

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCATRONICA

TESI DI LAUREA MAGISTRALE

Caratterizzazione sperimentale delle relazioni
corrente-flusso magnetico concatenato di motori sincroni
per la simulazione

Relatore: Prof. Fabio Tinazzi

Laureando: Luca Magrin
2061970

ANNO ACCADEMICO: 2023-2024

SOMMARIO

Il lavoro di tesi è incentrato sulla caratterizzazione sperimentale delle relazioni non lineari tra corrente e flusso magnetico concatenato in motori sincroni a magneti permanenti anisotropi. L'obiettivo principale è sviluppare una metodologia efficace per ottenere le mappe di flusso del motore, necessarie per migliorare l'efficienza energetica. Dopo una fase teorica, in cui viene analizzato il comportamento del motore e della tecnica che permette di ricavarsi il flusso, si procede con un approccio sperimentale. Utilizzando un banco prova, vengono raccolti i dati per costruire le mappe di flusso utilizzando e implementando la tecnica studiata precedentemente. Successivamente, tali mappe vengono invertite attraverso un'interpolazione, permettendo di stabilire le relazioni inverse corrente-flusso, per poi implementarle attraverso un lookup table nell'azionamento già esistente nel banco prova. Infine si osserva l'effetto che queste relazioni hanno portato al controllo, attraverso un processo iterativo che richiedeva una sincronizzazione tra banco prova e acquisitore esterno che misura il rendimento del motore in diverse condizioni operative, così da creare una mappa di rendimento.

INDICE

1	Introduzione	1
2	Motori sincroni	3
2.1	Struttura e principio di funzionamento	3
2.2	Modello matematico	3
2.3	Tecniche di controllo MTPA	6
2.4	Modello magnetico del motore	7
3	Mappatura	11
3.1	Obbiettivo della tesi	11
3.1.1	Parte teorica	11
3.1.2	Procedura teorica	12
3.1.3	Parte sperimentale	14
4	Risultati ottenuti	27
4.0.1	Risultati ottenuti dalla simulazione	27
4.0.2	Risultati ottenuti dalla prova sperimentale	31
4.0.3	Inversione mappe di flusso	34
5	Misura del rendimento	39
5.0.1	Parte teorica	39
5.0.2	Procedura teorica	40
5.0.3	Parte sperimentale	40
	Conclusioni	49
	Appendice	
A	Programmi	53

ELENCO DELLE FIGURE

Figura 1	Rappresentazione schematica di un motore anisotropo.[1]	4
Figura 2	Rappresentazione schema a blocchi di un motore anisotropo. [1]	5
Figura 3	Curva MTPA per coppia positiva.[1]	6
Figura 4	Andamento reale di $\lambda_d = f_1(i_d, i_q)e\lambda_q = f_2(i_d, i_q)$.[1]	8
Figura 5	Andamento dei riferimenti i_{dq} .[2]	13
Figura 6	Mappa di lavoro.[2]	14
Figura 7	Flow Chart del programma.	15
Figura 8	Flow chart degli stati operativi.	17
Figura 9	Mappa dei punti con raggio 2,5 A, con 10 intervalli lungo d e q.	19
Figura 10	Relazioni dei flussi nel motore simulato	27
Figura 11	Mappa con i punti di lavoro della simulazione.	28
Figura 12	Andamento dei riferimenti dei punti di lavoro.	28
Figura 13	Confronto tra la stima lungo d e il calcolo.	29
Figura 14	Confronto tra la stima lungo q e il calcolo.	29
Figura 15	Errore tra la stima e il calcolo lungo d.	30
Figura 16	Errore tra la stima e il calcolo lungo q.	30
Figura 17	Mappa sperimentale diretta di λ_d .	31
Figura 18	Mappa sperimentale diretta di λ_q .	32
Figura 19	Interpolazione con scatteredInterpolant.	33
Figura 20	Interpolazione con griddatan.	33
Figura 21	Adattamento di curve con fit.	34
Figura 22	Presenza di non linearità lungo d.	35
Figura 23	Presenza di non linearità lungo q.	35
Figura 24	Coppie di flusso ricavate dall'esperimento.	36
Figura 25	Relazione inversa $i_d = h(\lambda_d, \lambda_q)$.	36
Figura 26	Relazione inversa $i_q = k(\lambda_d, \lambda_q)$.	37
Figura 27	Andamento dei riferimenti di coppia e velocità.	43
Figura 28	Schema di cablaggio.	44
Figura 29	Relazione lineare per scalare il valore del sensore.	46
Figura 30	Mappa di efficienza.	46
Figura 31	Mappa di efficienza con punti campionati.	47

ELENCO DELLE TABELLE

Tabella 1	Relazione tra i punti presenti nella mappa.	20
-----------	---	----

CITAZIONI

- 1 Appunti del professor Mauro Zigliotto
- 2 Eric Armando, Radu Iustin Bojoi, Senior Member, IEEE, Paolo Guglielmi, Member, IEEE, Gianmario Pellegrino, Member, IEEE, and Michele Pastorelli, "Experimental Identification of the Magnetic Model of Synchronous Machines", IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, VOL. 49, NO. 5, SEPTEMBER/OCTOBER 2013

INTRODUZIONE

Oggi, l'energia elettrica rappresenta una risorsa fondamentale per lo sviluppo e il funzionamento della società moderna. Circa la metà dell'energia elettrica prodotta viene convertita in energia meccanica attraverso l'uso di motori elettrici, sottolineando così l'importanza di questi dispositivi nel panorama industriale e civile. Tuttavia, la produzione di elettricità ha ancora un impatto ambientale significativo, poiché le fonti rinnovabili, pur in crescita, non sono ancora in grado di soddisfare l'intera domanda globale. Questo contribuisce all'aumento dell'effetto serra e all'uso intensivo di risorse non rinnovabili.

In quest'ottica, l'efficienza energetica diventa un fattore cruciale. Migliorare l'efficienza dei sistemi di conversione elettromeccanica, ovvero l'integrazione tra motore elettrico e azionamento, può contribuire a ridurre il fabbisogno energetico complessivo e, di conseguenza, l'impatto ambientale. Un sistema più efficiente consuma meno energia per svolgere lo stesso lavoro, riducendo così sia le emissioni sia l'uso di risorse energetiche.

Negli ultimi anni, lo studio e lo sviluppo dei motori sincroni hanno acquisito un'importanza crescente. Rispetto ai tradizionali motori a induzione o ai motori sincroni a magneti permanenti, i motori sincroni, specialmente quelli a riluttanza, offrono numerosi vantaggi: sono in grado di fornire elevate efficienze e una maggiore densità di coppia, risultando quindi particolarmente adatti a contesti in cui l'efficienza e la sostenibilità sono essenziali. Un altro aspetto rilevante riguarda la composizione di questi motori, basata su materiali ferromagnetici facilmente reperibili. Questo li rende particolarmente competitivi rispetto ai motori a magneti permanenti, che richiedono l'utilizzo di terre rare, soggette a fluttuazioni di prezzo e con un impatto ambientale significativo durante l'estrazione.

Tuttavia, l'impiego di motori a riluttanza non è privo di sfide tecniche. Uno dei principali problemi è rappresentato dalla saturazione magnetica del materiale ferromagnetico presente nel rotore. Quando si cerca di ottenere coppie elevate, il nucleo magnetico del rotore tende a saturarsi, causando una relazione non lineare tra flusso magnetico e corrente. Questa non linearità introduce complessità nell'azionamento del motore, rendendo difficile ottimizzarne il funzionamento senza una comprensione precisa del comportamento elettromagnetico.

Per affrontare questa criticità, è fondamentale sviluppare metodi di controllo e modellazione in grado di gestire accuratamente la non linearità tra flusso e corrente, garantendo così l'efficienza ottimale del sistema in qualsiasi condizione operativa. In questa tesi, il focus è stato posto sulla progettazione e realizzazione di un metodo

sperimentale per determinare queste relazioni non lineari, utilizzando un banco prova per raccogliere i dati necessari. I risultati ottenuti sono stati successivamente utilizzati per valutare l'effettiva efficienza del sistema di azionamento tramite misure di potenza, con l'ausilio di un wattmetro.

