo non pianificati possono risultare molto costosi, oltre che pericolosi per l'intera macchina.

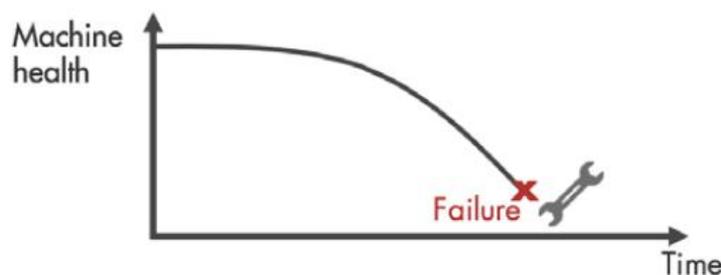


Figura 31 : Funzionamento Manutenzione Reattiva [16]

Il secondo approccio viene, quindi, maggiormente applicato rispetto al primo e prevede la programmazione della manutenzione ad intervalli regolari senza tenere in conto le effettive condizioni della macchina. Nonostante vengano ridotti i rischi di guasto rispetto al primo caso, la Manutenzione Preventiva risulta comunque molto costosa e prevede un aumento dei tempi di fermo macchina, oltre ad un aumento di pezzi di ricambio che devono essere utilizzati. Inoltre, i guasti imprevisti non vengono presi in considerazione, in quanto le ispezioni manutentive vengono effettuate periodicamente ad intervalli prestabiliti e non monitorano continuamente lo stato di salute in tempo reale.

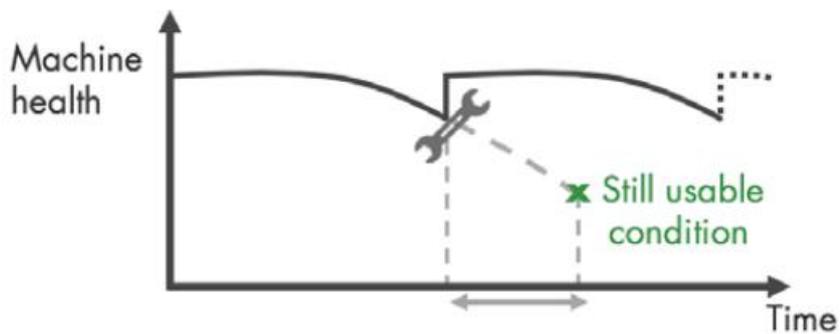


Figura 32 : Funzionamento Manutenzione Preventiva [16]

In questo contesto si inserisce la Manutenzione Predittiva, la quale è in grado di monitorare continuamente lo stato di salute corrente della macchina e di stimarne i futuri possibili guasti. Questo permette agli operatori di pianificare ed effettuare interventi manutentivi mirati e solo quando necessari. Ciò risulta molto vantaggioso, poiché vengono minimizzati i tempi di fermo macchina non programmati, vengono ridotti i costi operativi e vengono forniti degli avvertimenti in caso di guasti imprevisti.

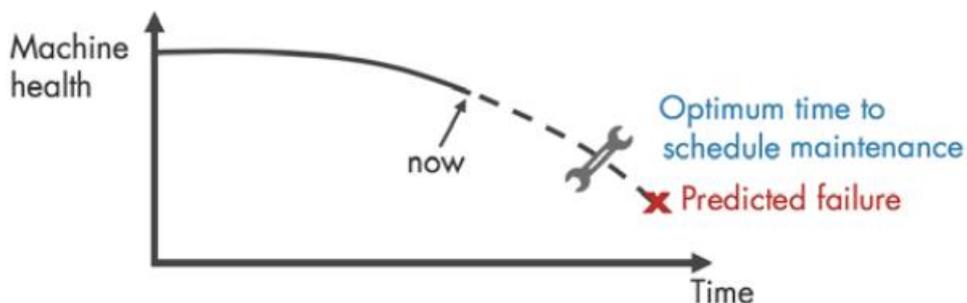


Figura 33 : Funzionamento Manutenzione Predittiva [16]

La Manutenzione Predittiva viene basata sullo sviluppo di un algoritmo, il quale analizza i dati dei sensori per individuare la presenza di anomalie, diagnosticare eventuali guasti alle apparecchiature e stimare la vita utile residua (*RUL*) del componente o sistema. Lo sviluppo di tale algoritmo deve necessariamente partire dall'acquisizione di dati grezzi che descrivano il funzionamento del sistema in un intervallo che va da una condizione di salute sana ad una condizione guasta. In base alla tipologia di dato che viene acquisito, questo va preprocessato in modo da portarlo in una forma in cui sia possibile estrarre i “*Condition Indicators*”, i quali sono delle caratteristiche del sistema che permettono di distinguere le condizioni di funzionamento sane da quelle guaste. Queste *features* risultano di fondamentale importanza, poiché verranno fornite al modello di machine learning utilizzato per addestrarlo a identificare le anomalie, a classificare i diversi tipi di guasto e a stimare la vita utile rimanente (*RUL*) del nostro sistema. Questo algoritmo può, successivamente, essere distribuito ed integrato nei sistemi IT/OT a cui vengono trasmessi i dati.

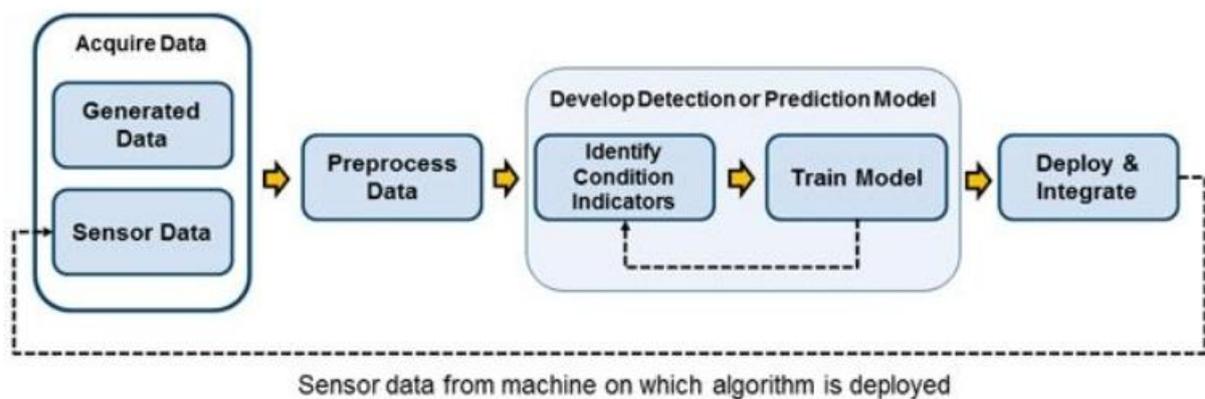


Figura 34 : Flusso di lavoro con cui sviluppare l'algoritmo per la Manutenzione Predittiva [15]

Acquisizione dei dati:

L'acquisizione dei dati è il primo passo fondamentale per lo sviluppo di un algoritmo di Manutenzione Predittiva. I dati raccolti devono rappresentare sia condizioni operative sane che condizioni di guasto, per questo è necessario valutare la macchina in diverse condizioni operative. L'algoritmo, infatti, risulta accurato solo se dispone di dati di addestramento robusti e che rappresentano i tipi di guasto che possono avvenire, ma spesso questi sono di difficile accesso e spesso risulta problematico ottenere il giusto tipo di dati con cui iniziare a costruire un algoritmo accurato. Una soluzione per questi imprevisti può essere quella di generare un modello matematico del nostro componente, tramite l'utilizzo di software come Simulink, e stimare i suoi parametri dai dati raccolti dai sensori. Successivamente, è possibile simulare il modello con stati di guasto differenti in diverse condizioni operative, in modo da generare dei dati di guasto. Questa tipologia di dati viene anche definita come dati sintetici e fungono da supplemento ai dati ricavati dai sensori. In questo modo, infatti, è possibile utilizzare una combinazione di dati sintetici e dati ricavati dai sensori per generare

dati di guasto in maniera sicura e vantaggiosa e per sviluppare l'algoritmo di Manutenzione Predittiva. Tali modelli basati sulla fisica del componente in analisi possono poi essere utilizzati nelle operazioni come *digital twins* ("gemelli digitali") per prevedere le prestazioni future.

Preprocessing dei dati:

Una volta che si sono ottenuti i dati, il passaggio successivo prevede il preprocessing di questi, per convertirli in una forma in cui i *Condition Indicators* possono essere estratti facilmente. Le tecniche di *preprocessing* prevedono la rimozione del rumore, degli *outlier* e dei valori mancanti. Inoltre, spesso queste tecniche vengono utilizzate per ottenere informazioni aggiuntive, che possono non apparire nella forma originale in cui sono forniti i dati. Per questo motivo vengono effettuate delle conversioni da dominio nel tempo a dominio in frequenza.

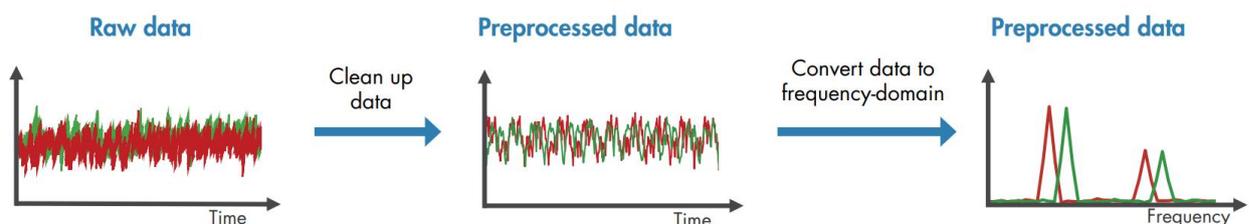


Figura 35 : Esempio di preprocessing dei dati [16]

Identificazione dei *Condition Indicators*:

Dopo aver raccolto tutti i dati necessari e averli preprocessati, si può procedere con l'identificazione dei *Condition Indicators* (*CI*) riducendo i dati ad un insieme di *features*. I *Condition Indicators* sono, infatti, nient'altro che delle *features* che indicano la differenza tra lo stato di salute sano e lo stato di salute guasto. Essi vengono estratti tramite una combinazione di tecniche statistiche, di elaborazione di segnali e di approcci basati sui modelli, utilizzando degli strumenti di progettazione come il *Diagnostic Feature Designer* presenti in software come Matlab. L'identificazione corretta dei *CI* è di fondamentale importanza per lo sviluppo di un algoritmo accurato, in quanto rilevano delle tendenze di degrado difficilmente identificabili.

In Matlab è possibile generare dei *Condition Indicators* sfruttando sia approcci basati sui segnali che sul modello. È possibile, infatti, catturare le dinamiche variabili nel tempo, spesso attraverso l'osservazione dell'analisi dei dati di vibrazione. Inoltre, per rilevare cambiamenti improvvisi all'interno dei dati raccolti, si possono calcolare delle *features*

basate su ricostruzioni dello spazio delle fasi, le quali monitorano le variazioni dello stato del sistema.

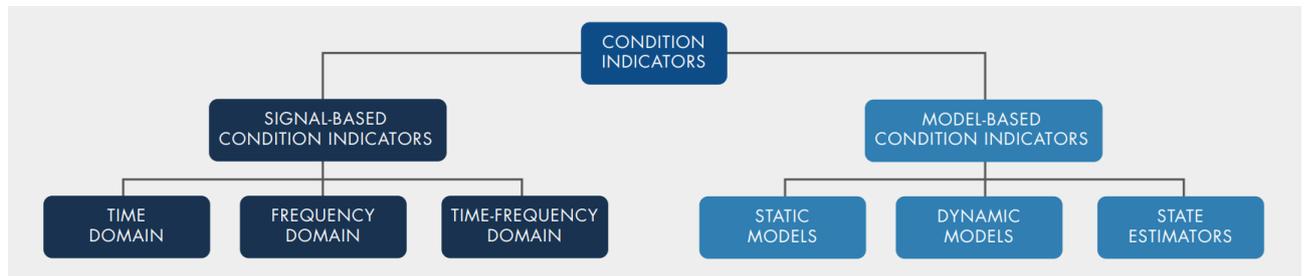


Figura 36 : Tipologie di Condition Indicators [16]

Addestramento del modello:

Il passo successivo all'identificazione dei *Condition Indicators* prevede l'addestramento dell'algoritmo predittivo. Fino a questo punto le *features* estratte dai dati a disposizione sono servite a identificare uno stato di salute sano e uno stato di salute guasto, ma non permettono di comprendere quale elemento del sistema ha bisogno di essere riparato o quanta vita utile operativa gli rimane. Proprio grazie all'addestramento degli algoritmi è possibile trasformare i dati provenienti dai sensori in decisioni di manutenzione, che possono essere distinte in tre categorie principali: il rilevamento delle anomalie, la cui presenza può essere determinata seguendo i cambiamenti che avvengono all'interno del nostro sistema; l'identificazione dei tipi di guasto, ossia la diagnostica, ponendo attenzione su quale parte del componente richiede attenzione; la previsione della transazione da stato di salute sano a stato di salute guasto, trovando un modello in grado di relazionare le *features* estratte con il trend di degrado del sistema, il quale permette di stimare quanto tempo è necessario prima che esso si guasti (*RUL*) e quando effettuare la manutenzione.

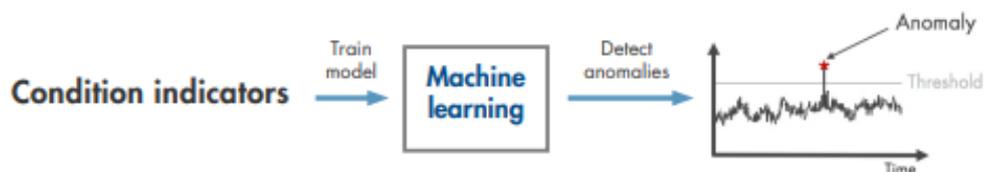


Figura 37 : Rilevamento delle anomalie [16]

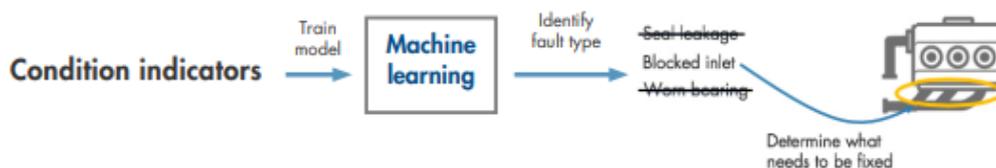


Figura 38 : Identificazione del tipo di guasto [16]

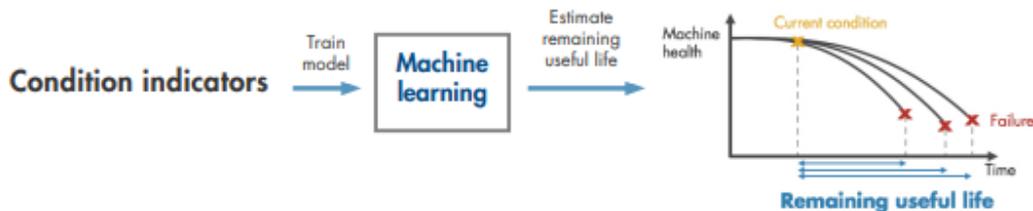


Figura 39 : Stima della vita utile residua (RUL) [16]

In base all’algoritmo scelto è possibile utilizzare metodi di apprendimento supervisionati o non supervisionati. Se i dati a disposizione sono etichettati allora è possibile utilizzare i metodi supervisionati per poter addestrare i modelli predittivi a differenziare le varie tipologie di modalità di guasto (Figura 38). Se i dati di addestramento non sono etichettati, invece, vengono utilizzati metodi non supervisionati, i quali risultano più adatti per il rilevamento delle anomalie (Figura 37), in quanto permettono di classificare le *features* estratte dal sistema come sane o guaste. Infine, per la stima della vita utile residua (Figura 39) vengono utilizzati dei metodi di apprendimento basati su probabilità e serie storiche, in quanto questi metodi utilizzano il valore attuale di un *Condition Indicator* e in base ad esso stimano quando il sistema subirà un guasto all’interno di un certo intervallo di confidenza.