Questo lavoro mira a fornire un contributo alla comprensione e al miglioramento delle prestazioni dei motori sincroni, rendendoli sempre più competitivi dal punto di vista dell'efficienza energetica e della sostenibilità ambientale.

MOTORI SINCRONI

2.1 STRUTTURA E PRINCIPIO DI FUNZIONAMENTO

I motori sincroni a magneti permanenti anisotropi detti IPM (Intern Permanent Magnet) sfruttano la coesistenza di due principi di funzionamento per la generazione di coppia, quello a riluttanza e quello elettrodinamico. La coppia che si genera equivale alla somma dei due principi.

La struttura di questi motori è composta dallo statore e dal rotore, dove allo statore sono presenti tre avvolgimenti tali che se alimentati da una terna di tensioni sfasate tra di loro di $\frac{2}{3}\pi$ generano un flusso rotante al traferro. Il rotore è costituito da una struttura anisotropa dove nei fori dell'asse diretto sono presenti i magneti permanenti i quali possiedono la permeabilità dell'aria e quindi mantengono l'anisotropia rotorica. Dalla figura 1 è possibile osservare che la struttura appena descritta, dove per convenzione l'asse diretto viene posto dove la riluttanza è maggiore ovvero dove sono presenti i magneti permanenti.

2.2 MODELLO MATEMATICO

Per rappresentare attraverso un modello matematico il motore sincrono si parte dal bilancio di tensione

$$\begin{aligned} u_a &= Ri_a + \frac{d\lambda_a}{dt} \\ u_b &= Ri_b + \frac{d\lambda_b}{dt} \\ u_c &= Ri_c + \frac{d\lambda_c}{dt} \end{aligned} \quad (1)$$

dove:

- u_a, u_b, u_c tensioni applicate rispettivamente alla fase a, b, c
- i_a, i_b, i_c corrente che scorre rispettivamente nella fase a, b, c
- $\lambda_a, \lambda_b, \lambda_c$ flusso concatenato totale rispettivamente dalla fase a, b, c

il flusso concatenato totale di ogni fase può essere scritto come composizioni di altri termini. Si prenda in considerazione come esempio λ_a e troviamo che:

$$\lambda_a = \lambda_{a,mg} + L_a i_a + M_{ab} i_b + M_{ac} i_c \quad (2)$$

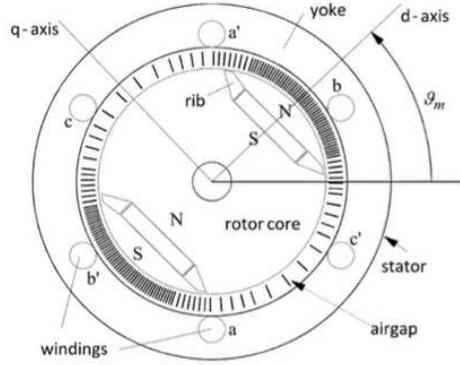


Figura 1: Rappresentazione schematica di un motore anisotropo.[1]

dove:

- $\lambda_{a,mg}$ è il flusso prodotto dai PM (Permanent Magnet) che si concatena con la fase a quindi si può riscrivere come $\lambda_{a,mg} = \Lambda_{mg} \cos(\theta_{me})$ dove θ_{me} è l'angolo che si crea tra l'asse della fase a e l'asse di percorrenza dei PM
- $L_a i_a$ è il flusso autoconcatenato dalla fase ovvero quel flusso che investe le spire prodotto dalla corrente stessa che ci scorre, questo flusso è composto da una parte costante e una variabile quindi riscrivo $L_a i_a$ come $L_a i_a = [(L_\sigma + L_o) - L_2 \cos(2\theta_{me})] i_a$ dove:
 - $L_a i_a$ è il flusso prodotto dalla fase a che si richiude su se stesso non passando per il rotore per cui non genera coppia
 - $L_o i_a$ è il flusso generato da fase a che passa interamente per il rotore e quindi genera coppia
 - $L_2 \cos(2\theta_{me}) i_a$ è quel termine che regola la variazione di induttanza al variare della posizione del rotore rispetto alla fase a
- $M_{ab} i_b, M_{ac} i_c$ flusso concatenato dalla fase a però prodotto sia dalla fase b che dalla fase c

A questo punto sostituisco in 1 la 2 così ottengo:

$$\begin{aligned}
 u_a &= R i_a + L_a \frac{di_a}{dt} + M_{ab} \frac{di_b}{dt} + M_{ac} \frac{di_c}{dt} + i_a \frac{dL_a}{dt} + i_b \frac{dM_{ab}}{dt} + i_c \frac{dM_{ac}}{dt} + e_a \\
 u_b &= R i_b + M_{ab} \frac{di_a}{dt} + L_b \frac{di_b}{dt} + M_{bc} \frac{di_c}{dt} + i_a \frac{dM_{ab}}{dt} + i_b \frac{dL_b}{dt} + i_c \frac{dM_{bc}}{dt} + e_b \\
 u_c &= R i_c + M_{ac} \frac{di_a}{dt} + M_{bc} \frac{di_b}{dt} + L_c \frac{di_c}{dt} + i_a \frac{dM_{ac}}{dt} + i_b \frac{dM_{bc}}{dt} + i_c \frac{dL_c}{dt} + e_c
 \end{aligned} \tag{3}$$

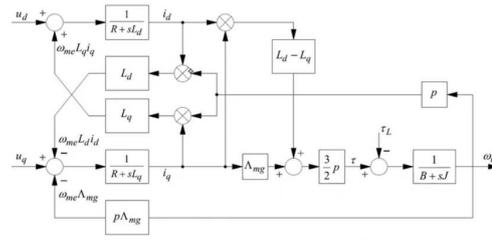


Figura 2: Rappresentazione schema a blocchi di un motore anisotropo. [1]

Ottenute queste tre nuove equazioni che non possono essere combinate e trasformate in un vettore spaziale in quanto L_a, L_b, L_c sono diverse a seconda della posizione in cui si trova il rotore per cui attraverso la modalità matriciale e il cambio di riferimento che passa da riferimento stazionario abc ad uno solidale al rotore dq . Ottengo quindi una nuova relazione matriciale:

$$u_{dqo} = R i_{dqo} + L_{dqo} \frac{d i_{dqo}}{dt} + T \frac{dT^{-1}}{dt} L_{dqo} i_{dqo} + e_{dqo} \quad (4)$$

dove T è la matrice di cambio riferimento, si riscrive la 4 in tre nuove equazioni:

$$\begin{aligned} u_d &= R i_d + L_d \frac{d i_d}{dt} - \omega_{me} L_q i_q \\ u_q &= R i_q + L_q \frac{d i_q}{dt} + \omega_{me} L_d i_d + \omega_{me} \Lambda_{mg} \\ u_o &= R i_o + L_\sigma \frac{d i_o}{dt} \end{aligned} \quad (5)$$

Troviamo la coppia attraverso il bilancio energetico di 5 e otteniamo quindi:

$$\tau = \frac{3}{2} p [\lambda_{mg} i_q + (L_d - L_q) i_d i_q] \quad (6)$$

si può vedere come la coppia è composta da due termini, il primo ovvero la parte magnetica $\frac{3}{2} p \lambda_{mg} i_q$ mentre il secondo dovuto alla differenza di riluttanza $\frac{3}{2} p (L_d - L_q) i_d i_q$. Questo ne comporta l'aumento del rapporto $\frac{\tau}{i}$ rispetto ai motori a magneti permanenti in quanto si aggiunge la componente dovuta alla riluttanza, inoltre risulta anche più economico in quanto servono meno magneti permanenti per raggiungere la stessa coppia proprio per il fatto che la differenza viene sopperita dal delta di riluttanza. Attraverso la conoscenza della 5 e 6 possiamo rappresentare lo schema a blocchi ?? del motore sincrono a magneti permanenti anisotropo che ne rappresenta la dinamica

Questo permette di capire durante le simulazioni del motore qual è la dinamica che si genera nel momento in cui applico una certa u_{dq} , si può vedere come la conoscenza delle caratteristiche L_{dq} sono fondamentali per avere una corretta conoscenza di ω_m .