Distribuzione dell’algoritmo:

Il passaggio finale prevede la distribuzione dell’algoritmo per il suo utilizzo. Dopo averlo sviluppato, infatti, è possibile implementarlo eseguendolo sul *cloud*, quando si sono raccolti una grande quantità di dati, o su un dispositivo *edge* come i *controller* industriali che possono elaborare rapidamente i dati dei sensori di alta frequenza in tempo reale. Un’altra via può essere quella di eseguirlo su dei dispositivi *embedded*, in particolare nei casi in cui non si abbia una connessione internet disponibile. Una combinazione di queste soluzioni prevede l’esecuzione delle fasi di pre-elaborazione ed estrazione delle *features* su un dispositivo *edge*, in modo da limitare i costi di trasmissione dei dati, e l’invio delle sole *features* estratte al modello di previsione, il quale verrà poi eseguito nel *cloud* (Figura 40).

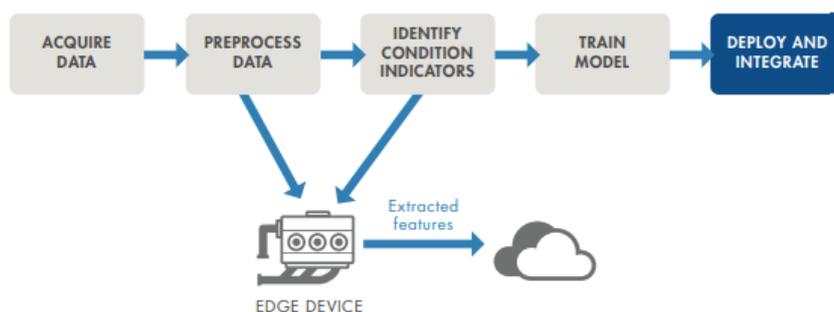


Figura 40 : Distribuzione dell’algoritmo su dispositivo edge e cloud [16]

4.2 Remaining Useful Life (RUL)

In questa sezione verrà approfondito maggiore l'argomento della stima della vita utile residua (RUL), in quanto sarà la tipologia di algoritmo su cui si basa l'algoritmo proposto in questa tesi. La RUL viene definita come il tempo che rimane tra lo stato attuale del nostro sistema e lo stato di guasto e la sua stima è uno degli obiettivi principali della Manutenzione Predittiva. La sua unità di misura non è definita a priori ma varia in base alla tipologia di sistema in analisi. Infatti, si può esprimere in giorni, numero di voli, cicli o quale che sia l'unità di nostro interesse.

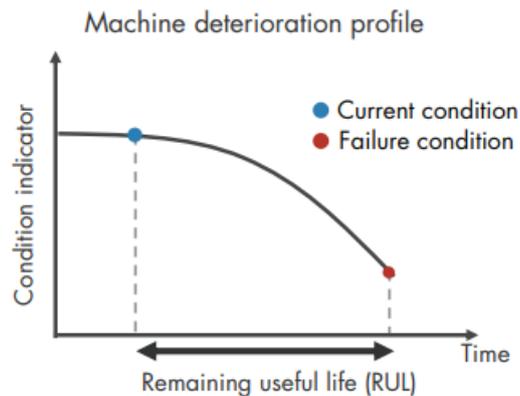


Figura 41 : Stima della RUL [17]

La RUL può essere stimata attraverso l'utilizzo di diversi modelli specializzati, che verranno introdotti a breve, i quali vengono scelti in base alla tipologia di dati misurati dal sistema. I modelli possono essere implementati quando si hanno a disposizione dei dati storici e delle informazioni come la storia di funzionamento fino al guasto (*run-to failure*) di macchine o sistemi simili a quelli di nostro interesse, un valore soglia (*threshold*) noto che indichi lo stato da cui inizia il guasto e quanto tempo di utilizzo è stato necessario a macchine o sistemi simili prima di raggiungere lo stato di guasto (*lifetime*). Tutte queste informazioni risultano essenziali per addestrare il modello ed effettuare delle previsioni sulla vita utile residua.

I modelli di stima della RUL seguono generalmente sempre lo stesso schema di lavoro che prevede: la scelta del miglior modello di stima in base ai dati e alle informazioni a disposizione; allenare il modello di stima utilizzando i dati storici in possesso; stimare la RUL del componente in analisi utilizzando i *test data* della stessa tipologia dei dati storici. I *test data* possono, inoltre, essere utilizzati più di una volta per aggiornare alcuni tipi di modelli, per mantenere la previsione accurata.

I modelli di stima della *RUL* possono essere diversi, come anticipato precedentemente, e vengono distinti in tre categorie, le quali vengono scelte basandosi sui dati e le informazioni di sistema a disposizione:

- I *Similarity Model* vengono utilizzati quando i dati a disposizione spaziano l'intero stato di un sistema, da uno stato di salute sano a uno stato di salute guasto

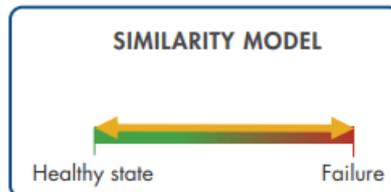


Figura 42 : Similarity Model [17]

- I *Degradation Model* vengono utilizzati quando si hanno dei dati intermedi tra lo stato di salute sano e lo stato di guasto, ma si conosce un valore soglia di sicurezza che non deve essere superato

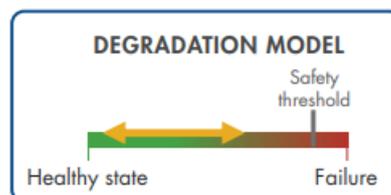


Figura 43 : Degradation Model [17]

- I *Survival Model* vengono scelti quando si hanno a disposizione solo i dati relativi al momento di guasto e nessuna storia completa di funzionamento (*run-to-failure*)

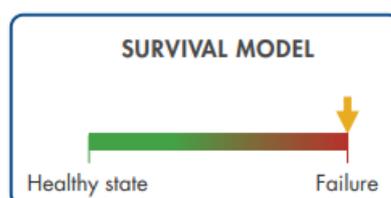


Figura 44 : Survival Model [17]

Similarity Model:

I *Similarity Model* vengono utilizzati per la previsione e la stima della *RUL* di un componente o sistema di cui si conosce il comportamento basandosi su database di macchine simili. Tali modelli, infatti, mettono a confronto gli andamenti dei dati o delle *features* estratte del componente in esame con le stesse informazioni estratte da sistemi simili. Per questo motivo risultano particolarmente utili quando si ha a disposizione la storia di funzionamento completo di componenti o sistemi simili, ossia si conosce come evolve lo stato di degradazione da uno stato di salute sano a uno stato di guasto, e i dati a nostra disposizione mostrano comportamenti di degrado simili a questi. Quindi, è possibile utilizzare i *Similarity Model* quando si possono estrarre dei profili di degrado dai nostri dati, i quali rappresentano l'evoluzione di uno o più *Condition Indicators* man mano che il sistema transita da uno stato sano a uno stato guasto.

All'interno del software Matlab è possibile implementare questo tipo di modello utilizzando il Toolbox "*Predictive Maintenance*", il quale include tre tipi diversi di *Similarity Model*. Ognuno di questi permette di stimare la *RUL* basandosi sul confronto tra l'andamento di degrado dei dati storici e dei dati a nostra disposizione, tramite l'utilizzo del comando "*predictRUL*". Per questa tipologia di modelli il comando calcola la *RUL* del componente come la media della vita operativa dei componenti più simili a quello in esame e sottraendogli il valore attuale della vita utile di quest'ultimo. I tre diversi modelli di *Similarity Model* differiscono tra loro in base a come definiscono e quantificano il concetto di somiglianza e si suddividono nel seguente modo [18]:

- Il primo modello che è possibile utilizzare tramite il software Matlab è il modello *Hashed-Feature Similarity*, il quale trasforma i dati storici sul degrado di ogni componente in informazioni a dimensioni fissa. Questo processo prevede l'estrazione di caratteristiche come la media, la potenza totale, i valori di massimo e minimo e altre quantità rappresentative. Il software calcola le *hashed features* a partire dai dati di degrado attuali e le confronta con i valori presenti nella tabella delle *hashed features* storiche. Questo confronto consente di identificare le somiglianze tra il componente in esame e quelli precedenti, migliorando la stima della *RUL*. Il modello *Hashed-Feature Similarity* risulta particolarmente utile quando si hanno a disposizione una grande quantità di dati di degrado, poiché esso riduce la quantità di dati necessari immagazzinati per la previsione. Tuttavia, la sua accuratezza dipende dall'accuratezza dell'*hash function* utilizzata dal modello.
- Il secondo modello disponibile è il modello *Pairwise Similarity*, il quale è in grado di stimare la *RUL* identificando i componenti i cui andamenti di degrado storici sono maggiormente correlati agli andamenti di degrado dei dati attuali. Il modello calcola la distanza tra due diverse serie temporali, dove la distanza viene definita come una correlazione, una *dynamic time warping (dtw)*, ossia una tecnica per misurare la similarità tra due serie temporali, oppure come una metrica personalizzata da noi fornita. Poiché considera i profili di degradazione nel tempo, la stima effettuata da questo modello risulta migliore rispetto a quella fornita del modello *Hashed-Feature Similarity*.

- Il terzo modello proposto è il modello *Residual Similarity*, il quale adatta a priori i dati in modo da modellarli con un andamento di utilizzo lineare o esponenziale nel tempo. Successivamente, calcola il residuo tra i dati storici e i dati attuali del nostro componente, similmente a quanto fa il modello *Pairwise Similarity*, ma la distanza metrica viene sostituita dall'intensità dei residui. Questo approccio risulta molto utile quando le conoscenze a nostra disposizione del sistema prevedono il modello di degrado.

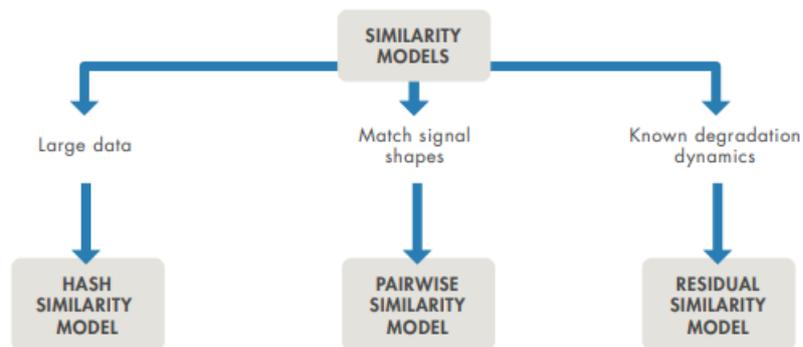


Figura 45 : Tipologie di Similarity Model [17]

Degradation Model:

I *Degradation Model* vengono utilizzati per la stima della *RUL* di un componente adattando un modello di tipo lineare o esponenziale ad un profilo di degrado di un *Condition Indicator*, noto il profilo di degrado storico del componente. Il profilo di degrado del componente o sistema in analisi viene, successivamente, utilizzato per determinare statisticamente la vita residua utile prima che si raggiunga un valore di soglia prestabilito, il quale rappresenta la condizione di guasto. Tramite l'utilizzo del software Matlab è possibile creare il modello e, successivamente, inizializzarlo utilizzando i dati storici riguardo la salute di un componente simile con il comando “fit” e stimare la *RUL* utilizzando “predictRUL”.

I *Degradation Model* spesso utilizzano un singolo *Condition Indicator*, ma è possibile anche utilizzare un *Condition Indicator* frutto di una fusione di informazioni di diversi indicatori. In ogni caso, l'indicatore utilizzato deve mostrare un evidente andamento crescente o decrescente, in modo da rendere il processo affidabile.

Come anticipato, esistono due tipologie di modelli di degradazione [18]:

- Il modello *Linear Degradation* viene utilizzato per descrivere il comportamento di degrado come lineare e risulta utile nel caso in cui il sistema non presenti un degrado cumulativo.
- Il modello *Exponential Degradation* viene utilizzato per descrivere il comportamento di degrado come esponenziale e risulta utile nel caso in cui il sistema presenti un degrado cumulativo.

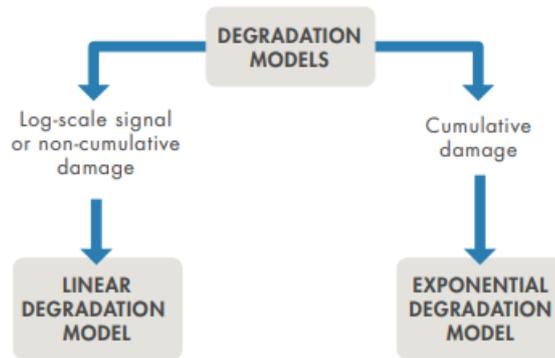


Figura 46 : Tipologie di Degradation Model [17]

Survival Model:

I *Survival Model* sono metodi di analisi statistica utilizzati per modellare i dati relativi solamente ad un singolo evento nel tempo e non si dispone di storie di funzionamento fino al guasto (*run-to-failure*). Esistono due tipologie di modelli che possono essere implementati tramite il software Matlab in questo caso [18]:

- Il modello *Reliability Survival* viene utilizzato nel caso in cui si disponga solamente di dati riguardo alla durata di vita operativa di componenti simili a quello in analisi. Note le informazioni storiche riguardo ai tempi di guasto di una flotta di componenti simili, il modello è in grado di calcolare la distribuzione di probabilità dei tempi di guasti, la quale verrà utilizzata per la stima della *RUL* del componente testato.
- Il modello *Covariate Survival* viene impiegato nel caso in cui sia disponibili sia la durata di vita operativa del componente, che altri dati variabili come le covariate, collegati alla stima della *RUL*. Le covariate sono anche note come variabili ambientali o variabili esplicative e contengono informazioni riguardo ai fornitori del componente, il lotto di produzione e i regimi operativi in cui il componente è stato utilizzato. Questo tipo di modello sfrutta, quindi, la durata di vita operativa e le covariate per calcolare la probabilità di sopravvivenza del componente in esame.

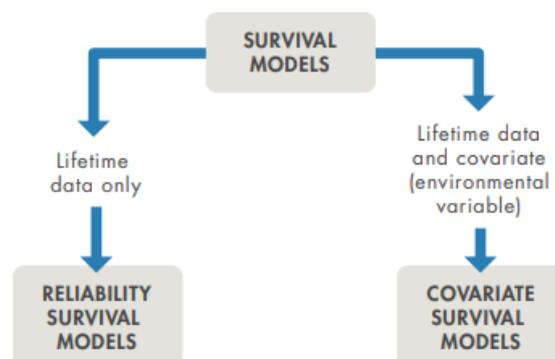


Figura 47 : Tipologie di Survival Model [17]

4.3 Sviluppo dell'algoritmo

In questa sezione verrà introdotto, spiegato e sviluppato l'algoritmo su cui si basa il lavoro di questa tesi. L'obiettivo che ci si è posti per tale lavoro è stato quello di sviluppare un processo logico da seguire per creare un algoritmo per la Manutenzione Predittiva attraverso un codice Matlab e tramite l'utilizzo di alcuni Toolbox presenti al suo interno, per ottenere una stima di vita utile residua (*RUL*) del modello di attuatore EMA presentato nel Capitolo 3. Lo studio del degrado si è concentrato, in particolare, sulla componente del motore elettrico brushless del servomeccanismo, andando a generare in maniera artificiale dei dati sintetici di degrado con cui poter procedere allo sviluppo dell'algoritmo, tralasciando la componente meccanica che veniva è stata poco sviluppata all'interno del modello. Le due variabili principali che sono state fatte degradare sono la resistenza e l'induttanza del motore, i cui trend di degrado sono basati su ragionevoli ipotesi prese dalla letteratura esistente e su leggi di degrado generali.

4.3.1 Definizione del parametro di degrado

Il primo compito che si è dovuto svolgere nello sviluppo di questo algoritmo ha riguardato l'identificazione delle variabili del motore elettrico che causano maggiore degrado e con quale andamento questi parametri degradano. Si è deciso di concentrarsi sulla resistenza e sull'induttanza del motore elettrico, identificando questi due parametri come variabili da controllare per verificare l'usura del motore e per stimarne la vita utile residua.

La sfida principale è stata ricavare delle leggi matematiche che rappresentassero l'andamento di degrado della resistenza e di induttanza. Non esistendo delle leggi specifiche per questi due parametri, si è scelto di basarsi su una formula empirica di degrado ricavata da "*Adaptive Control of Actuator Lifetime - Levent U. Gokdere, Alexander Bogdanov, Stephen L. Chiu, Kirby J. Keller, John Vian*" [19]. Essa rappresenta un'elaborazione della legge di *Arrhenius*, con la quale è possibile ricavarsi una legge di degrado di un componente generale. Nel nostro caso di studio si è voluto ottenere il parametro di degrado D tramite l'utilizzo di questa legge rielaborata, il quale si può esprimere come una funzione del tempo di funzionamento operativo del componente t , della temperatura degli avvolgimenti del motore elettrico T_w e di due coefficienti sperimentali α e β :

$$D = D_0 * e^{\gamma t}$$

Il parametro D_0 rappresenta il valore iniziale del parametro di degrado, per cui, essendo che il motore elettrico nell'istante iniziale si trova in uno stato di salute sano, dovrebbe essere nulla. In questo caso, però, viene assunto pari ad un valore di 10^{-4} , in maniera tale da far sì che il guasto si innesti come viene suggerito all'interno del documento "*Windings Fault Detection and Prognosis in Electro-Mechanical Flight Control Actuators Operating in Active-Active Configuration - Andrea De Martin, Giovanni Jacazio, and George Vachtsevanos*" [20].

Il parametro γ fa riferimento, invece, al degrado della vita operativa del componente in funzione della temperatura degli avvolgimenti del motore elettrico T_w :

$$\gamma = \beta * e^{-\alpha * T_w}$$

I due parametri sperimentali α e β si basano sempre sul documento da cui è stata presa la legge di degrado di D . Il parametro β rappresenta il tempo totale di vita operativa del componente, che nel caso del nostro motore elettrico è stato assunto, tramite una stima, a circa 20000 ore di funzionamento, per cui il tempo t viene fatto variare da 0 al tempo di funzionamento massimo pari a β . Tale valore, dopo alcune prove, è stato trasformato in settimane, in modo da avere valori del parametro γ non troppo elevati che potessero appesantire tutta la stima. Il parametro α è stato, invece, assunto proprio come coefficiente sperimentale, per cui si è preso pari al valore di $\alpha = 0.0693 \frac{1}{^\circ\text{C}}$ proposto all'interno del documento (Attenzione che i coefficienti α e β sono stati invertiti come nomenclatura rispetto a quella proposta dallo studio [19]).

Successivamente, si è dovuta valutare la temperatura degli avvolgimenti del motore elettrico T_w . Non avendo un valore prestabilito di questo parametro, si è deciso di valutarlo assegnando un range di temperatura, tra cui questo valore potesse stare. L'intervallo di funzionamento scelto è stato tra 80°C e 160°C. Per decidere quale tra queste temperature fosse stata la più adatta, si è deciso di effettuare un "plot" su Matlab del parametro di degrado D con i parametri sopra indicati e variando la temperatura degli avvolgimenti tramite un ciclo "for". Il grafico in Figura 48 mostra come sull'asse x sia presente il tempo di funzionamento massimo del componente, il quale si interrompe per un valore pari a 119 settimane (vita massima), mentre sull'asse y si trovi il *Fault Ratio* D . Avendo imposto un limite massimo sull'asse x , le curve graficate alle diverse temperature degli avvolgimenti si interrompono per una $T_w = 110 \text{ }^\circ\text{C}$, per cui la temperatura massima ammissibile con i parametri utilizzati è quella di $T_w = 100 \text{ }^\circ\text{C}$. In questo modo è stato possibile effettuare una stima della temperatura degli avvolgimenti del motore elettrico senza conoscere il valore a priori.

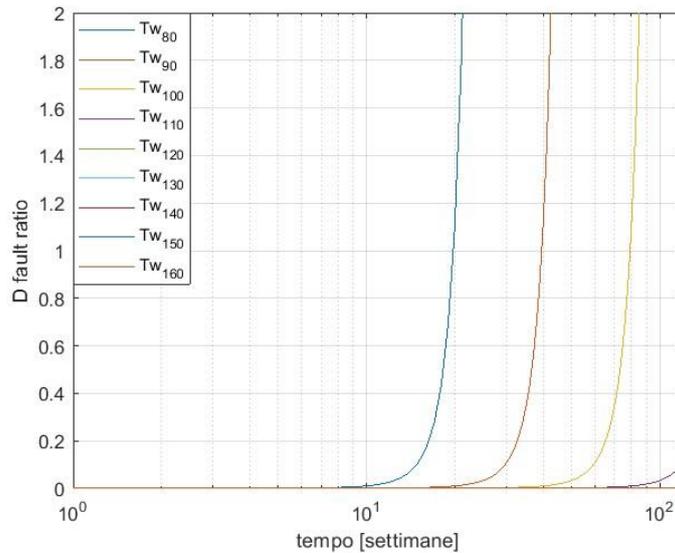


Figura 48 : Grafico del parametro di degrado D in funzione di diverse temperature degli avvolgimenti del motore elettrico

Dopo aver definito il valore della temperatura degli avvolgimenti del motore elettrico e avere scelto, quindi, la curva del parametro di degrado D di nostro interesse, si è deciso di individuare su essa dieci valori circa equi spazati tra loro, che rappresentassero uno stato di degrado sempre più avanzato. Si sono scelti, dunque dieci valori di tempo t , corrispondenti ad una determinata settimana del nostro tempo di utilizzo, partendo dall'istante iniziale ed arrivando ad un valore di t pari a 76 settimane. Tale valore è stato scelto in quanto corrispondente ad un valore di D pari a 0.7, il quale rappresenta il valore di soglia oltre cui il componente entra in uno stato di guasto avanzato e quindi catastrofico. L'assunzione di tale valore si è basata sulla scelta fatta all'interno del documento "*Windings Fault Detection and Prognosis in Electro-Mechanical Flight Control Actuators Operating in Active-Active Configuration - Andrea De Martin, Giovanni Jacazio, and George Vachtsevanos*" [20], in cui il valore di 0.7 è stato scelto come valore soglia per il parametro di degrado rappresentante il numero di avvolgimenti guasti sul numero di avvolgimenti totali per la fase in analisi.

Come mostrato in Figura 49, i dieci valori di t selezionati si trovano in uno stato già molto avanzato di funzionamento operativo, in quanto con la legge di degrado selezionato per le prime circa 40 settimane operative non si ha degrado, mentre, successivamente, il *Fault Ratio* aumenta vertiginosamente fino a raggiungere il valore limite di 0.7.

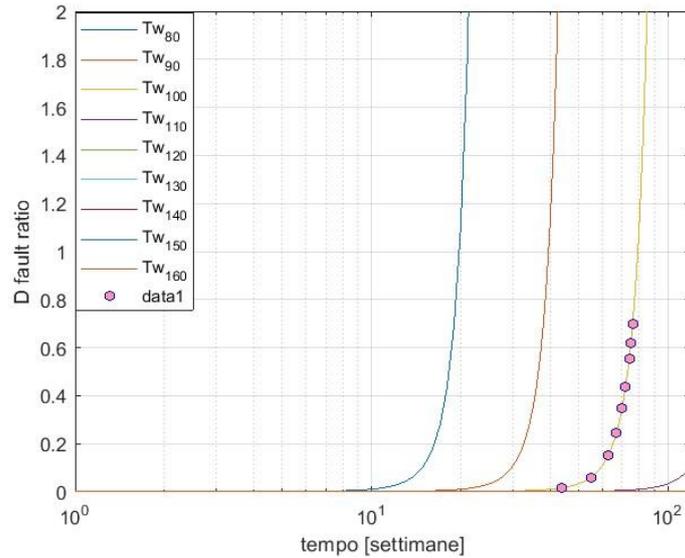


Figura 49 : Individuazione dei dieci valori del parametro di degrado D

A questo punto, noti i parametri β , α e T_w , è possibile calcolare il parametro γ per il nostro caso di studio. Tramite questo valore e i dieci valori di t selezionati nel grafico soprastante, è stato, infine, possibile definire i dieci valori del parametro di degrado D . Questi rappresentano diverse condizioni di salute del nostro componente, partendo dallo stato sano e arrivando al valore di soglia limite. In questo modo è stato, quindi, possibile definire dieci valori della resistenza e dell'induttanza del motore elettrico per ogni istante t individuato, utilizzando delle leggi di degrado lineare per i due parametri:

$$R = R_0 + D * R_0$$

$$L = L_0 + D * L_0$$

I valori di R_0 e L_0 rappresentano la condizione iniziale della resistenza e dell'induttanza per uno stato di salute sano e sono stati assunti pari ai valori di riferimento del documento su cui è basato il modello dell'attuatore elettromeccanico Matlab-Simulink presentato nel Capitolo 3.

4.3.2 Generazione dei dati

Grazie alle equazioni sopra riportate è stato possibile generare dieci valori di resistenza e dieci valori di induttanza variabili nel tempo, che andassero a simulare un degrado all'interno del motore elettrico del nostro modello Matlab-Simulink, seguendo un andamento basato sul parametro di degrado D .

A questo punto è stato possibile andare a simulare all'interno del modello questo degrado, lanciando dieci simulazioni per i dieci valori diversi di R ed L e andando a salvare in output le variabili di nostro interesse da andare a monitorare.

I parametri con cui sono state effettuate le simulazioni sono i seguenti:

<i>Tempo di simulazione</i>	4 s
<i>Intervallo di campionamento</i>	0.0005 s
<i>Frequenza del segnale</i>	0.5 Hz
<i>Frequenza di campionamento</i>	2000 Hz

Le simulazioni sono state effettuate lanciando direttamente un codice Matlab collegato al modello, in cui, tramite un ciclo "for", sono state salvate le variabili di output che si volevano monitorare. Si ricavano così delle *timeseries* composte nella prima colonna dall'istante di campionamento del segnale e dalla seconda all'undicesima colonna dai valori della variabile degradata nel tempo. Le variabili che si è voluto monitorare sono state:

- La corrente diretta i_d ;
- La corrente di quadratura i_q ;
- La coppia del motore elettrico T ;
- La velocità angolare dell'albero del motore elettrico ω_m ;
- L'accelerazione angolare dell'albero del motore elettrico ω_m' ;
- La posizione attuale dell'attuatore x_{pos} ;
- La velocità lineare di movimento dell'attuatore x_{pos}' ;
- La potenza attiva P_{att} ;
- La potenza apparente P_{app} ;
- La potenza elettrica persa per effetto Joule $P_{e_{loss_{joule}}}$;
- La potenza totale persa $P_{loss_{tot}}$;
- La potenza meccanica persa $P_{m_{loss}}$;
- La potenza meccanica erogata P_{mecc} ;

Per ognuna di queste variabili si è quindi ottenuta una *timeseries* contenente i propri valori degradati per le dieci simulazioni e per valutare quali tra queste subisse maggiormente l'usura dovuta alla variazione della resistenza e dell'induttanza, si sono graficati i loro andamenti al variare di questi due parametri.