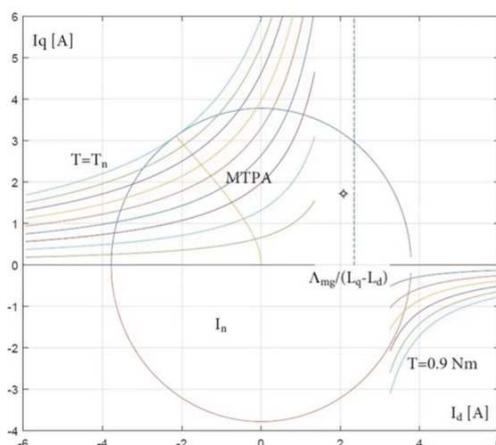


Figura 3: Curva MTPA per coppia positiva.[1]

2.3 TECNICHE DI CONTROLLO MTPA

Per controllare un motore sincrono gli anelli di controllo utilizzati in generale sono sempre quelli di velocità e di corrente. L'anello di velocità in base all'errore genera un modulo di corrente I che a sua volta dovrà essere scomposto in i_d e i_q . Osservando la formula 6 possiamo vedere come per ottenere una certa coppia τ possiamo avere infinite combinazioni di i_{dq} ma solo una coppia possiede il modulo minimo per ottenere una certa coppia. Questa tecnica viene detta MTPA ovvero ottenere la massima coppia a parità di corrente, quindi è fondamentale conoscere per ogni coppia possibile la corrispettiva coppia di correnti i_{dq} creando così una curva formata da sole coppie MTPA da come possiamo vedere in figura 3.

Per ottenere questa curva ci sono due principali tecniche che possono essere utilizzate: MTPA-offline o MTPA-online. La prima consiste nel ricavare la curva attraverso un banco prova per poi applicarla in un secondo momento ad un generatore di riferimento. L'idea è quella di collegare tra di loro due motori di cui uno viene comandato in velocità per comportarsi da carico e mantenere la velocità mentre l'altro è comandato in corrente. Al MUT (motor under test) viene applicato un modulo costante di corrente e fatto variare l'angolo, per ogni coppia di correnti i_{dq} misuro la coppia all'albero attraverso un torsionometro e quella con il τ maggiore è il punto di MTPA. Questa operazione viene svolta per ogni modulo di corrente da zero alla nominale. La tecnica MTPA-online svolge questa operazione senza l'utilizzo di componenti ulteriori ma sfruttando la tecnica perturb and observ ovvero perturbare il sistema con un disturbo in ingresso sovrapposto al riferimento ed osservare l'andamento dell'uscita. In questo caso perturbiamo la fase del vettore di corrente fin tanto che non troviamo il punto di MTPA, il processo è fattibile anche con il carico applicato in quanto il disturbo risulta quasi impercettibile al carico. Entrambi i metodi presentano delle complessità

per esempio nel caso offline la costruzione di un banco prova che dal punto di vista economico ne influenza molto la scelta, mentre per l'online la tarazione dei vari filtri che si introducono nell'anello di retroazione. La scelta di uno a differenza dell'altro sta al costruttore e alle sue esigenze.

2.4 MODELLO MAGNETICO DEL MOTORE

Un aspetto che finora non è stato preso in considerazione a partire dalle equazioni 5 e di conseguenza riportato anche nello schema a blocchi del motore 2 è l'approssimazione del modello magnetico del motore a dei valori costanti e indipendenti dal punto di lavoro. Questa semplificazione comporta una certa imprecisione nel modello del motore in quanto per correnti elevate si genera il fenomeno della saturazione magnetica. La saturazione magnetica genera delle non linearità nella relazione tra flusso concatenato e corrente, per cui ad esempio non è corretta la relazione $\lambda_d = L_d i_d$ in quanto non è valida per correnti elevate. È altrettanto complesso e poco sensato scrivere una relazione matematica valida per tutti i motori in quanto questa relazione $\lambda = f(i)$ dipende da troppi parametri come il materiale utilizzato e la forma del rotore, per questo si introducono dei nuovi tipi di induttanze.

Si introduce quindi il concetto di induttanza differenziale L_{diff} ovvero il valore del coefficiente angolare della tangente valutata in un certo punto di lavoro, in altre parole la derivata del flusso concatenato valutata per una certa corrente

$$L_{diff} = \frac{d\lambda}{di} \quad (7)$$

Un classico comportamento è rappresentato in figura (4) dal quale si nota che la L_{diff} per correnti piccole varia poco rispetto alle correnti elevate. Si può scrivere a partire dalla legge di Lenz e sostituendoci 7 che:

$$\begin{aligned} u(t) &= \frac{d\lambda(i(t))}{dt} \\ &= \frac{d\lambda(i)}{di} \frac{di(t)}{dt} \\ &= L_{diff} \frac{di(t)}{dt} \end{aligned} \quad (8)$$

A questo punto le equazioni 5 e 6 devono essere riscritte per generalizzare queste relazioni e tener conto della saturazione:

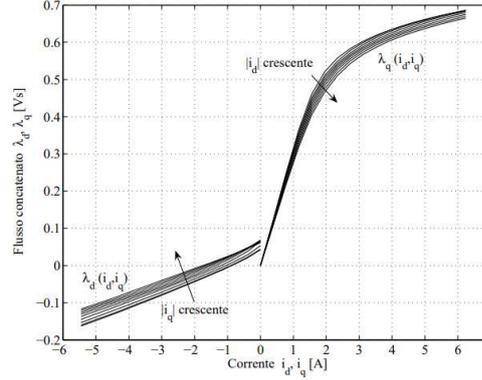


Figura 4: Andamento reale di $\lambda_d = f_1(i_d, i_q)$ e $\lambda_q = f_2(i_d, i_q)$. [1]

$$\begin{aligned}
 u_d &= Ri_d + \frac{d\lambda_d}{dt} - \omega_{me}\lambda_q \\
 u_q &= Ri_q + \frac{d\lambda_q}{dt} + \omega_{me}\lambda_d \\
 u_o &= Ri_o + \frac{d\lambda_\sigma}{dt}
 \end{aligned} \tag{9}$$

$$\tau = \frac{3}{2}p[\lambda_d i_q + \lambda_q i_d] \tag{10}$$

Un'ulteriore aspetto da tenere in considerazione è la cross-saturation ovvero il fatto che la saturazione lungo un asse è influenzata da entrambe le correnti. Per cui ad esempio la saturazione di λ_d per la maggior parte è dovuto a i_d ma c'è comunque l'influenza di i_q , allo stesso modo vale per l'altro asse. La minor influenza non è trascurabile in più risulta ancora non lineare la relazione a due variabili per cui quasi impossibile da riprodurre attraverso una funzione matematica ben definita.

Le relazioni generalmente sono descritte da:

$$\begin{aligned}
 \lambda_d &= f_1(i_d, i_q) \\
 \lambda_q &= f_2(i_d, i_q)
 \end{aligned} \tag{11}$$

Nella figura 4 che riporta un esempio reale di motore IPM si nota come per entrambi i flussi questo effetto è presente. In particolare lungo l'asse q dove la quantità di ferro è maggiore rispetto all'asse diretto per la presenza dei magneti permanenti, l'effetto della saturazione è più visibile in quanto solo la i_q non satura tutto il ferro presente e quindi anche la i_d interviene nella saturazione.