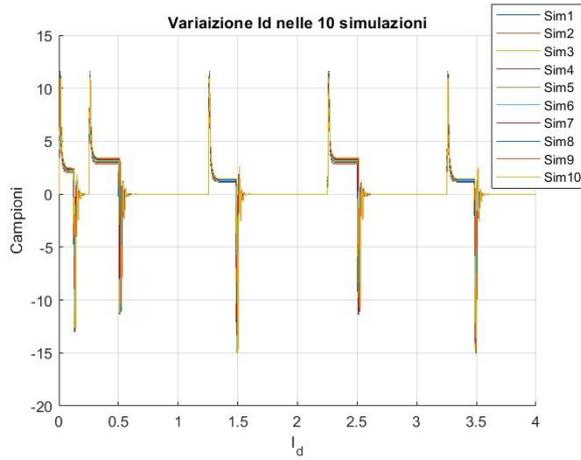


Figura 50 : Variazione della corrente diretta

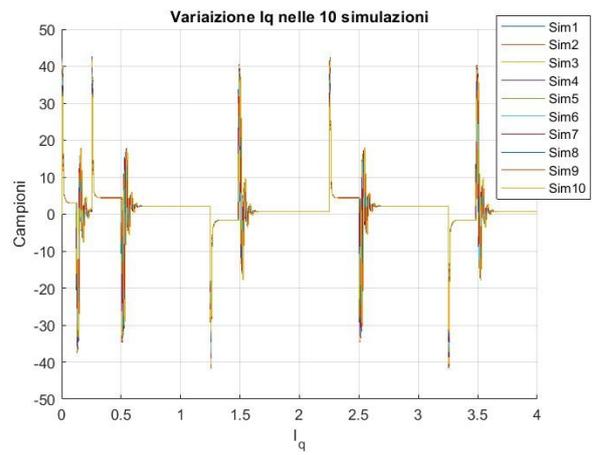


Figura 51: Variazione della corrente di quadratura

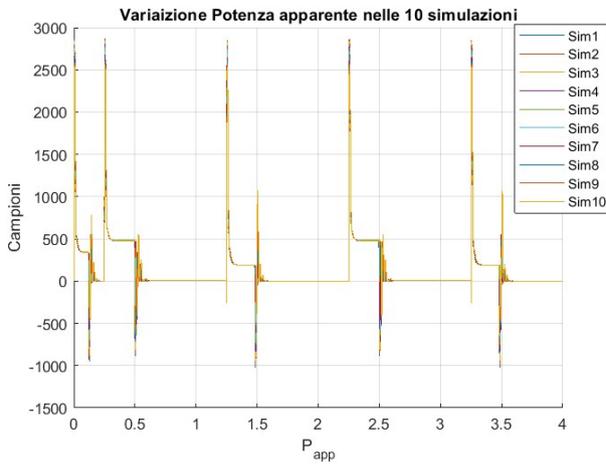


Figura 52 : Variazione della potenza apparente

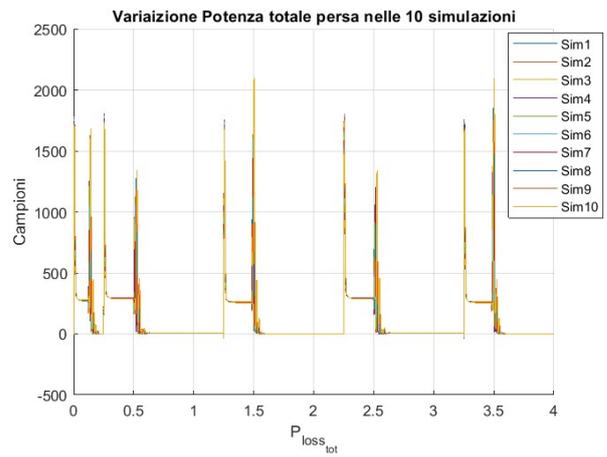


Figura 53 : Variazione della potenza totale persa

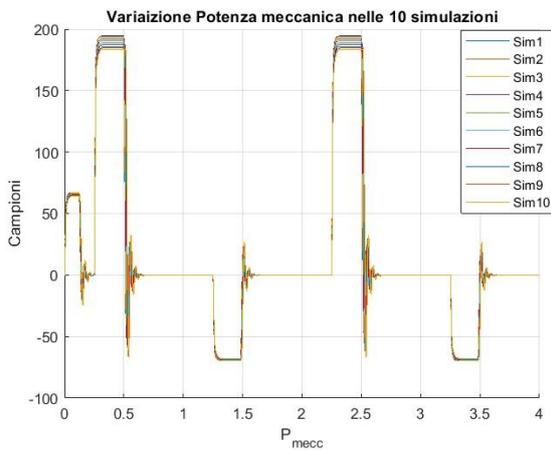


Figura 54 : Variazione della potenza meccanica

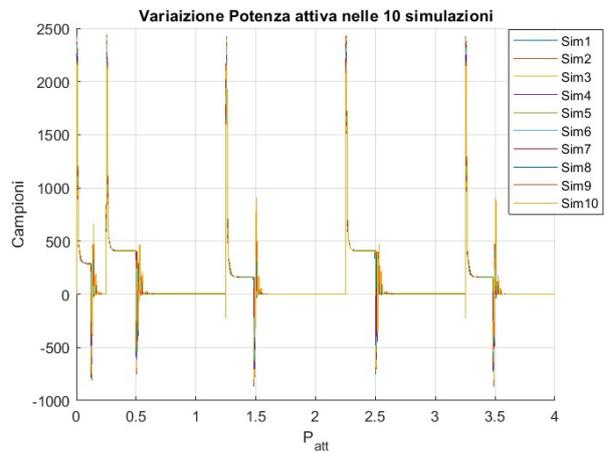


Figura 55 : Variazione della potenza attiva

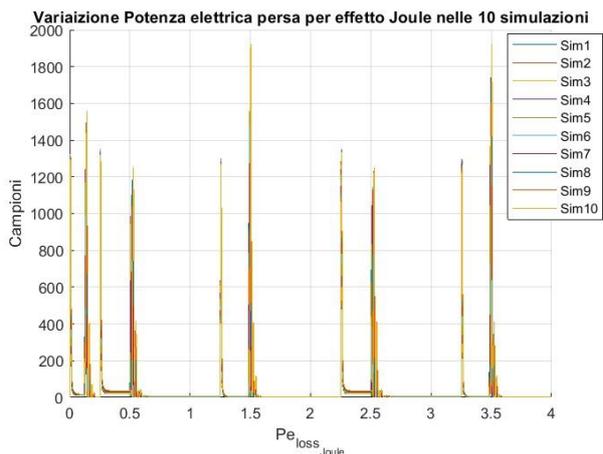


Figura 56 : Variazione della potenza elettrica persa

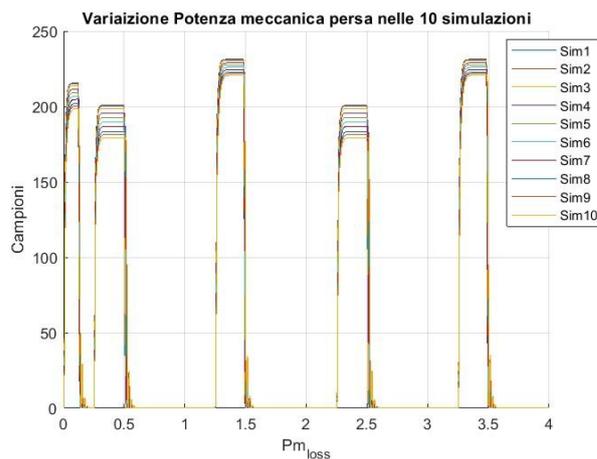


Figura 57 : Variazione della potenza meccanica persa

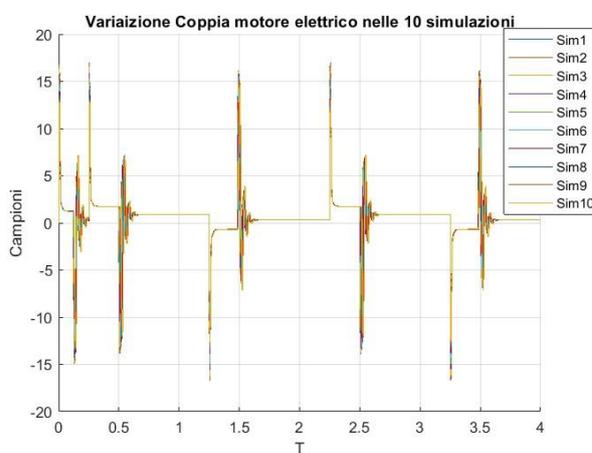


Figura 58 : Variazione della coppia

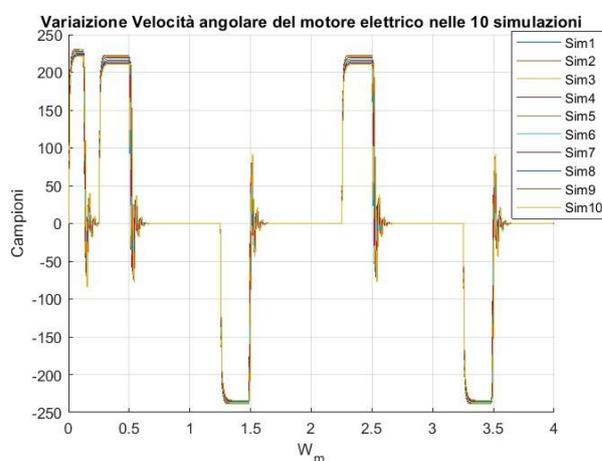


Figura 59 : Variazione della velocità angolare

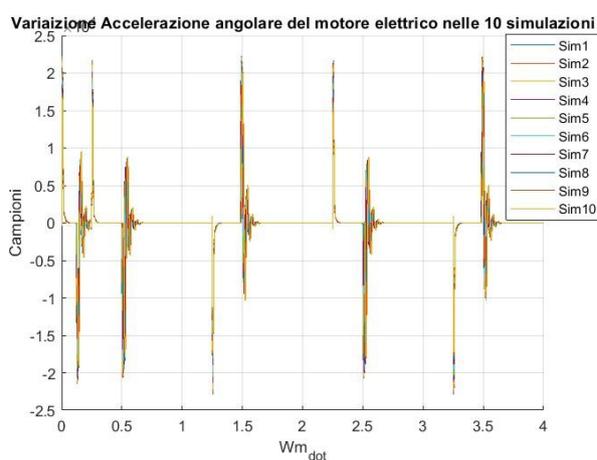


Figura 60 : Variazione dell'accelerazione angolare

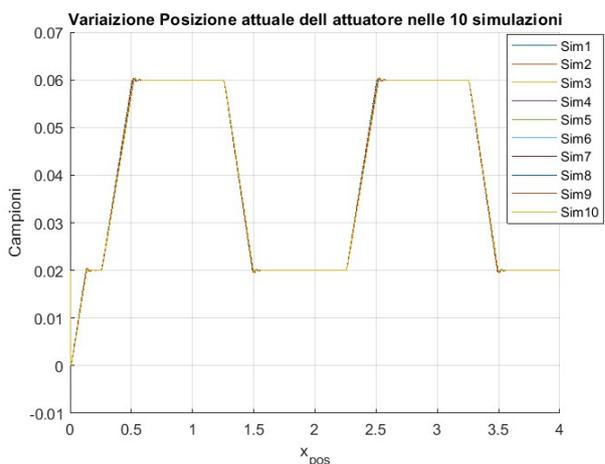


Figura 61 : Variazione della posizione attuale

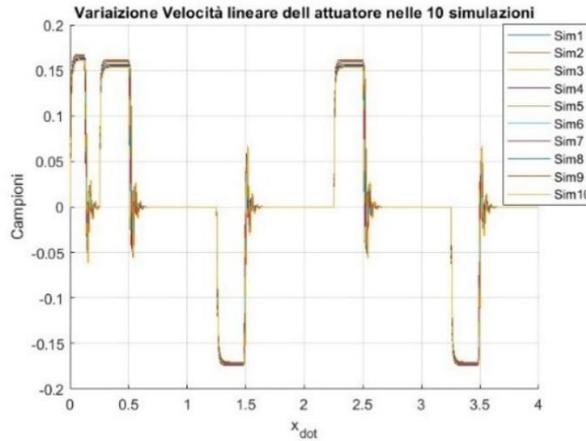


Figura 62 : Variazione della velocità lineare dell'attuatore

Tramite questi grafici è possibile osservare come alcune variabili subiscano maggiormente il degrado dei parametri di resistenza ed induttanza rispetto ad altri. Ad esempio, in Figura 50, in Figura 54, in Figura 57, in Figura 59 e in Figura 62, rappresentati rispettivamente gli andamenti di corrente diretta, potenza meccanica, potenza meccanica persa, velocità angolare del motore elettrico e velocità lineare dell'attuatore, si può notare una netta distinzione tra le varie simulazioni, a dimostrare come questi parametri vengano fortemente influenzati dal degrado di R ed L . Una volta ottenute, quindi, le variabili di output desiderate, è stato necessario trasformarle in un formato adatto per poterle fornire in input al Toolbox Matlab "Diagnostic Feature Designer", che verrà introdotto successivamente. Per far questo, si è dovuta compiere una manipolazione dei dati a nostra disposizione, poiché il formato richiesto in input dal Toolbox era di tipo "Tabella", contenete tutti i valori di tutte le variabili per ogni simulazione. Nel nostro caso, invece, eravamo in possesso solo di *timeseries* per ogni variabile al cui interno erano presenti i valori campionati per ogni simulazione. Si è quindi proceduto a trasformare ogni colonna di ogni vettore *timeseries* in un formato "timetable", in modo da avere una *timetable* per ogni simulazione di ogni variabile. Avendo tredici variabili e dieci simulazioni si generano quindi centotrenta *timetable*, le quali sono state inserite all'interno della *Tabella* costruita nel seguente modo: ogni colonna della *Tabella* rappresenta una variabile, mentre ogni riga i -esima della colonna contiene la *timetable* per ogni simulazione. In questo modo si è costruita una *Tabella* 10x13 composta come riportato in Figura 63.

	1	2	3	4	5	6	7	8	9	10	11	12	13
	Id	Iq	P_loss_tot	P_mecc	P_app	P_att	Pe_loss_Joule	Pm_loss	T	Wm	Wm_dot	x_dot	x_pos
1	8001x1 timetable												
2	8001x1 timetable												
3	8001x1 timetable												
4	8001x1 timetable												
5	8001x1 timetable												
6	8001x1 timetable												
7	8001x1 timetable												
8	8001x1 timetable												
9	8001x1 timetable												
10	8001x1 timetable												

Figura 63 : Formato della Tabella da fornire in input al Diagnostic Feature Designer

4.3.3 Diagnostic Feature Designer

Il Toolbox Matlab “*Diagnostic Feature Designer*” consente all’utente di generare delle *features* dai dati che gli vengono forniti in input attraverso un’interfaccia grafica multifunzionale. Tramite questa applicazione è possibile progettare e confrontare le *feature* in modo interattivo per determinare quali siano le più efficaci per determinare la tipologia di dati proveniente da un sistema sano e da un sistema guasto. Queste verranno utilizzate come *Condition Indicators* per la diagnosi dei guasti e per la prognostica [21].

Proprio per tale motivo il “*Diagnostic Feature Designer*” è stato utilizzato all’interno del nostro algoritmo, in quanto ci ha permesso di estrarre le *feature* dai dati sintetici generati e di utilizzarle nei passaggi successivi come indicatori di condizione.

L’utilizzo dell’applicazione prevede che alla sua apertura venga importata la *Tabella* da noi creata nel passaggio precedente e contenente tutte le variabili per tutte le simulazioni. Avendogli fornito i dati nel corretto formato, il tool è in grado di suddividere le informazioni presenti all’interno della *Tabella* riconoscendo la variabile indipendente *Tempo*, il *Segnale* corrispondente alla variabile e il *Nome* della variabile stessa. A questo punto, si importano i segnali suddivisi all’interno dell’applicazione ed è possibile procedere alla loro manipolazione per l’estrazione delle *features*.

Il primo passaggio prevede il *preprocessing* dei segnali forniti. Ciò viene fatto andando nella sezione “*Data processing*”, selezionando “*Filtering & Averaging*” e successivamente andando in “*Signal Envelop*”.

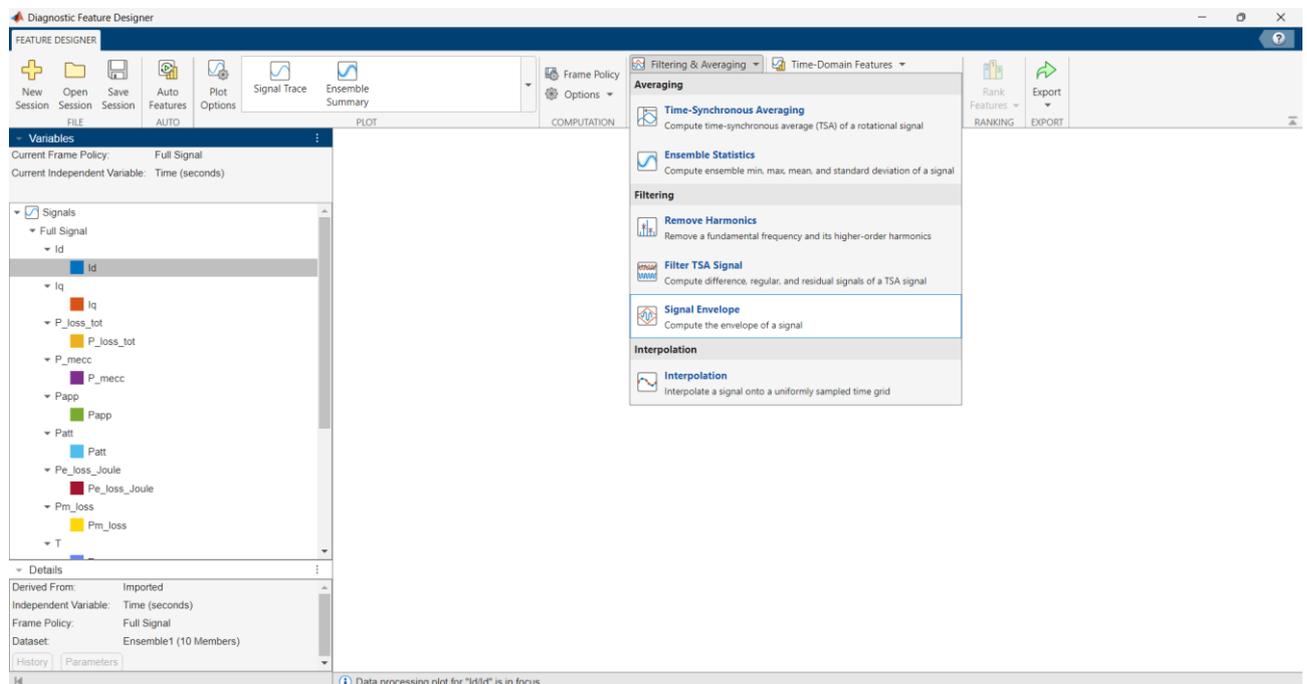


Figura 64 : Preprocessing dei segnali forniti

Questo primo passaggio serve per isolare le modulazioni di ampiezza nel segnale, per filtrare le componenti rumorose e per la diagnosi delle vibrazioni. Il metodo scelto nel “*Signal Envelope*” è il metodo di *Hilbert*, uno dei metodi più comuni per svolgere questa funzione, mentre il filtro passa-banda non viene applicato in quanto non si riscontra la presenza di rumore o vibrazioni per la natura artificiosa dei nostri segnali.

Dopo aver effettuato questa prima fase di processamento dei segnali, è possibile passare alla sezione “*Feature Generation*”. Qui è possibile scegliere la tipologia di *features* che si vuole estrarre e nel nostro caso utilizzeremo delle *feature* nel dominio del tempo, per cui si andrà in “*Time-domain Feature*” e si selezionerà “*Signal Feature*”.

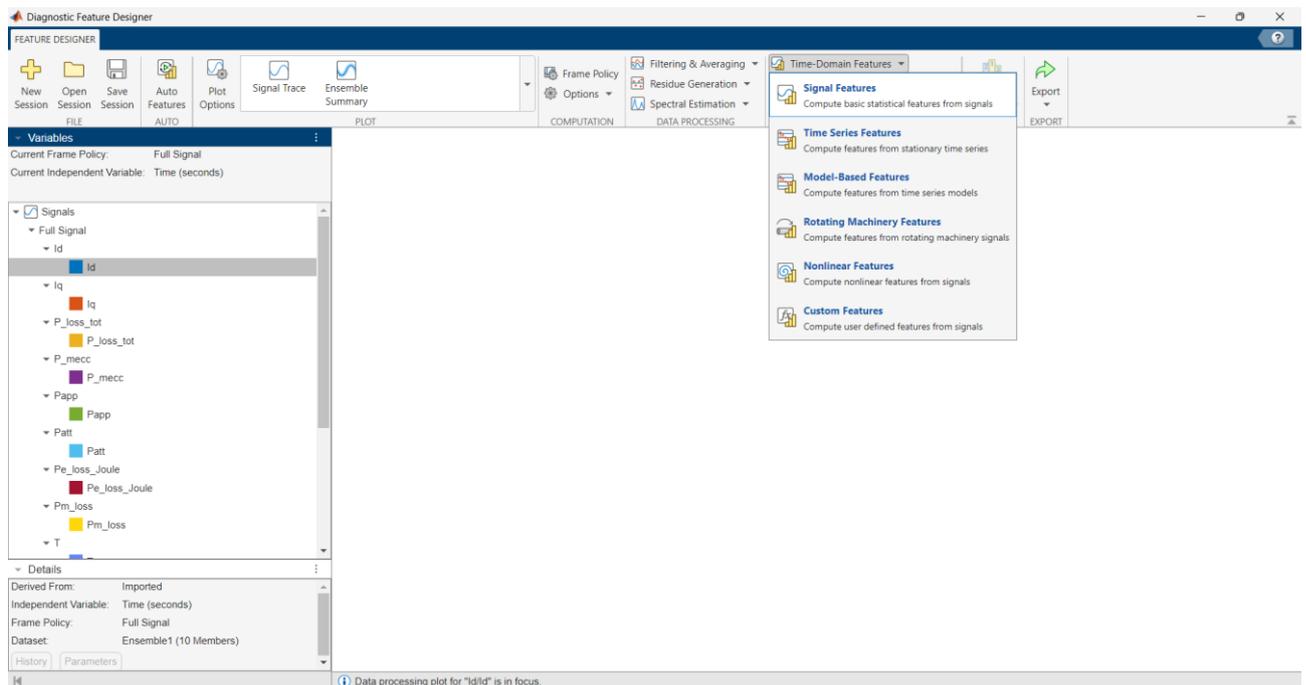


Figura 65 : Scelta della tipologia di feature

Tramite questa scelta è possibile generare delle *features* statistiche base, le quali possono essere applicate a qualsiasi tipo di segnale. All’interno del “*Signal Features*” è possibile selezionare le caratteristiche di nostro interesse che vengono suddivise in tre categorie riportate di seguito [22].

Statistical Features:

Le features statistiche includono la media, la deviazione standard e il valore quadratico medio (RMS) come caratteristiche principali. Inoltre, si trovano anche il *Fattore di Forma* e le statistiche di ordine superiore *Kurtosis* e *Skewness*, le quali forniscono informazioni sul

comportamento del sistema attraverso il quarto momento e il terzo momento del segnale di vibrazione rispettivamente.

Il *Fattore di Forma* dipende dalla forma del segnale, mentre è indipendente dalle dimensioni del segnale stesso. Esso si calcola come il valore quadratico medio diviso la media del valore assoluto.

$$x_{SF} = \frac{x_{RMS}}{\frac{1}{N} \sum_{i=1}^N |x_i|}$$

Il *Kurtosis* descrive la lunghezza delle code di una distribuzione del segnale o, in altre parole, la propensione del segnale a contenere valori anomali (*outlier*). L'aumento del degrado può, infatti, aumentare il numero di *outlier*, determinando un incremento del valore del *Kurtosis*. Esso assume un valore pari a 3 per una distribuzione normale, mentre un valore superiore a questo indica la presenza di una maggiore concentrazione di valori estremi rispetto ad una distribuzione normale, suggerendo la presenza di eventuali anomalie.

$$x_{Kurt} = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^4}{\left[\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \right]^2}$$

Lo *Skewness* descrive l'asimmetria della distribuzione di un segnale. Le anomalie possono impattare sulla simmetria della distribuzione e perciò far aumentare il valore dello *Skewness*.

$$x_{Skew} = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^3}{\left[\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \right]^{\frac{3}{2}}}$$

Impulsive Metrics:

Le metriche impulsive sono delle caratteristiche legate ai picchi dei segnali ed includono:

Il *Valore di Picco* che rappresenta il massimo valore assoluto del segnale e viene utilizzato per calcolare le altre metriche impulsive.

$$x_p = \max_i |x_i|$$

Il *Fattore di Impulso* mette a confronto l'altezza dei picchi con il valore medio dei segnali.

$$x_{IF} = \frac{x_p}{\frac{1}{N} \sum_{i=1}^N |x_i|}$$

Il *Fattore di Cresta* viene definito come il *Valore di Picco* di un segnale diviso per il valore quadratico medio. Le anomalie, spesso, tendono a manifestarsi inizialmente con delle variazioni nei picchi del segnale prima ancora di manifestarsi nell'energia rappresentata dal valore quadratico medio del segnale stesso. Per questo motivo, il fattore di cresta serve da indicatore precoce quando i primi guasti iniziano a svilupparsi, consentendo di rilevare anomalie prima che diventino evidenti attraverso altre metriche.

$$x_{Crest} = \frac{x_p}{\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}}$$

Il *Clearance Factor* viene calcolato come il *Valore di Picco* diviso per la media quadratica dei valori assoluti delle ampiezze dei segnali. In particolare, questa caratteristica risulta molto utile per le macchine rotanti, in quanto assume un valore massimo quando i cuscinetti sono in uno stato di salute sano e tende a diminuire progressivamente in presenza di difetti che vanno dal difetto sulla sfera, al difetto sulla pista esterna e al difetto sulla pista interna.

$$x_{Clear} = \frac{x_p}{\left(\frac{1}{N} \sum_{i=1}^N \sqrt{|x_i|}\right)^2}$$

Signal Processing Metrics:

Le metriche di elaborazione del segnale comprendono funzioni di misurazione della distorsione utili per rilevare il degrado di un sistema. Il deterioramento di un componente può, infatti, manifestarsi con un aumento del rumore e una variazione dell'ampiezza di una componente armonica rispetto alla fondamentale. Esistono tre tipologie di feature per questa metrica.

Il *Rapporto Segnale-Rumore (SNR – Signal-to-Noise Ratio)* misura il rapporto tra la potenza del segnale e la potenza del rumore ed un suo valore elevato indica quindi un segnale più pulito con una minore presenza di rumore.

La *Distorsione Armonica Totale (THD – Total Harmonic Distortion)* misura il rapporto tra la potenza totale delle componenti armoniche indesiderate e la potenza della componente fondamentale, per cui un suo aumento può indicare una degradazione del sistema.

Il *Rapporto Segnale-Rumore e Distorsione (SINAD – Signal-to-Noise and Distortion)* rappresenta il rapporto tra la potenza totale del segnale e la potenza complessiva di rumore e distorsione. Tale rapporto è utile per valutare la qualità complessiva del segnale in presenza di entrambi i tipi di degrado.

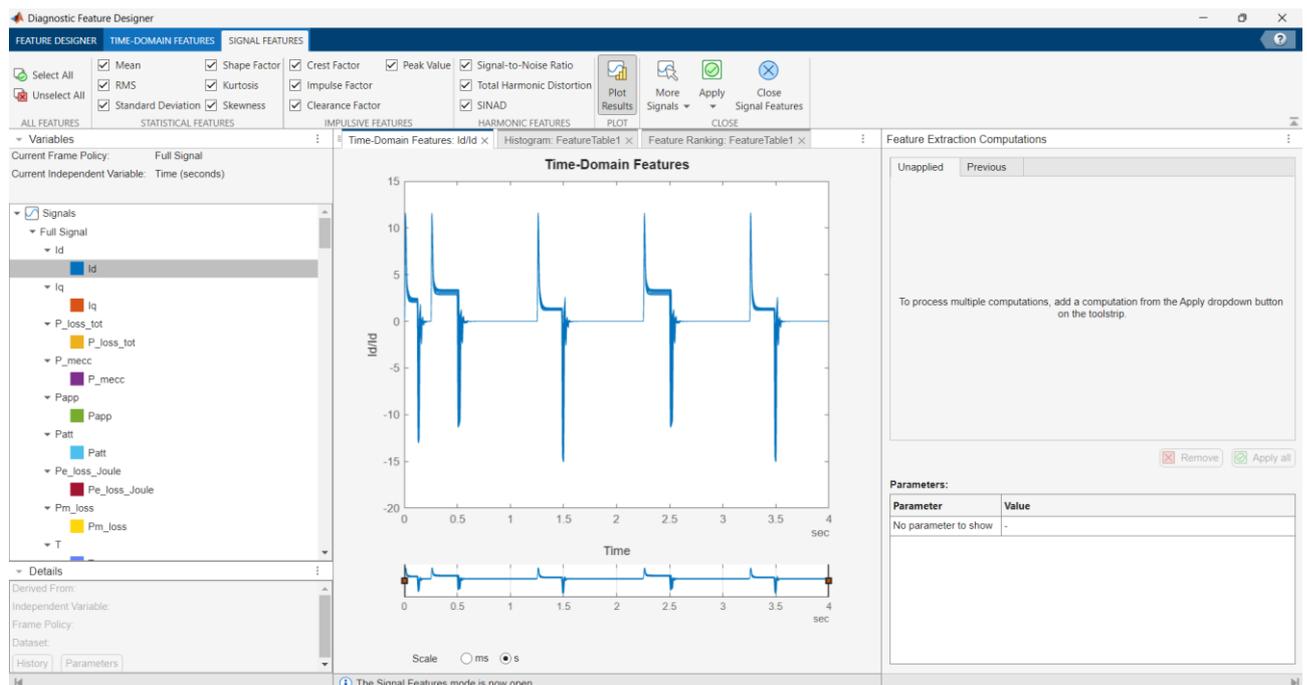


Figura 66 : Selezione delle feature da estrarre

Selezionate tutte le *features* sopra riportate insieme a tutti i segnali che si vogliono analizzare, si può procedere all'estrazione delle caratteristiche tramite il comando “*Apply*”. Si aprirà una nuova finestra in cui compariranno degli istogrammi rappresentati tutte le *features* selezionate per ognuno dei nostri segnali. Questi servono per valutare l'andamento delle caratteristiche e tramite il comando “*Select Feature*” è possibile graficare alcune *features* a nostra scelta nel caso in cui ci si voglia concentrare solo su di esse e non su tutte quelle a disposizione.

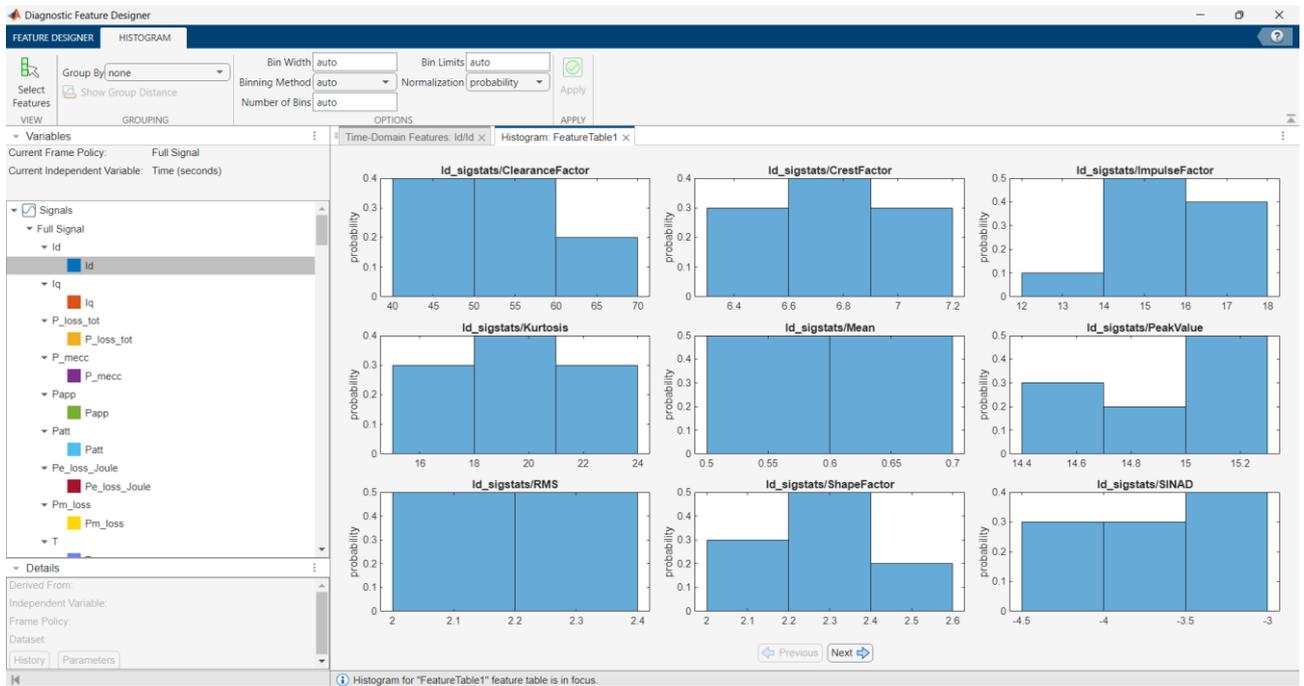


Figura 67: Istogrammi delle features per il segnale della corrente diretta i_d

L'ultimo passaggio da seguire riguarda la classificazione delle *feature* create per determinare quali tra queste sono le migliori nell'individuare le variazioni di comportamento all'interno dei segnali, tramite l'utilizzo del comando "Rank Feature". Si seleziona la *FeatureTable* creata al comando precedente, contenente tutte le caratteristiche di ogni segnale, e si procede alla selezione del metodo di classificazione. Esistono tre metodi con cui è possibile classificare le *features* che sono: la classificazione supervisionata, la quale viene utilizzata quando le *features* sono etichettate, per determinare quale tra queste è più adatta a distinguere i comportamenti nominali da quelli anomali; la classificazione non supervisionata, la quale viene utilizzata quando i nostri segnali non hanno nessuna variabile di condizione o etichetta per determinare quali *features* mostrano il miglior raggruppamento con le altre e sono più adatte per distinguere le condizioni operative da quelle anomale; la classificazione prognostica viene utilizzata per estrarre le *features* quando si hanno a disposizione dai dati di funzionamento fino al guasto (*run-to-failure*), le quali risultano utili per determinare la vita utile rimanente (*RUL*).

Nel caso in studio è stato scelto il "Prognostic Ranking". Questo metodo permette di valutare quanto efficacemente ogni *feature* traccia il degrado dei segnali, classificando per prime quelle migliori per la previsione della *RUL*. Il metodo utilizzato per classificare ognuna di queste caratteristiche è quello della monotonicità, il quale caratterizza la tendenza di una *feature* mentre il sistema evolve verso lo stato guasto. Quando il sistema si avvicina progressivamente alla condizione di *failure*, un *Condition Indicator* idoneo ha una tendenza monotona positiva o negativa. Una volta selezionato il metodo è possibile, inoltre, impostare il fattore "Correlation Importance", il quale consente di escludere le *features* che trasmettono informazioni simili alle features di rango superiore, fornendo un set di

caratteristiche maggiormente diversificato nei ranghi più alti. Il criterio su cui si basa questa esclusione riguarda l'insieme dei coefficienti di cross-correlazione che una *feature* possiede nei confronti di un'altra di rango maggiore. Un elevato valore di questo cross-correlazione tra le due implica che queste forniscono informazioni ridondanti. Lasciando il valore di *default* dell'applicazione pari a 0, la ridondanza tra le caratteristiche non viene tenuta in conto, mentre più si aumenta il valore di “*Correlation Importance*”, più l'influenza della cross-correlazione tra le varie *features* viene presa in considerazione nella classificazione di queste. Un valore più alto fa diminuire progressivamente il numero di *features* ridondanti e per questo viene scelto di impostare tale fattore pari a 0.5.