A questo punto per descrivere la dinamica del motore con la presenza di saturazione sono necessari dei nuovi schemi a blocchi che ne tengano conto. Nel modello del motore

serviranno le relazioni inverse ovvero data una corrente mi restituiscono i flussi. Non saranno rispettivamente le due inverse ma delle nuove funzioni [12](#)

$$\begin{aligned}i_d &= f_3(\lambda_d, \lambda_q) \\i_q &= f_4(\lambda_d, \lambda_q)\end{aligned}\tag{12}$$

MAPPATURA

3.1 OBIETTIVO DELLA TESI

L'obiettivo principale della tesi è lo sviluppo di una procedura per ottenere le mappe di flusso di un motore sincrono a magneti permanenti anisotropo. Il processo di sviluppo si basa inizialmente su un approfondito studio teorico del problema, seguito da un'implementazione pratica. Lo studio teorico è condotto attraverso l'analisi di un paper che forniva una spiegazione dettagliata del calcolo dei flussi in corrispondenza di specifici riferimenti di corrente. Successivamente, i dati forniti dal paper vengono elaborati per ricavare le informazioni necessarie. Dal punto di vista pratico, un codice MATLAB viene sviluppato e implementato su un banco prova preesistente. Una volta ottenute le mappe di flusso diretto, vengono invertite utilizzando una funzione di interpolazione, permettendo così di stabilire la relazione inversa corrente-flusso. La nuova relazione viene utilizzata per migliorare il controllo del banco prova. Infine, l'efficienza del motore è stata misurata mediante un wattmetro, che sfruttava un sistema di controllo basato sulle mappe ricavate e successivamente invertite.

3.1.1 Parte teorica

L'identificazione delle relazioni tra flussi concatenati e correnti, e viceversa, offre diversi vantaggi nell'azionamento del motore, tra cui:

- La determinazione analitica delle relazioni di coppia e potenza in diversi punti di lavoro.
- L'individuazione delle curve MTPA (Maximum Torque per Ampere) e MTPV (Maximum Torque per Voltage), al fine di ottimizzare l'uso del motore.

Esistono diverse metodologie per ottenere le mappe di flusso, tra cui metodi analitici basati su simulazioni e metodi sperimentali basati su misurazioni. I metodi analitici fanno uso di tecniche di simulazione come il Finite Element Method (FEM), che richiedono però informazioni dettagliate di progetto, generalmente accessibili solo ai progettisti del motore. I metodi sperimentali, invece, si basano sui dati di targa del motore e possono essere ulteriormente suddivisi in approcci con rotore bloccato o rotore trascinato a velocità costante. Il metodo con rotore bloccato risulta particolarmente efficace, poiché considera gli effetti della saturazione, come il cross-coupling e le non linearità del sistema come il metodo a rotore trascinato. Inoltre da la possibilità di

identificare effetti dovuti alla posizione, cosa che con i metodi a rotore in movimento non si riesce a fare. Tuttavia, questo metodo presenta una criticità in quanto in queste condizioni, l'inverter non opera alla tensione massima, questo perchè la tensione applicata va tutta sulla resistenza. Nel momento in cui venisse applicata la tensione massima, la corrente aumenterebbe fin tanto che i fili rischiano di non essere più isolati tra loro. Per ovviare a questo problema si diminuisce la tensione applicata, questa riduzione aumenta l'imprecisione dei risultati perchè l'inverter non lavora a valori nominali. Anche il metodo a rotore trascinato a velocità costante permette di ricavare tutti gli aspetti legati alle mappe di flusso. Permette di compensare la variazione della resistenza di rotore, in quanto varia al variare di temperatura ovvero punto di lavoro e consente di lavorare fino a valori nominali. Confrontando i due metodi si decide di utilizzare quello a rotore trascinato così da avere una minor incertezza nelle grandezze applicate, nonostante comporti l'utilizzo di un'ulteriore motore che funge da carico avente una velocità di trascinamento costante.

3.1.2 Procedura teorica

La procedura di identificazioni della 11 parte dalla relazione 9. Per ricavare il flusso è necessario lavorare a regime, ovvero quando il motore ha completato il transitorio di velocità dovuto alla variazione di corrente. I termini derivativi vanno a zero e risulta:

$$\begin{aligned} u_d &= Ri_d - \omega_{me}\lambda_q \\ u_q &= Ri_q + \omega_{me}\lambda_d \end{aligned} \quad (13)$$

A questo punto è necessario compensare la resistenza di statore attraverso l'applicazione di correnti con un certo andamento. Se si applica al controllo di corrente prima un vettore $i_1 = (i_d, i_q)$ poi un vettore $i_2 = (i_d, -i_q)$ ovvero il suo coniugato si ottiene l'eliminazione del termine resistivo mediando i due termini:

$$\begin{aligned} u_{d1} - u_{q2} &= Ri_{d1} - \omega_{me}\lambda_q - Ri_{d2} - \omega_{me}\lambda_q \\ u_{q1} + u_{q2} &= Ri_{q1} + \omega_{me}\lambda_d + -Ri_{q2} + \omega_{me}\lambda_d \\ u_{d1} - u_{d2} &= -2\omega_{me}\lambda_q \\ u_{q1} + u_{q2} &= 2\omega_{me}\lambda_d \\ -\frac{u_{d1} - u_{d2}}{2\omega_{me}} &= \lambda_q \\ \frac{u_{q1} + u_{q2}}{2\omega_{me}} &= \lambda_d \end{aligned} \quad (14)$$

quindi la relazione appena ricavata sarà utilizzata per stimare i flussi.

Un ulteriore aspetto è la variazione di resistenza dovuta allo scorrimento della corrente negli avvolgimenti, che attraverso l'effetto Joule il suo valore cambia. La

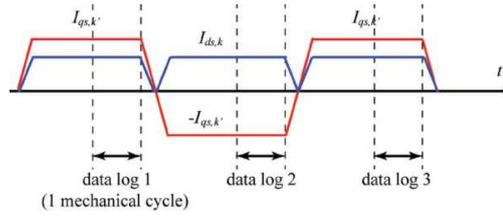


Figura 5: Andamento dei riferimenti i_{dq} -[2]

variazione della resistenza ha un relazione lineare nel tempo, nel momento in cui si utilizzassero solo due vettori di corrente i_{d1} e i_{d2} la compensazione non avverrebbe in quanto il valore continua a crescere linearmente. Per ovviare a questo problema viene introdotto un terzo vettore di corrente i_{d3} il quale viene applicato dopo i_{d2} ed è uguale a i_{d1} . Questo terzo segnale ha il compito di compensare questa variazione, in quanto facendo la media tra il primo e il terzo segnale il valore di resistenza media risulta uguale a quella del secondo segnale. In questo modo avviene una completa eliminazione della corrente. Durante l'applicazione di un punto di lavoro, la temperatura del motore non subisce variazioni in quanto la costante di tempo del modello termico è più grande rispetto a quella di corrente quindi la compensazione non è influenzata da ciò.

Per la formula per stimare i flussi cambia e diventa:

$$\begin{aligned}\lambda_q &= -\frac{1}{2} \cdot \left(\frac{u_{d1} + u_{d3}}{2} - u_{d2} \right) \cdot \frac{1}{\omega_{me}} \\ \lambda_d &= \frac{1}{2} \cdot \left(\frac{u_{q1} + u_{q3}}{2} + u_{q2} \right) \cdot \frac{1}{\omega_{me}}\end{aligned}\quad (15)$$

Questa operazione viene svolta per ogni punto di lavoro il quale avrà coordinate diverse (i_d, i_q). L'andamento dei riferimenti di corrente per un certo punto è riportato nella figura 5.

A questo punto è necessario scegliere un piano di lavoro, ovvero tutti i punti in cui verranno calcolati i rispettivi flussi. In generale la scelta del piano di lavoro va in base ai valori nominali del motore ma questa zona può essere ampliata o ristretta se si conosce già a posteriori le zone di lavoro dove il motore andrà ad operare. Generalmente viene creata una mappa rettangolare o quadrata dove tra un punto di lavoro e l'altro ci si sposta con un valore costante ad esempio la figura 6.