Come si può notare dalla Figura 68, le diverse caratteristiche estratte da ogni segnale vengono classificate utilizzando il metodo scelto e si decide di tenere in conto solamente di quelle che hanno un “*Ranking Score*” superiore ad un valore di 0.5, considerando irrilevanti quelle con un valore inferiore. Ciò viene fatto per eliminare alcune *features*, in quanto se ne dispone di un numero molto elevato.

Infine, è possibile esportare le *features* estratte, classificate e selezionate tramite il comando “*Export*” con cui si può generare una *function* Matlab contenente tutti i passaggi appena compiuti. In questo modo, fornendo una nuova *Tabella* con altri dati alla *function* “*diagnosticFeature*” appena generata, si ricaverà in output una *FeatureTable* con le nuove caratteristiche che fungeranno da *Condition Indicators*.

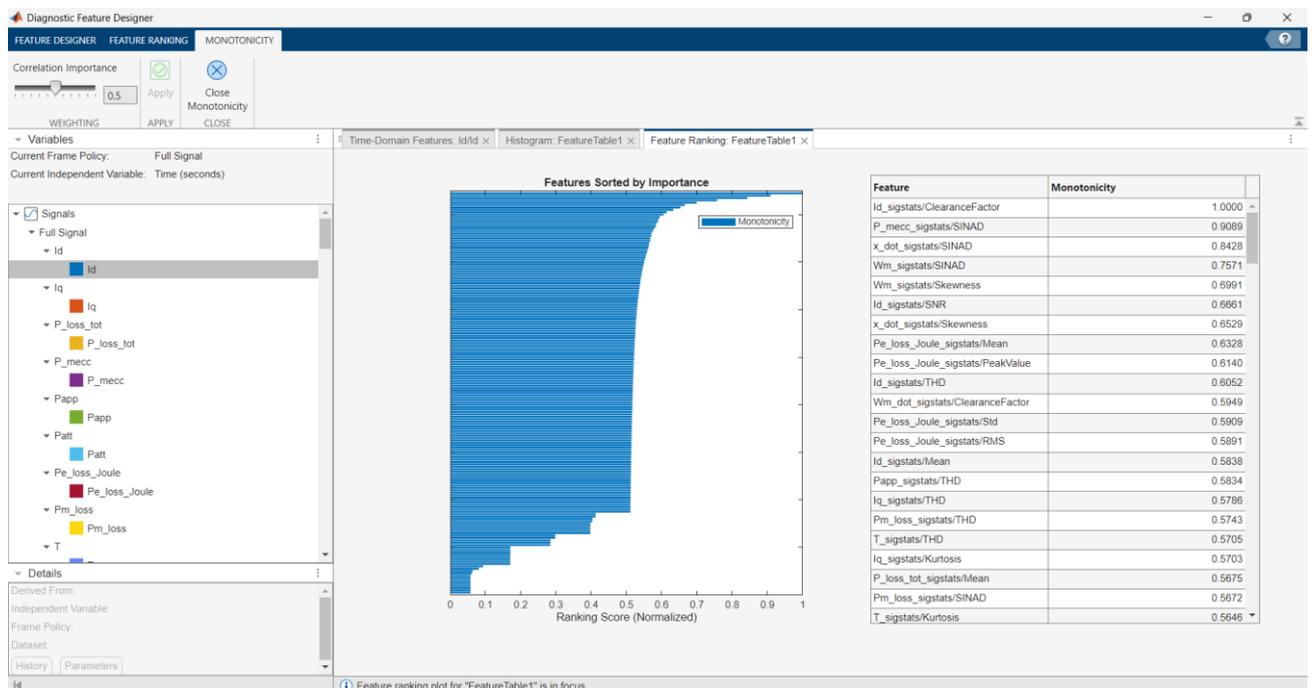


Figura 68 : Classificazione delle feature con il metodo “*Prognostic Ranking*”

4.3.4 Health Indicator Designer

Il Toolbox successivo, che viene utilizzato all'interno del nostro algoritmo, è il “*Health Indicator Designer*”. Questa applicazione consente di unire un insieme di *features* all'interno di un singolo indicatore di salute (*HI*), che rappresenta lo stato di salute generale del sistema e che può essere utilizzato per predire la vita utile rimanente (*RUL*) del componente.

Il “*Health Indicator Designer*” prende in input la *FeatureTable* generata dal “*Diagnostic Feature Designer*” e la trasforma in un valore scalare (*HI*) tramite l'utilizzo della seguente equazione di regressione lineare [23]:

$$H(t) = a_0 + a_1f_1(t) + a_2f_2(t) + \dots + a_nf_n(t)$$

All'interno di questa equazione i termini a_i sono dei coefficienti che pesano il contributo della *feature* per l'identificazione dello stato di salute, i termini f_i sono i valori delle *features* stesse all'istante t , mentre il termine a_0 è il valore dell'intercetta, ossia l'ordinata a cui la retta interseca l'asse y . Il termine H così calcolato varia in un intervallo che va da 0 a 1 o da 1 a 0. Tramite l'utilizzo dell'applicazione è possibile impostare una serie di parametri, che possono essere fatti variare in base alle richieste, e sono presenti dei grafici, che mostrano i risultati dell'adattamento della retta dell'equazione con i parametri forniti. In questo modo è possibile controllare il numero di *features* che l'indicatore di salute *HI* ingloba, oltre che l'errore quadratico medio (*MSE*) dell'adattamento degli *HI* rispetto ai valori *target*. L'obiettivo principale è cercare di utilizzare il minor numero possibile di *features* all'interno degli *HI* per ottenere l'errore minimo richiesto. Gli indicatori di salute rappresenteranno lo stato di salute del componente da una condizione sana, con *HI* pari a 1, fino ad una condizione guasta, *HI* pari a 0, in modo che le *features* generate in precedenza vengano associate ad un determinata condizione di salute durante la vita operativa del componente.

Il primo passaggio da compiere, non appena viene aperta l'applicazione, riguarda l'importazione della *FeatureTable* generata al passaggio precedente, dove viene richiesto il “*Feature data*”. Successivamente, si imposta come retta *target*, a cui si dovrà adattare quella generata con i nostri dati, un *Negative Linear Trend* in “*Target health indicator*”, poiché, essendo che il nostro componente si sta degradando nel tempo, partirà da uno stato di salute sano pari a 1 e andrà verso uno stato di salute guasto pari a 0. Tale retta *target* rappresenta il profilo di degrado che ci si aspetta segua il modello di degradazione *RUL*.

Come mostrato in Figura 69, all'interno del tool sono presenti altri due ulteriori grafici, i quali sono quelli che permettono di modificare i parametri della nostra retta degli indicatori di salute. Il grafico sulla destra, identificato come “*Feature Coefficient Trajectory*”, presenta sull'asse x il parametro di ottimizzazione λ e mostra come variano i valori dei coefficienti dell'equazione di H al variare di λ . Più è elevato il valore dei coefficienti, maggiore sarà il contributo della rispettiva *feature* al valore di *HI*, mentre quando il valore dei coefficienti è nullo, allora questo non contribuirà all'indicatore di salute. Il grafico sulla sinistra, invece, identificato come “*Cross-validated Mean Squared Error*”, mostra l'errore quadratico medio e la varianza dell'errore del modello corrente rispetto al profilo *target*. Sull'asse delle ascisse

è presente nuovamente il parametro di ottimizzazione λ , il quale può variare da sinistra a destra assumendo valori elevati, per cui ci sarà un basso numero di *features* all'interno del modello *HI* e perciò un alto errore, oppure valori inferiori, a cui corrispondono un alto numero di *features* e un piccolo errore. Su entrambi i grafici sono presenti, infine, tre linee verticali, le quali vengono chiamate “*Selected Fit*”, “*Best Fit*” e “*Optimal Fit*”. La prima rappresenta il valore corrente del parametro λ nel modello, la seconda rappresenta il valore di λ che il software ritiene produca il miglior adattamento in termini di errore quadratico medio, mentre la terza rappresenta il valore di λ che il software ritiene produca il miglior adattamento in termini sia di errore quadratico medio che di varianza dell'errore.

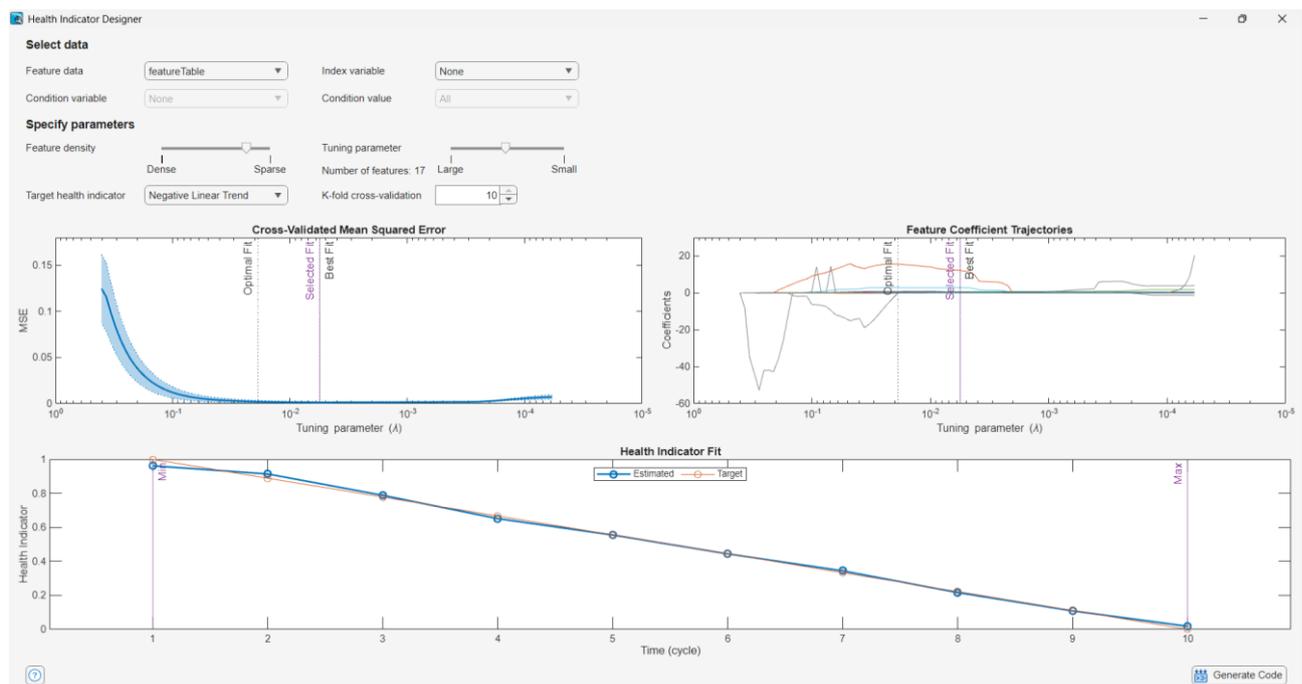


Figura 69 : Health Indicator Designer

A questo punto si fa variare il “*Tuning parameter*”, in modo da allineare la linea del “*Selected Fit*” sui grafici con quella del “*Best Fit*”, in modo da minimizzare il valore dell'errore quadratico medio. Guardando il “*Number of features*” si può riscontrare, quindi, che le caratteristiche prese in considerazione all'interno degli *HI* sono solamente diciassette e con tale numero vediamo che la nostra retta stimata segue molto bene la retta *target*. Come prova si è fatto variare il “*Tuning parameter*” verso destra, in modo da prendere in considerazione tutte le *features* e ottenere il valore minimo di errore quadratico medio a scapito di un costo computazionale maggiore, ma non si è riscontrata nessuna grande variazione nell'adattamento tra le due rette, per cui sono sufficienti le sole diciassette *features* selezionate, mentre le restanti possono essere escluse. Infine, spostando, invece, il “*Tuning parameter*” verso sinistra, nella condizione di minor numero di *features* ma a

discapito di un elevato errore quadratico medio, si è notato come la retta stimata non segua affatto quella *target*, a prova che un elevato *MSE* porta ad un errato adattamento della retta.

Un ulteriore parametro che può essere modificato è il “*Feature Density Slider*”, il quale è utile per regolare la correlazione tra le diverse *features*. Muovendo il parametro verso la condizione “*Dense*”, il tool non tiene in considerazione la cross-correlazione tra le caratteristiche, mentre spostandolo verso la condizione “*Sparse*”, questa verrà valutata esaltando le *features* con una classificazione d’importanza più elevata.

Infine, è presente il “*K-fold cross-validation*”, il quale utilizza i dati che non sono stati impiegati durante l’addestramento del modello per validarne le prestazioni. Il metodo suddivide i dati in sotto-dataset e addestra e testa il modello in modo iterativo per determinare quale abbia le prestazioni migliori. Tipicamente questo parametro è impostato ad un valore pari a 5 o 10 [23], poiché questi valori offrono un buon compromesso tra l’affidabilità della stima dell’errore e il costo computazione. L’uso della cross-validazione consente di calcolare la varianza dell’errore e aiuta a raggiungere il più piccolo errore desiderato con il minor numero di *features*.

Una volta che si è soddisfatti dei parametri impostati e dell’adattamento della retta stimata con la retta *target*, è possibile generare una Matlab *function* chiamata “*healthIndicator*” tramite il comando “*Generate Code*”. Tramite la funzione generata si può fornire in ingresso la *FeatureTable* ottenuta dalla funzione “*diagnosticFeature*” ed ottenere in uscita i valori degli indicatori di salute *HI* che variano da 1 a 0. Poiché nel nostro caso abbiamo effettuato dieci simulazioni, avremo in uscita dalla funzione dieci valori di *HI* che rappresentano lo stato di salute del nostro motore elettrico durante il suo periodo di funzionamento operativo.

I valori di *HI* ottenuti devono essere a questo punto associati all’istante che rappresentano, in quanto ogni determinato indicatore di condizione rappresenta lo stato di salute del motore elettrico all’istante *t* in cui sono stati campionati i valori di resistenza ed induttanza. Per far ciò si crea una nuova variabile di tipo “*timetable*”, in cui ad ogni valore di *HI* viene associato il valore di *t* a cui si ha campionato *R* ed *L*. Questo diventa quindi il nostro dataset di addestramento con cui nei passaggi successivi verrà allenato il modello *RUL*. Poiché, a questo punto, disponiamo di un solo insieme di dati di simulazione di funzionamento fino al guasto (*run-to-failure*) per allenare la *RUL*, si è deciso di perturbare il dataset di addestramento a nostra disposizione con una piccola quantità di rumore Gaussiano con varianza pari a 0.1, in modo da simulare un secondo insieme di dati di simulazione di funzionamento fino al guasto, per effettuare un addestramento del modello *RUL* più accurato.

In Figura 70 viene riportato il grafico dell’andamento dei valori degli indicatori di *HI* nel tempo di funzionamento *t*, in giorni, del motore elettrico. La curva blu rappresenta i valori di *HI* generati tramite la funzione “*healthIndicator*”, mentre la curva rossa rappresenta i valori di *HI* ottenuti perturbando con del rumore i primi.

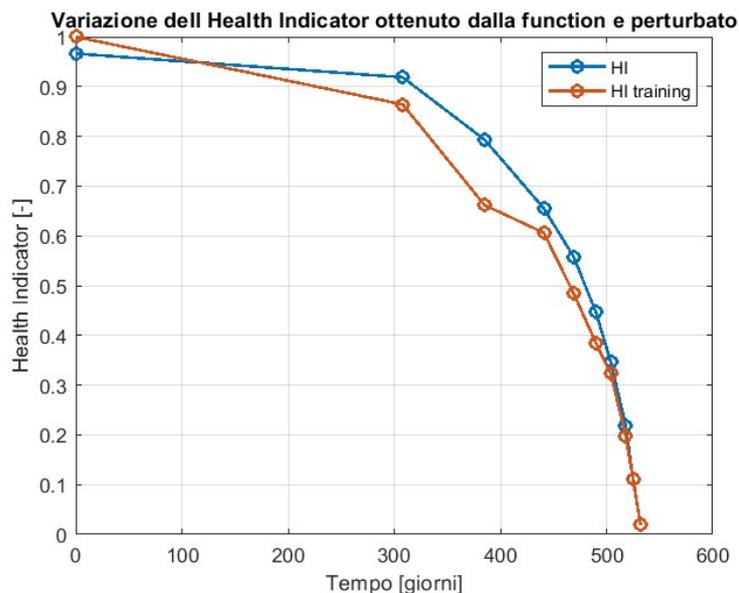


Figura 70 : Variazione degli indicatori di salute nel tempo t

4.3.5 Creazione del modello *RUL*

Essendosi ricavati i due insieme di dati di addestramento è ora possibile utilizzarli per allenare un modello di degradazione a nostra scelta. Viene, dunque, scelto il modello di degradazione lineare “*linearDegradationModel*” per stimare la vita utile rimanente (*RUL*) del componente, poiché, tramite tale modello di degrado, è possibile prevedere il momento in cui il segnale monitorato supererà una soglia predefinita, identificando quindi il raggiungimento dell’istante in cui effettuare la manutenzione. La scelta dell’utilizzo di questo modello è stata fatta in quanto esso risulta particolarmente utile quando si ha la presenza di un segnale monitorato che segue una scala logaritmica, rendendo la visualizzazione del degrado più facilmente rappresentabile attraverso un andamento lineare, oppure quando il componente in esame non subisce un degrado cumulativo, ma piuttosto un deterioramento progressivo, che risulta più approssimabile attraverso tale modello [24].

La configurazione del modello varia in base al tipo specifico di componente e può seguire due tipologie di approcci. La prima prevede la stima dei parametri iniziali del modello utilizzando dei dati storici relativi allo stato di salute di componenti o sistemi simili a quello in analisi, tramite l’utilizzo della funzione Matlab “*fit*”, la quale permette di adattare il modello ai dati di addestramento forniti [25]. Il secondo approccio prevede di impostare manualmente i parametri iniziali del modello nel momento in cui viene creato, basandosi sulla conoscenza del processo di degrado del componente; perciò, è necessario disporre di informazioni pregresse sull’andamento del degrado senza dover utilizzare dati storici. Nel nostro caso, non disponendo né di conoscenze pregresse né di database di dati storici, si è deciso di utilizzare il primo approccio sfruttando i due insiemi di dati creati come dati storici con cui allenare il modello. Dopo aver impostato tutti i parametri necessari al modello di

degrado, è possibile stimare la vita utile rimanente (*RUL*) del motore elettrico utilizzando la funzione Matlab "*predictRUL*", la quale fornisce una previsione della *RUL* basandosi sull'andamento del degrado osservato fino a quel momento [27].

Il modello di degradazione lineare implementa la seguente equazione di degradazione lineare continua nel tempo:

$$S(t) = \varphi + \theta(t) * t + \varepsilon(t)$$

I termini presenti in questa equazione sono l'intercetta del modello φ , che risulta essere sempre costante, la pendenza del modello $\theta(t)$, la quale viene modellata come una variabile casuale con una distribuzione normale con media *Theta* e varianza *ThetaVariance*, che sono due parametri del modello, e il rumore aggiunto $\varepsilon(t)$, il quale viene modellato come una distribuzione normale con media zero e varianza *NoiseVariance* (anch'esso un parametro del modello).

I parametri da impostare all'interno del modello sono diversi e vengono descritti di seguito:

- *Theta*: rappresenta il valor medio attuale del parametro di pendenza $\theta(t)$ nel modello di degradazione e viene indicato come uno scalare; esso viene impostato quando viene creato il modello.
- *ThetaVariance*: rappresenta la varianza corrente del parametro di pendenza $\theta(t)$ nel modello di degradazione e viene definito come uno scalare non negativo; anch'esso viene impostato durante la creazione del modello.
- *Phi*: indica il valore dell'intercetta corrente φ per il modello di degradazione e viene indicato come uno scalare.
- *NoiseVariance*: è la varianza del rumore aggiunto $\varepsilon(t)$ nel modello di degradazione e viene indicato come uno scalare non negativo.
- *SlopeDetectionLevel*: determina l'inizio del processo di degradazione, individuando quando la pendenza del modello supera un certo valore e viene indicato come uno scalare compreso in un intervallo tra 0 e 1; il suo valore di *default* è pari a 0.05.
- *LifeTimeVariable*: è una variabile di durata, che viene specificata come una stringa e quando i dati di addestramento sono sottoforma di *timetable*, come nel nostro caso, è pari al nome della variabile tempo contenuta all'interno dei dati.
- *LifeTimeUnit*: rappresenta l'unità di misura della variabile di durata e viene specificata come una stringa.
- *DataVariables*: indica il nome della variabile di degrado, che nel nostro caso è l'indicatore di salute *HI* e viene specificata come una stringa.

All'interno del nostro algoritmo questi parametri vengono impostati nel seguente modo, fornendo principalmente valori di *default*, a parte per il parametro *Theta* che viene posto pari all'inverso del numero di giorni di vita totale del componente, con il segno meno davanti per indicare una pendenza negativa.

<i>Theta</i>	-0.0012
<i>ThetaVariance</i>	1
<i>NoiseVariance</i>	1
<i>SlopeDetectioLevel</i>	0.1
<i>LifeTimeVariable</i>	"Time"
<i>LifeTimeUnit</i>	"days"
<i>DataVariables</i>	"HealthIndicator"

Una volta impostati tali parametri è possibile utilizzare la funzione Matlab "*fit*", la quale stima dei nuovi parametri del modello *RUL*, utilizzando i dati di addestramento che gli sono stati forniti e adattandoli al modello creato. I nuovi parametri sono i seguenti:

<i>Theta</i>	-0.0017
<i>ThetaVariance</i>	$1.9703 \cdot 10^{-9}$
<i>NoiseVariance</i>	0.0312
<i>SlopeDetectioLevel</i>	0.1
<i>Phi</i>	1.1513
<i>LifeTimeVariable</i>	"Time"
<i>LifeTimeUnit</i>	"days"
<i>DataVariables</i>	"HealthIndicator"

4.3.6 Stima della vita utile rimanente (*RUL*)

L'ultimo passaggio dell'algoritmo prevede la stima della vita utile residua del nostro motore elettrico rappresentata tramite un grafico ottenuto con una funzione Matlab generata da noi e definita come "*estimateRUL*".

Avendo ora il nostro modello di degrado allenato tramite dei dati sintetici che rappresentano diversi stati di guasto del nostro sistema, è possibile valutare la vita utile rimanente del motore elettrico per tutta la sua vita operativa, che va dall'istante zero al valore in cui si ha il *Fault Ratio D* massimo pari a settantasei settimane. Per effettuare una stima nel tempo si decide di campionare dei valori di resistenza e di induttanza ogni settimana e non più solo per i dieci valori iniziali.

Si inizializzano, dunque, le due nuove variabili R ed L sempre tramite le formule:

$$R = R_0 + D * R_0$$

$$L = L_0 + D * L_0$$

con D che rappresenterà il parametro di guasto campionato ogni settimana sul grafico rappresentato in Figura 49. Tali parametri vengono successivamente modificati, aggiungendo del rumore Gaussiano per imitare le incertezze del degrado nella vita reale. In questo abbiamo un nuovo set di parametri di resistenza ed induttanza degradati da poter fornire al nostro modello Matlab-Simulink e con cui ricavarci dei nuovi valori delle variabili in uscita del modello da poter sfruttare per la stima della RUL .

Per rappresentare graficamente la stima della vita utile residua si è utilizzata una funzione creata ad hoc, come sopra riportato, e chiamata “*estimateRUL*”. I parametri che sono stati forniti in ingresso alla funzione sono il modello di degradazione creato al passaggio precedente, il vettore tempo t , rappresentante la vita operativa del componente, che va da zero a settantasei settimane, equi spaziate di una settimana alla volta, e i nuovi parametri della resistenza R e dell’induttanza L perturbati con del rumore e campionati ogni settimana.

Il processo seguito all’interno della funzione è esattamente lo stesso che è stato svolto fino ad ora, almeno per la parte iniziale. Tramite i nuovi parametri di resistenza ed induttanza si effettuano non più dieci, ma settantasette simulazioni, da cui si ricavano sempre le stesse variabili indicate nel paragrafo 4.3.2. Tramite le nuove *timeseries* generate, composte sempre dalla prima colonna contenente il tempo e da altre settantasette colonne contenenti i valori delle variabili campionate all’aumento del degrado, si genera la *Tabella* nella stessa maniera riportata precedentemente.

A questo punto, si effettua una modifica all’algoritmo. Viene creato un ciclo “*for*” all’interno del codice e, per ogni simulazione, alla funzione generata in precedenza “*diagnosticFeature*” non viene più fornita l’intera *Tabella*, ma solamente la riga di questa contenente tutte le variabili corrispondenti alla simulazione in analisi. In questo modo, la funzione ci fornirà in uscita una *FeatureTable* contenente le *features* corrispondenti a tale simulazione e quindi ad un determinato stato di salute, partendo ovviamente da uno sano fino ad arrivare ad uno stato guasto. Si procede, dunque, alla trasformazione delle caratteristiche estratte in un *Condition Indicator* corrispondente all’indicatore di salute HI per quel determinato istante. Tale indicatore viene utilizzato per aggiornare il modello di degradazione lineare precedentemente creato tramite la funzione Matlab “*update*”, la quale aggiorna la stima dei parametri del modello di degrado, utilizzando le misurazioni più recenti fornitegli [26]. In particolare, per il nostro modello vengono aggiornati i parametri Θ e Θ Variance.

Aggiornato il modello con i nuovi parametri. È possibile stimare la vita utile rimanente del componente in quel determinato istante per quei determinati valori di resistenza ed induttanza degradati tramite la funzione Matlab “*predicRUL*”. Tale funzione stima la RUL di un componente dato un modello di stima e avendo informazioni sul suo tempo di utilizzo e

sul suo profilo di degrado. Inoltre, gli viene fornito in ingresso anche un termine aggiuntivo corrispondente al livello di confidenza ‘Alpha’, imposto pari a 0.1. Tale valore viene fornito poiché la funzione calcola tramite questo l’intervallo di confidenza all’interno del quale è contenuta la stima della nostra vita utile rimanente. L’intervallo di confidenza viene stimato come una regione compresa tra $Alpha/2$ e $(1-Alpha/2)$ [27]. I termini in uscita dalla funzione sono la vita utile rimanente del componente stimata, sottoforma di scalare e con unità pari a quella fornita nel modello di degrado, e l’intervallo di confidenza ad esso associato. Questi due valori vengono graficati iterativamente per ogni ciclo sul nostro grafico finale rappresentante la *RUL* del componente, per cui ad ogni iterazione avremo una stima della vita utile con i parametri degradati forniti in ingresso. Su tale grafico viene anche riportato lo “*Slope detection time*”, il quale rappresenta l’istante in cui viene rilevata una pendenza significativa per cui si ha l’inizio del processo di degradazione.

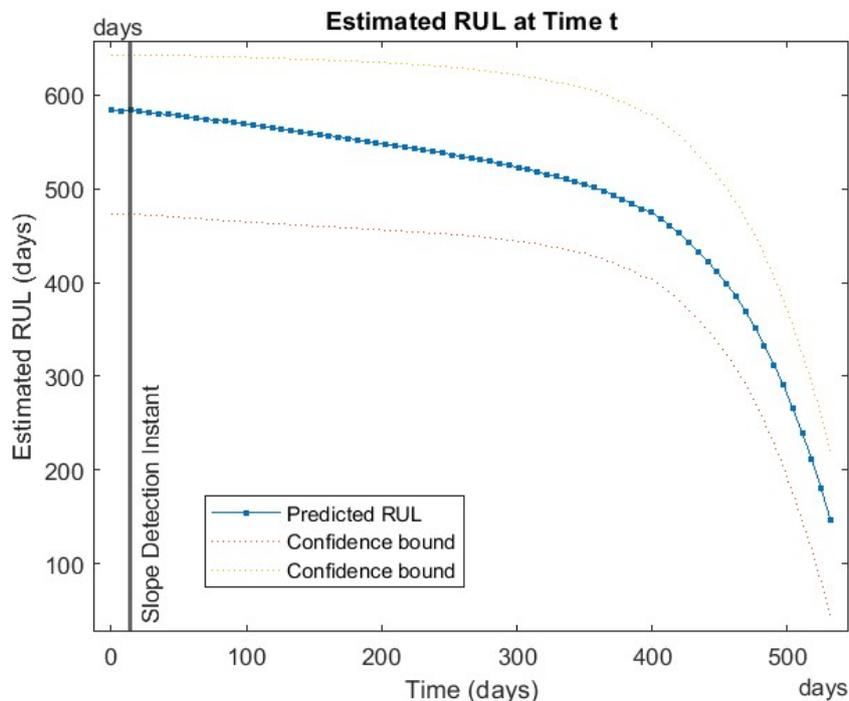


Figura 71 : Stima della vita utile rimanente del componente per ogni settimana

L’andamento della *RUL* prevede una diminuzione lineare costante di questa per i primi circa 350 giorni, per poi subire un brusco calo fino alla fine della sua vita operativa imposta. Ciò è in linea con la legge di degrado con cui si è stimato il *Fault Ratio D* e con cui si sono fatti degradare i parametri di resistenza ed induttanza del motore elettrico. Questa legge prevedeva, infatti, un primo tratto in cui *D* rimaneva circa costante, per poi subire una crescita esponenziale nell’ultimo nelle settimane finali.

Si è graficato, in seguito, anche l'andamento degli indicatori di salute HI calcolati ad ogni iterazione per monitorare la loro variazione, la quale è riportata in Figura 72.

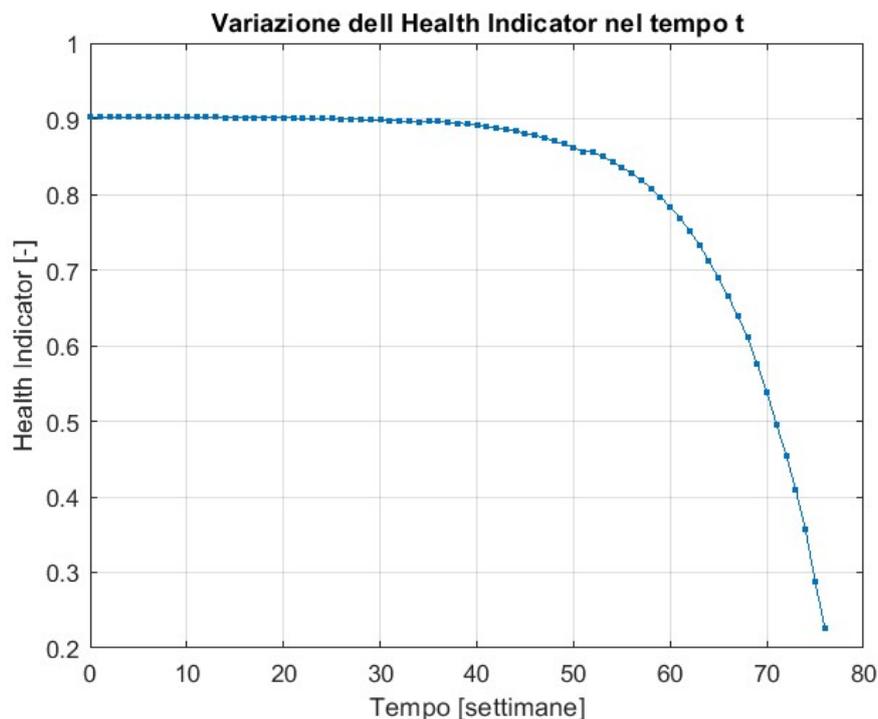


Figura 72: Variazione degli indicatori di salute HI per ogni iterazione

Come si può notare l'andamento segue circa lo stesso del grafico della stima della RUL , in quanto quest'ultimo è basato proprio sul valore dell'indicatore di condizione in quell'istante. Si può notare come HI non parta dal valore di 1 e non raggiunga il valore di 0. Ciò è dovuto al fatto che i dati forniti per la stima della RUL sono diversi da quelli utilizzati per sviluppare l'algoritmo e per questo potrebbero non rispecchiare lo stato di salute completamente sano, come anche quello di completamente guasto. Poiché l' HI raggiunge un minimo valore di circa 0.2, allora anche la RUL stimata non raggiungerà il valore di 0, ma si fermerà ad una condizione in cui i giorni di vita utile operativa sono ancora pari a circa 100, in quanto non ci troviamo ancora nella condizione di guasto. Se si conoscesse un valore soglia limite oltre cui non si vuole andare, sapendo che i componenti raggiungerebbero una condizione troppo critica, allora sul grafico della RUL verrebbe indicato dopo quanti giorni tale valore sarebbe raggiunto e sarebbe necessario effettuare una manutenzione.