Si definiscono quindi i punti di lavoro come tutte le combinazioni di k e k' della formula 16, m e n sono il numero di intervalli lungo d e q mentre Δ_{id} e Δ_{iq} è lo spostamento minimo tra un punto e l'altro.

$$\begin{aligned}i_{dk} &= i_{d, \min} + k\Delta_{id} \quad k = 1, \dots, m \\ i_{qk'} &= i_{q, \min} + k'\Delta_{iq} \quad k' = 1, \dots, n\end{aligned}\quad (16)$$

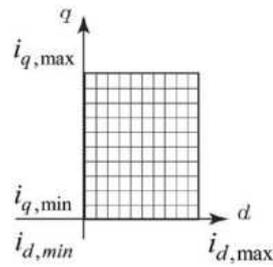


Figura 6: Mappa di lavoro.[2]

La scelta del numero di intervalli verticali e orizzontali ne influenza la veridicità della mappa che si genererà al termine della prova, tanto più una mappa è fitta ovvero Δ piccoli tanto più è simile a quella reale. In particolare nel momento in cui una mappa è costituita da zone particolarmente non lineari l'aumento di punti rende migliore la mappa. Mentre se la mappa è quasi lineare la scelta di avere lo stesso numero di punti risulta eccessiva.

Infine l'ultimo aspetto da tenere in considerazione è il tempo in cui i valori di tensione e velocità vengono campionati. Tale tempo risulta un compromesso tra l'averne un buon numero di campioni così da avere un minor errore rispetto al valore vero e l'aumento di temperatura che influenza il valore di R . Come mostrato in figura 5 il campionamento avviene per un giro meccanico, quindi per completare un punto di lavoro sono necessari tre giri meccanici in quanto questo processo si ripete ogni i_j dove $j = 1, 2, 3$.

3.1.3 Parte sperimentale

La parte sperimentale di questa tesi ha come obiettivo lo sviluppo di un metodo che metta in pratica attraverso diversi dispositivi la parte teorica presentata in precedenza. Questo processo è diviso in due fasi temporali:

1. Simulazione del processo
2. Applicazione al banco prova

Queste due fasi sono fondamentali perchè la fase di simulazione permette un primo approccio al problema attraverso delle simulazioni che altrimenti nella realtà risulterebbero impossibili da fare in quanto potrebbero creare dei problemi al sistema reale inteso come inverter e motore. In più, permettono di essere maggiormente flessibili in modifiche da compiere prima dei test reali.

3.1.3.1 Simulazione del processo

La simulazione viene compiuta attraverso l'utilizzo del programma *MatlabSimulink*. L'idea iniziale è stata quella di ricreare un azionamento reale, cioè un sistema discreto

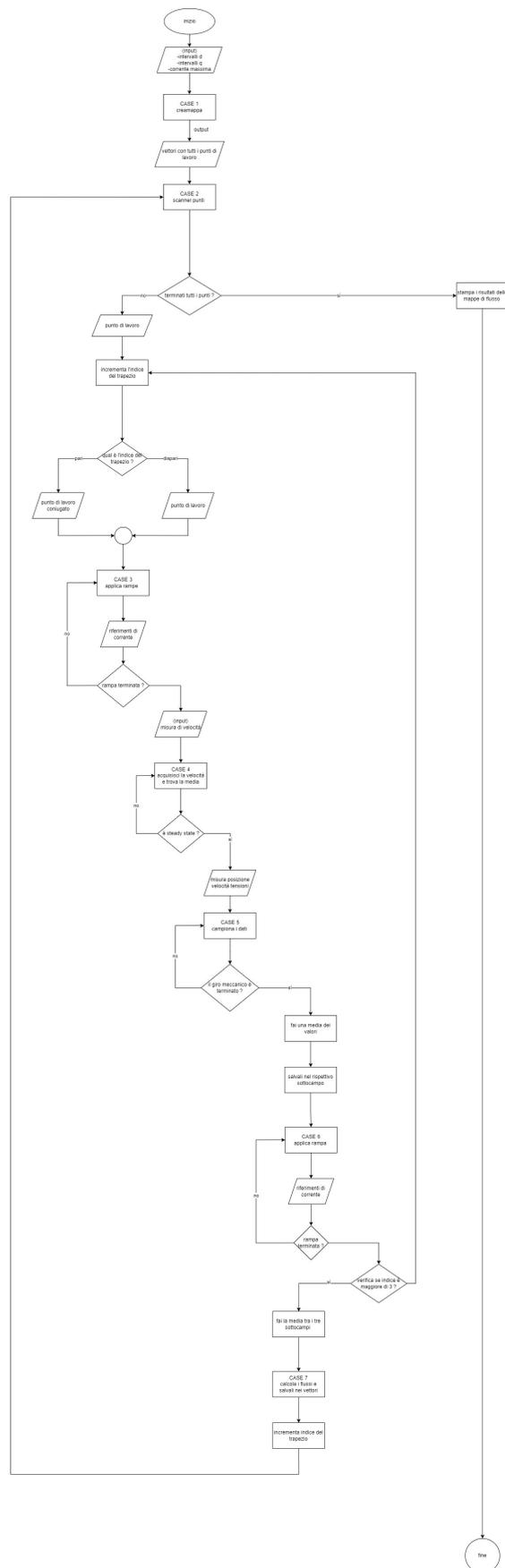


Figura 7: Flow Chart del programma.

nel tempo dove un motore è comandato in velocità. L'azionamento iniziale era già stato sviluppato precedentemente a lezione infatti la simulazione sfrutta quel sistema composto da motore sincrono a magneti permanenti anisotropo e inverter i quali sono controllati da un controllo di velocità con annidato un controllo di corrente.

Come primo step si applica il test ad un motore del quale si conoscono già le relazioni di flusso, per poi confrontare i risultati ottenuti con quelli noti. L'idea alla base è quella di creare una funzione tale da generare i riferimenti di corrente i_{dq} da applicare al controllo.

La Matlab Function in questione oltre al compito di generatore di riferimenti dovrà anche stimare il valore del flusso in base ai valori campionati durante i tre segnali visti nella formula 15. Ricapitolando gli ingressi necessari sono:

- Intervalli lungo d
- Intervalli lungo q
- I_{max} : modulo massimo di corrente che verrà applicato
- ω_{mut} : velocità con cui il MUT viene trascinato
- u_{dq} : necessari per calcolare 15
- θ_{mut} : la posizione è fondamentale per capire quando è completato il giro meccanico

Mentre come uscite sono presenti:

- i_d^{REF} : corrente di riferimento di asse d
- i_q^{REF} : corrente di riferimento di asse q
- vettore del flusso λ_d : vettore dove all'interno sono presenti le stime di flusso d per tutti i punti di lavoro
- vettore del flusso λ_q : vettore dove all'interno sono presenti le stime di flusso q per tutti i punti di lavoro

Il processo termina nel momento in cui tutti i punti di lavoro scelti in precedenza verranno applicati. In quel momento sono presenti quattro vettori della stessa dimensione: uno con le coordinate di i_d^{REF} , uno con i_q^{REF} i quali serviranno per generare il dominio dove verranno applicati in un grafico il vettore dei flussi λ_d mentre nell'altro i flussi di λ_q . Si ottengono così due superfici le quali rappresentano la relazione 11.