Per verificare la correttezza del processo, i nuovi parametri degradati di resistenza ed induttanza da fornire alla funzione "estimateRUL", sono stati fatti variare, in modo da crearsi dieci dataset da fornire alla funzione con cui stimare ogni volta la vita utile rimanente del sistema con i nuovi parametri di ingresso. Queste stime sono state poi riportate su uno

stesso grafico per verificare che l'andamento della *RUL* fosse sempre circa lo stesso, come mostrato in Figura 73.

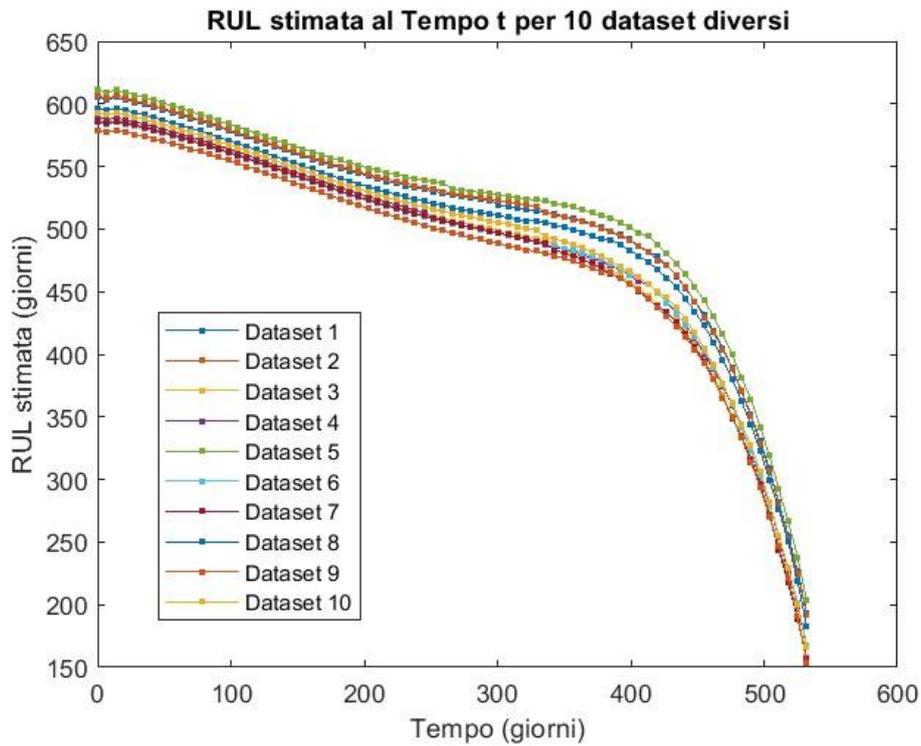


Figura 73 : Stima della RUL per dieci dataset diversi di parametri *R* ed *L*

Come dimostra la figura sopra riportata l'andamento seguito è sempre circa lo stesso con alcune stime che partono da un numero di giorni di vita utile e finiscono una vita utile residua maggiore ad altri, in base a come sono stati generati randomicamente i valori di resistenza *R* ed induttanza *L* tramite la perturbazione del rumore Gaussiano.

5 CONCLUSIONI

Il presente lavoro di tesi ha affrontato il tema della Manutenzione Predittiva nel settore aeronautico, con particolare attenzione allo sviluppo di un algoritmo per la stima della *Remaining Useful Life (RUL)* di un motore elettrico all'interno di un attuatore elettromeccanico (EMA). Come ribadito anche in precedenza, l'obiettivo del lavoro non era esclusivamente ottenere risultati concreti e immediatamente applicabili, ma piuttosto sviluppare una metodologia che venga strutturata per applicare la prognostica a diversi sistemi e che possa rappresentare un punto di partenza per ulteriori ricerche e miglioramenti nel campo della Manutenzione Predittiva.

Inizialmente, nella tesi si è introdotto il concetto di prognostica, illustrando il suo ruolo sempre più rilevante nel settore aerospaziale e i vantaggi che essa può apportare in termini di sicurezza, riduzione dei costi e ottimizzazione degli interventi di manutenzione. Successivamente, si è analizzato come la prognostica consenta di passare da un approccio basato su interventi di manutenzione programmata ad un approccio maggiormente dinamico, in cui la manutenzione viene eseguita in base allo stato attuale del componente.

Nel successivo capitolo, è stato esaminato lo stato dell'arte della prognostica, introducendo i modelli utilizzati per la previsione della vita residua dei componenti, tra i quali i modelli basati sui dati, che sfruttano tecniche di machine learning per individuare dei pattern di degrado, i modelli fisici, che si basano su equazioni matematiche per descrivere il comportamento del sistema, e i modelli ibridi, che combinano entrambe le metodologie per migliorare l'accuratezza delle previsioni. Inoltre, si sono evidenziati anche i principali vantaggi e le principali sfide di ciascuno di questi approcci, riportando i limiti legati alla raccolta e alla gestione di grandi quantità di dati e la necessità di sviluppare dei modelli sempre più affidabili e adeguati alle specifiche applicazioni aeronautiche.

Il lavoro si è, poi, concentrato sulla modellazione dell'attuatore elettromeccanico utilizzato nei sistemi di controllo di volo di un velivolo regionale, basandosi su una semplificazione del modello descritto all'interno del documento "*Simulation of an All-Electric Flight Control System for the Evaluation of Power Consumption*" [1]. Il modello proposto rappresenta un *Flight Control System (FCS)*, in cui gli attuatori idraulici tradizionali vengono sostituiti dagli attuatori elettromeccanici, riducendo il peso e migliorando l'efficienza energetica del velivolo. In particolare, per il nostro modello ci si è concentrati solo sull'EMA impiegato per il comando delle superfici di volo, il quale è composto da un motore sincrono *brushless* a tre fasi accoppiato con una trasmissione meccanica per la trasmissione del movimento. Tale modello è stato sviluppato tramite l'utilizzo del software Matlab-Simulink ed è stato sfruttato nelle fasi preliminari dello sviluppo dell'algoritmo per effettuare le simulazioni con cui generare i dati sintetici.

L'attenzione è stata posta maggiormente sul degrado del motore elettrico ed in particolare sui parametri di resistenza ed induttanza, i quali sono risultati due parametri critici per il monitoraggio dello stato di salute del sistema. Il loro degrado può, infatti, influenzare significativamente le prestazioni del motore, causando un aumento delle perdite energetiche

e una riduzione dell'efficienza complessiva dell'attuatore. Si è proceduto, dunque, alla generazione di dati sintetici degradati, basandosi su equazioni tratte dalla letteratura e modelli di degrado approssimati. In particolare, il degrado della resistenza e dell'induttanza è stato fatto variare progressivamente nel tempo, in modo da generare dei parametri in uscita dal modello che potessero replicare un comportamento realistico nel tempo. Il codice Matlab, che è stato sviluppato, ha permesso di elaborare questi dati, utilizzando degli specifici Toolbox forniti dal software per analizzare il comportamento del sistema in diverse condizioni operative. Grazie al "*Diagnostic Feature Designer*" sono state estratte le caratteristiche più importanti per il monitoraggio del degrado e, tramite il "*Health Indicator Designer*", sono state trasformate in degli indicatori di salute, che hanno poi permesso di creare un modello di degradazione con cui prevedere la *RUL*.

L'analisi condotta ha mostrato la grande potenzialità della Manutenzione Predittiva nell'ottimizzazione della gestione dei sistemi avionici e non solo, evidenziando come, grazie all'utilizzo di alcuni software, sia possibile sviluppare degli algoritmi capaci di individuare preventivamente il degrado dei componenti. In particolare, tramite l'utilizzo dei Toolbox è stato possibile definire degli indicatori riguardanti lo stato di salute del motore elettrico, correlando il degrado dei parametri con delle variazioni nelle prestazioni complessive del sistema. Tuttavia, il lavoro proposto presenta alcune limitazioni che devono essere affrontate per migliorare l'affidabilità e l'applicabilità dei risultati. La principale tra queste riguarda l'utilizzo di dati sintetici, che, pur essendo basati su delle assunzioni ragionevoli tratte dalla letteratura, non sempre riflettono accuratamente il comportamento reale del motore nelle sue condizioni operative. Inoltre, anche le semplificazioni adottate nel modello di riferimento potrebbero influire sulla precisione delle previsioni ottenute, come anche l'utilizzo dei parametri di degrado di resistenza ed induttanza, i quali solitamente non sono gli indicatori principali con cui viene monitorato un motore elettrico. Per superare tali limiti, sarebbe necessario validare il nostro algoritmo tramite l'utilizzo di dati sperimentali reali, raccolti tramite dei test su dei componenti fisici operanti in ambienti di lavoro aeronautico. Inoltre, anche l'integrazione di sensori per il monitoraggio in tempo reale delle condizioni operative del motore elettrico potrebbe fornire dei dati più accurati, consentendo all'algoritmo di migliorare la sua capacità di adattarsi a diverse condizioni di degrado e di incrementare la sua affidabilità.

Un ulteriore miglioramento potrebbe riguardare l'integrazione di tecniche avanzate di intelligenza artificiale e machine learning e nello specifico di *Reti Neurali* e di algoritmi di apprendimento automatico supervisionato e non-supervisionato. Questi strumenti potrebbero consentire una modellazione più accurata delle relazioni non lineari presenti tra i parametri di degrado e la stima della *RUL*, migliorando la capacità del sistema di adattarsi a diverse condizioni operative. Inoltre, l'utilizzo di metodi come le *Support Vector Machines* (SVM) e gli *alberi decisionali* potrebbe fornire ulteriori strumenti per la classificazione dello stato di salute dei componenti, contribuendo a una manutenzione sempre più efficace ed affidabile.

In conclusione, il lavoro svolto rappresenta un primo approccio per lo sviluppo di un sistema prognostico applicabile agli attuatori elettromeccanici nel settore aeronautico e non solo. Nonostante i risultati ottenuti non siano definitivi, essi forniscono, comunque, una buona

base metodologica per futuri approfondimenti e miglioramenti. L'approccio utilizzato ha permesso di esplorare il potenziale della Manutenzione Predittiva, dimostrando come sia possibile stimare la vita utile rimanente (*RUL*) di un componente attraverso l'analisi di parametri chiave.

L'evoluzione della manutenzione predittiva, supportata dall'innovazione tecnologica e dall'integrazione di modelli sempre più avanzati, potrà contribuire significativamente alla sicurezza e all'efficienza operativa dei velivoli. Un sistema di diagnostica e prognostica ben sviluppato non solo ridurrà i costi di manutenzione e i tempi di fermo, ma aumenterà anche l'affidabilità complessiva dei sistemi di bordo, garantendo una maggiore disponibilità e sicurezza di questi. Questo lavoro si pone, quindi, come un punto di partenza per futuri sviluppi nella Manutenzione Predittiva con l'obiettivo di rendere i sistemi aeronautici più efficienti, affidabili e sostenibili.

Bibliografia:

- [1] Schettini F. *, Denti E. *, Di Rito G. * and Galatolo R. : “Simulation of an All-Electric Flight Control System for the Evaluation of Power Consumption”
- [2] <https://en.wikipedia.org/wiki/Prognostics#:~:text=Prognostics%20is%20an%20engineering%20discipline,used%20to%20meet%20desired%20performance.>
- [3] George Vachtsevanos, Frank Lewis, Michael Roemer, Andrew Hess and Biqing Wu: “Intelligent Fault Diagnosis And Prognosis For Engineering Systems”
- [4] Raffaele Giordano, Master’s Thesis: “Development of a digital twin for the validation of prognostic algorithms for electromechanical actuators”
- [5] <https://www.cfisteph.com/components-controls>
- [6] Mirko Mazzoleni, Fabio Previdi, Matteo Scandella, Giulio Pispola : “Experimental Development of a Health Monitoring Method for Electro-Mechanical Actuators of Flight Control Primary Surfaces in More Electric Aircrafts”
- [7] Shuai Fu * and Nicolas P. Avdelidis : “Review Prognostic and Health Management of Critical Aircraft Systems and Components: An Overview “
- [8] Yu, Y.; Yao, H.; Liu, Y. : “Aircraft Dynamics Simulation Using a Novel Physics-Based Learning Method.” *Aerosp. Sci. Technol.* 2019, 87, 254–264
- [9] Coppe, A.; Pais, M.J.; Haftka, R.T.; Kim, N.H. : “Equivalent Damage Growth Parameters Using a Simplified Model.” In *Proceedings of the ASME Design Engineering Technical Conference, Washington, DC, USA, 28–31 August 2011; Volume 2, pp. 1317–1328*
- [10] Liao, L.; Köttig, F. : “Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction.” *IEEE Trans. Reliab.* 2014, 63, 191–207
- [11] Grewal, M.S.; Kain, J. : “Kalman Filter Implementation with Improved Numerical Properties.” *IEEE Trans. Automat. Control* 2010, 55, 2058–2068
- [12] Liao, L.; Köttig, F. : “A Hybrid Framework Combining Data-Driven and Model-Based Methods for System Remaining Useful Life Prediction.” *Appl. Soft Comput.* 2016, 44, 191–199
- [13] Gianluca Parisi, Master’s Thesis: “Modellazione multibody su ADAMS di vite a ricircolo di sfere per identificazione modalità di guasto progressivo per finalità di prognostica di attuatori elettromeccanici per comandi primari di volo”
- [14] https://it.wikipedia.org/wiki/Potenza_elettrica
- [15] <https://it.mathworks.com/discovery/predictive-maintenance.html>
- [16] <https://it.mathworks.com/content/dam/mathworks/ebook/gated/predictive-maintenance-ebook-all-chapters.pdf>
- [17] <https://it.mathworks.com/content/dam/mathworks/ebook/gated/predictive-maintenance-ebook-part3-estimating-rul-with-matlab.pdf>

[18] <https://it.mathworks.com/help/predmaint/ug/rul-estimation-using-rul-estimator-models.html>

[19] Levent U. Gokdere, Alexander Bogdanov, Stephen L. Chiu, Kirby J. Keller, John Vian: “Adaptive Control of Actuator Lifetime “

[20] Andrea De Martin, Giovanni Jacazio, and George Vachtsevanos: “Windings Fault Detection and Prognosis in Electro-Mechanical Flight Control Actuators Operating in Active-Active Configuration “

[21] <https://it.mathworks.com/help/predmaint/ref/diagnosticfeaturedesigner-app.html>

[22] <https://it.mathworks.com/help/predmaint/ug/signal-features.html>

[23] <https://it.mathworks.com/help/predmaint/ref/healthindicatordesigner-app.html>

[24] <https://it.mathworks.com/help/predmaint/ref/lineardegradationmodel.html>

[25] <https://it.mathworks.com/help/predmaint/ref/lineardegradationmodel.fit.html>

[26] <https://it.mathworks.com/help/predmaint/ref/lineardegradationmodel.update.html>

[27] <https://it.mathworks.com/help/predmaint/ref/lineardegradationmodel.predictrul.html>

[28] <https://it.mathworks.com/help/predmaint/ug/rul-estimation-of-dc-link-capacitor-in-power-converter.html>