Alla base di questa Matlab function viene gestita come una macchina a stati, in quanto questo processo possiede una ciclicità in quanto il processo per ottenere la stima del flusso deve essere ripetuto per ogni punto di lavoro. Attraverso un breve layout si presenta il ciclo base della macchina a stati come in figura 8. Osservando il flowchart del principio di funzionamento della macchina a stati, il primo stato ha il compito di

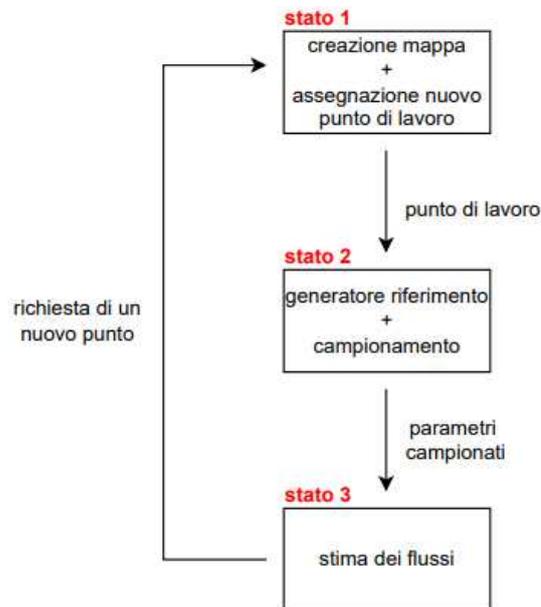


Figura 8: Flow chart degli stati operativi.

creare la mappa di lavoro ovvero tutte le combinazioni dei punti di lavoro scelte in base agli input presenti nella funzione. Questo processo è singolo infatti viene compiuto una sola volta in quanto sarà una seconda funzione a passare quando necessario il punto di lavoro al secondo stato solo nel momento in cui la stima del flusso è stata compiuta; il processo terminerà solamente quando esauriscono tutti i punti. Il secondo stato invece ha il compito di generare l'andamento voluto come nella figura 5 ma anche campionare le misure di tensione. Infine l'ultimo stato dovrà stimare il valore di flusso salvarlo per poi confermare la richiesta di un nuovo punto.

Dopo una descrizione generale si approfondisce ora tutte le funzioni presenti nei vari case della macchina a stati. La matlab function è stata gestita da un grande switch case composto da otto stati, per passare da uno stato all'altro si va a cambiare la variabile persistent *statomacchina* che ha in base al valore va nel case corretto. Il case 1 viene svolto solo una volta in quanto ha il compito di creare tutti i punti di lavoro secondo un certo andamento scelto, il dominio su cui si va ad operare è un cerchio avente centro nel punto $(0,0)$ e raggio pari al valore nominale della corrente I_n del motore da testare il quale è presente nei dati di targa. La scelta di avere un dominio a circolare invece che un quadrato centrato sempre in $(0,0)$ ma di lato $2I_n$ in quanto opero per valori positivi e negativi come I_n e $-I_n$ ma nel momento in cui si volesse operare nei vertici del quadrato il modulo di corrente risulta $\sqrt{2}I_n$ e questo è un problema nel momento

in cui si applica in quanto potrebbe corrispondere a zone di lavoro poco sicure per il motore under test. Mentre con un dominio circolare il modulo massimo corrisponde alla valore del raggio.

Da tenere in considerazione per la creazione della funzione che genererà i punti è il tipo di distribuzione che si vorrà impostare. Ad esempio se vogliamo imporre una distribuzione radiale dove la distanza è omogenea tra un punto e l'altro si genera un problema in vicinanza del centro dove ci sono molti più punti rispetto alle zone lungo il perimetro. Oppure è possibile creare una distribuzione a griglia la quale migliora il problema nel centro mentre lungo il perimetro la disposizione non risulta efficace. La soluzione finale risulta un ibrido tra le due infatti si è inscritto nel cerchio un quadrato dove al suo interno la distribuzione è a griglia mentre risulta radiale per le zone tra i due perimetri.

La funzione che si utilizza è *creamappa* che ha come ingressi:

- intervalli lungo l'asse d
- intervalli lungo l'asse q
- raggio del cerchio (modulo della corrente massima)

mentre in uscita:

- matrice dei riferimenti i_d
- matrice dei riferimenti i_q

L'idea alla base è generare prima i punti di un quarto di piano (quello positivo) per poi specchiare i valori prima con l'asse q poi con l'asse d. Il quarto di cerchio è stato diviso in tre zone: il quadrato e i due spicchi rimanenti. La zona del quadrato è suddivisa dagli intervalli definiti dagli ingressi mentre le altre due aree vengono gestite diversamente. I punti lungo le due aree seguono un andamento radiale con direzioni che passano dal centro alle intersezioni con i punti presenti lungo il perimetro, mentre la spaziatura viene ridefinita. Questo valore sfrutta quello presente negli intervalli e viene fatto variare affinché si genera un numero naturale di intervalli. Questo processo ha l'obiettivo di creare una maggior omogeneità nel piano. Come riportato nella figura 9.

Terminata la funzione *creamappa* che come output da in uscita due matrici, essendo i dati salvati a forma di cerchio in diverse celle risulta presente il valore "NaN" (Not a Number) in quanto non sono presenti valori. A questo punto per elaborare meglio i dati, le matrici vengono trasformate in vettori e poi attraverso la funzione *togliNan* si eliminano i "Nan" e i valori vengono sheftati in modo tale che in un vettore ottengo tutti i valori di i_d^{REF} e nell'altro i_q^{REF} . In questo modo risulta semplice passare da un punto ad un altro in quanto basta incrementare l'indice dei due vettori.

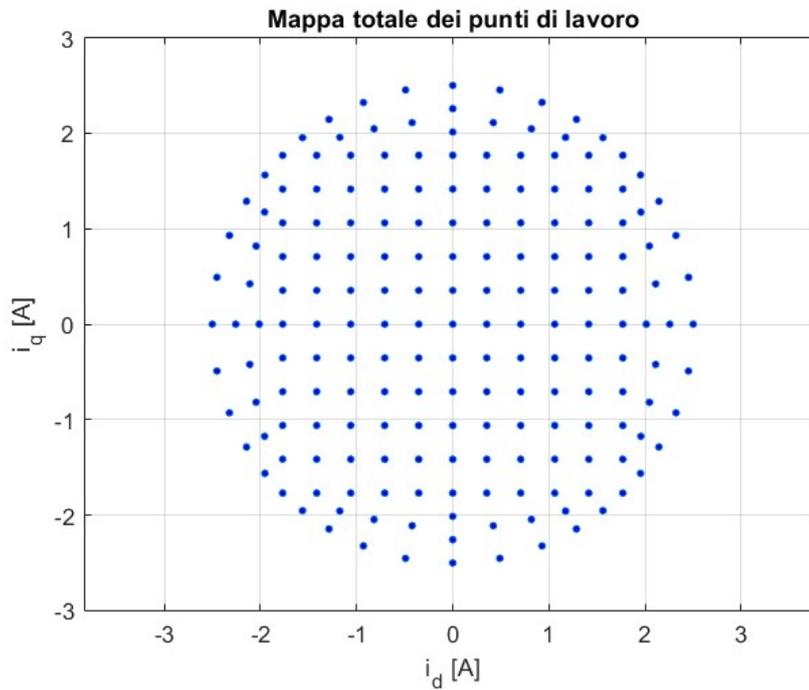


Figura 9: Mappa dei punti con raggio 2,5 A, con 10 intervalli lungo d e q.

Inoltre si genera un problema ovvero il fatto di non sapere precisamente quanti sono i punti di lavoro presenti all'interno della mappa. Risulta complicato trovare una funzione che lega intervalli a punti totali, si riporta un esempio nella tabella 1 dove non è presente una relazione lineare. Ricapitolando, il case 1 termina con la creazione di due vettori che contengono tutte le combinazioni di punti da applicare durante il test per poi passare al case 2.

Il case 2 sfrutta la funzione *scanner-punti*, la quale ha il compito di generare il punto di lavoro voluto. Per passare da un punto ad un altro è necessario incrementare il valore dell'indice dei vettori. La funzione *scanner-punti* ha come ingressi:

- vettore dei riferimenti di i_d
- vettore dei riferimenti di i_q
- enable-scanner
- l'indice del trapezio
- l'indice del vettore

mentre in uscita:

Intervalli	Punti nel quadrato	Punti totali
6	49	89
8	81	129
10	121	189
12	169	277
14	225	357

Tabella 1: Relazione tra i punti presenti nella mappa.

- riferimento i_d
- riferimento i_q
- termine della mappa

La funzione restituisce il punto di lavoro ogni qual volta che viene richiamata ma la conferma del valore è data da *enable-scanner*. Essendo che questo processo si ripete in modo ciclico ovvero per ogni punto di lavoro sono necessari tre andamenti di corrente a forma di trapezio dove in ognuno di essi si applica lo stesso processo, il termine *indice-trapezio* ha il compito di riconoscere in quale trapezio si sta operando. L'*indice-trapezio* è fondamentale in quanto se vale 2 il valore di regime ha i_q negativa e i_d invariata mentre quando vale 3 significa che dal prossimo trapezio si aggiorna il punto di lavoro e prima di ogni inizio gli indici si azzerano. Il processo continua fin tanto che l'*indice-vettore* non supera la dimensione di uno dei due vettori in ingresso, a quel punto significa che tutti i punti sono stato fatti e il processo termina in quanto lo *statomacchina* va direttamente nel *case 8*.

Ora che il valore di regime a cui arrivare per entrambe le correnti è noto, attraverso la funzione *rampa* il riferimento di corrente passa da 0 al valore di regime. Il processo ora è momentaneamente bloccato nel *case 3* fin tanto che le funzioni non terminano perchè viene attivata in parallelo, una per la corrente i_d e una per la i_q . Il valore in uscita dalla funzione *rampa* rappresenta l'uscita della Matlab function *generatore* in quanto sono i riferimenti da dare ai controllori di corrente. La funzione *rampa* ha un comportamento discreto in modo da simulare la realtà dei microcontrollori quindi ha come ingressi:

- tempo di durata della rampa
- valore di partenza della corrente
- valore finale della corrente
- valore di corrente al passo precedente

- periodo di commutazione del processo

mentre in uscita:

- valore di corrente attuale
- enable di conclusione rampa

Il tempo di durata della rampa è l'unico ingresso che risulta come input del processo totale. È fondamentale in quanto permette di gestire la sovra elongazione della corrente terminata la rampa, questo incremento va di conseguenza ad influenzare la coppia e quindi la velocità del sistema. Infatti per tempi molto piccoli risulta quasi un gradino quindi l'effetto del sovra elongazione è ampio. Il controllo di corrente ha un effetto limitato in quanto i valori precisi del flusso sono sconosciuti e la costante di tempo è imprecisa. La scelta del tempo della rampa risulta un compromesso tra la sovra elongazione della corrente e durata totale del processo. È chiaro che la sovra elongazione ha un impatto minore sulla scelta perchè i valori utili vengono campionati in un momento di steady-state cioè quando la sovra elongazione è terminata. Il funzionamento di base è quello di incrementare ogni volta che la funzione viene chiamata il valore del riferimento presente in uscita dalla funzione di 17 dove i_{end} è il valore finale della corrente, i_{start} è il valore di partenza della corrente, t_{rampa} è il tempo di durata della rampa, T_s è il periodo di commutazione del processo.

$$\Delta i = (i_{end} - i_{start}) / t_{rampa} * T_s \quad (17)$$

Questa funzione è richiamata due volte nel *case 3* una per ogni riferimenti di corrente. Il termine avviene in modo sincronizzato perchè l'incremento non è costante ma dipende dalla formula 17 quindi attivano nello stesso ciclo di calcolo il termine *enable di conclusione rampa*. Quando i due enable sono attivi significa che le rampe sono terminate e da ora in poi i riferimenti sono costanti e pari al *valore finale della corrente* cioè l'output di *scanner-punti*. A questo punto si incrementa il valore di *statomacchina* per passare al prossimo case.

Il *case 4* ha il compito di riconoscere se stiamo operando in steady-state. Dal punto di vista pratico significa che la velocità risulta costante o entro un certo range. La velocità cambia nel momento in cui si passa da un punto di lavoro ad un altro, in quel momento quando la velocità non è costante se si andasse a campionare i valori non si troverebbe il corretto flusso in quanto si introdurrebbe il termine di derivata. Questo compito è svolto dalla funzione *steady-state* la quale ha come ingressi:

- velocità misurata
- velocità con cui viene svolta la prova

- numero di campioni per la media
- valore di epsilon
- indice di incremento per epsilon
- vettore che contiene i campioni di velocità
- indice del vettore di velocità

mentre in uscita:

- enable di conferma dello stato steady state
- indice di incremento per epsilon
- vettore che contiene i campioni di velocità
- indice del vettore di velocità

Fin tanto che non si raggiunge lo stato di steady state si rimane all'interno di questo case. In questo periodo la funzione *steady-state* ha il compito di campionare la velocità ogni qual volta venga chiamata per poi salvarla all'interno del *vettore che contiene i campioni di velocità*. La dimensione si decide attraverso un metodo empirico cioè sfruttando delle prove precedenti per capire quanto può durare il periodo del disturbo. Questa chiaramente è una stima molto grossolana in quanto cambierà a seconda del punto di lavoro la durata del disturbo. La scelta di salvare i dati in un vettore è stata fatta per osservare i valori di velocità nel momento della scelta del numero di campioni. Terminato il campionamento, i valori vanno mediati per poi confrontare se questi sono compresi all'interno di un intervallo definito. Per regolare l'ampiezza dell'intervallo si cambia *valore di epsilon*, costante che regola la percentuale rispetto al *velocità a cui viene svolta la prova*. Attraverso la formula 18:

$$|w_{\text{riferimento}} - w_{\text{stimata}}| < \epsilon \cdot w_{\text{riferimento}} \quad (18)$$

dove $w_{\text{riferimento}}$ è la *velocità a cui viene svolta la prova*, w_{stimata} è la media del *vettore che contiene i campioni di velocità* mentre $\epsilon w_{\text{riferimento}}$ rappresenta la percentuale di w_{stimata} .

La relazione 18 serve per capire se il valore medio di velocità è compreso all'interno dell'intervallo, alle volte può risultare che questa relazione non sia vera a quel punto la funzione *steady-state* completa un nuovo campionamento di velocità per poi incrementare ϵ e verificare che 18 sia vera. Questo processo di aumento del range è stato utilizzato per non bloccare il processo nel momento in cui il controllo non sia fatto in modo efficiente. Questo incremento necessita chiaramente di un saturatore per epsilon ovvero di un valore entro il quale non può aumentare ulteriormente, perchè altrimenti la stima del flusso può non essere corretta.

Nel momento in cui il valore di uscita *enable di conferma dello stato steady state* della funzione *steady-state* risulta vero, allora si cambia case e si passa al successivo. All'interno del *case 5* si campionano diverse grandezze del motore da sfruttare poi per la stima del flusso. Come per i case precedenti, è presente una funzione per gestire il campionamento e fin tanto che non termina, il processo rimane bloccato nel seguente case. Si sfrutta la funzione *campiona* che ha come ingressi:

- campioni di diverse grandezze
- valore cumulato delle diverse grandezze
- posizione di partenza
- indice della quantità di campioni
- indice che rappresenta uno dei tre trapezi
- vettore che contiene i campioni di velocità
- indice del vettore di velocità

mentre in uscita:

- *enable di conferma della conclusione del campionamento*
- valore medio di ogni misura
- valore cumulato delle diverse grandezze
- indice della quantità di campioni

Questa funzione viene attivata dopo aver ricevuto la conferma che lo stato di lavoro in cui si trova è *steady state*. L'obiettivo è quello di campionare tensione e velocità per un giro meccanico, infatti la funzione riceve in ingresso la posizione in cui si trova il rotore durante la prima chiamata della funzione e terminerà quando il rotore passa in quel punto. Si deve rispettare la relazione 19

$$|(thm_{iniziale} + 2 * pi) - misura.thm| \geq 0 \quad (19)$$

dove $thm_{iniziale}$ è la posizione iniziale salvata nel momento in cui è terminata la funzione *steady-state*, *misura.thm* è la posizione meccanica del rotore nell'istante in cui è chiamata. Fin tanto che questa relazione è valida si sommano al *valore cumulato delle diverse grandezze* i *campioni di diverse grandezze* e si incrementa *indice della quantità di campioni*. Si esegue in *free run* ovvero ad ogni tempo di campionamento del processo in modo tale da essere più precisi possibili. Essendo per ogni punto di lavoro si campiona per tre volte le stesse grandezze, il termine *valore medio di ogni misura* è composta da un campo dove a sua volta contiene tre sottocampi. In questo caso sono presenti i termini:

1. valoremedio.ud
2. valoremedio.uq
3. valoremedio.wm

Ognuno di questi possiede altre tre sottocampi ad esempio *valoremedio.ud* contiene:

1. valoremedio.ud.x1
2. valoremedio.ud.x2
3. valoremedio.ud.x3

Questo tipo di struttura ha il compito di salvare i dati in un sottogruppo diverso per ogni trapezio. Infatti a seconda dell'*indice che rappresenta uno dei tre trapezi* si salva il rispettivo sottogruppo. Quando il campionamento è terminato si salvano i valori medi a nel rispettivo campo e poi si incrementa il valore dell'*indice che rappresenta uno dei tre trapezi* mentre tutti gli altri valori si azzerano e si incrementa lo *statomacchina*.

Arrivati al *case 6* si applica le funzioni *rampa* in modo tale da portare i riferimenti a zero. Chiaramente i parametri sono uguali al *case 3* solo che la rampa sarà decrescente per cui la corrente di arrivo è zero mentre quella di partenza è il punto di lavoro. Terminato questo processo a seconda dell'*indice che rappresenta uno dei tre trapezi* le strade del processo possono essere due. Nel momento in cui manca rimena da campionare almeno un trapezio lo *statomacchina* ritorna al *case 2* dove si definisce il punto di lavoro (nel caso del secondo trapezio il punto di lavoro possiede la i_q negata) e tutto il processo riparte per poi terminare e salvare i dati in un nuovo sottogruppo. L'altra strada possibile si presenta nel momento in cui il processo precedentemente descritto si ripete per tre volte, a questo punto si passa al *case 7*.

All'interno del *case 7* l'obiettivo è quello di calcolare il flusso. L'operazione è svolta dalla funzione *calcola-flusso* dove si applica la formula 15 sfruttando i sottocampi trovati precedentemente. Per cui si sfrutta la funzione *calcola-flusso* che ha come ingressi:

- valore medio di ogni misura
- coppie polari del motore

mentre in uscita:

- flusso lungo d
- flusso lungo q

Al termine della funzione i valori di flusso verranno salvati in due vettori nella cella di memoria con indice pari a quello che avevano i due riferimenti. In questo modo ad ogni indice corrisponde una coppia di correnti i_d , i_q e i flussi rispettivi λ_d , λ_q . La conoscenza del numero di *coppie polari del motore* è necessario per ricavarsi la velocità elettrica. Infine si azzerava l'indice che rappresenta uno dei tre trapezi e si incrementa l'indice dei vettori per il nuovo punto di lavoro. Tutto il processo termina nel momento in cui si completano tutti i punti di lavoro da lì in poi i riferimenti di corrente saranno nulli.

3.1.3.2 Implementazione sul banco prova

Dopo aver sviluppato un programma funzionante in Simulink ed essersi assicurati che il processo funzionasse, ora andava implementato nella realtà. L'implementazione è avvenuta in un banco prova già presente in laboratorio composto da diversi componenti. Era composto da due motori calettati tra di loro, e comandati dai loro relativi inverter. Il comportamento di questi due motori è indipendente tra loro, è presente poi un'elettronica che fornisce attraverso dei DAC i valori di vari sensori. Per far comunicare il programma Simulink con il banco prova è presente una dSpace che connette in real time i due processi.

Il programma utilizzato durante l'implementazione reale non è rimasto uguale a quello per la simulazione. La logica del processo è rimasta invariata in quanto i passaggi da compiere sono gli stessi, cioè il numero di case è rimasto invariato. Le modifiche subite sono state due:

- *la creazione dei punti di lavoro*: in simulazione la matlab function riceveva il numero di intervalli lungo d e q come input per cui la dimensione della mappa poteva cambiare. Nella logica del programma una volta generata la mappa dei punti questi valori non potevano essere cambiati in quanto il *case 1* veniva processato solo una volta quindi anche se si cambiavano i valori non influivano. Ciò è possibile perchè Matlab da la possibilità di allocare la memoria in modo dinamico cioè a seconda della richiesta è in grado di aggiungere degli elementi ad un vettore o mappa. Questa proprietà permetteva di non dare errore durante il test. Nel microcontrollore della dSpace questa operazione non è possibile infatti è necessario definire prima la grandezza del vettore, la dimensione calcolarla prima è complicato quindi per ovviare al problema il *case 1* è stato svolto in un file esterno e passato alla matlab function come parametro.
- *riconoscimento del giro meccanico*: la funzione che gestiva questo aspetto era la funzione *campiona* in quanto il campionamento avviene per un giro meccanico. Durante la simulazione il valore della posizione continuava ad incrementarsi perciò era necessario [19](#). Nel banco prova il valore in uscita dall'encoder era un segnale che passava da 0 a 2π , con andamento a dente di sega, è stato necessario un cambiamento nel riconoscere il giro completo in quanto non vale più la relazione [19](#).

- *gestione della velocità*: nella simulazione per mantenere la velocità costante non è stato applicato un ulteriore motore. Variando i parametri del motore, cioè aumentando l'inerzia di un valore quasi infinito, e definendo la velocità iniziale paria a quella di riferimento. In questo modo la velocità iniziale non varia perchè partendo dalla relazione coppia τ e velocità ω :

$$\tau = J \frac{d\omega}{dt} + B\omega \quad (20)$$

dove J è il momento d'inerzia e B è il coefficiente di attrito viscoso, nel momento in cui $J \rightarrow \infty$ allora $B \simeq 0$ quindi

$$\frac{d\omega}{dt} = \frac{\tau}{J} \simeq 0 \quad (21)$$

questo significa che la variazione di velocità è nulla per cui la velocità rimane sempre costante, di conseguenza nel momento in cui la velocità iniziale risulta ω_0 non subisce variazioni.

Nella realtà questo non è possibile quindi si hanno due motori collegati tra loro. Il motore che si comporta da carico è comandato da un controllo in velocità infatti la mantiene costante. Il motore che si vuole testare è controllato in corrente cioè dai riferimenti generati dalla matlab function.

Dopo aver cambiato le parti che rendevano impossibile l'implementazione sono stati settati i parametri per svolgere le misure. Il tempo di salita impostato a 1000 volte il *tempo di campionamento* che risulta paria $1 \cdot 10^{-4}s$, un modulo di corrente pari a $20A$, 14 intervalli lungo d e q per 357 punti di lavoro. Infine prima di attivare la parte di potenza in modo continuativo del banco prova sono stati tarati i k_p e k_i grazie all'interfaccia utente ControlDesk della dSpace, dove si monitorano i segnali in tempo reale. Dando la possibilità di regolare i parametri osservando la dinamica del controllo, cioè visualizzando l'andamento del riferimento e del segnale reale. Terminata la regolazione si attiva la parte di potenza così da iniziare il processo per poi analizzare i dati ottenuti.

RISULTATI OTTENUTI

4.0.1 Risultati ottenuti dalla simulazione

Attraverso la simulazione è possibile vedere come si comporta il sistema e se i risultati sono attendibili. Il motore utilizzato per la simulazione aveva delle mappe di flusso che rispettavano la relazione 15

$$\begin{aligned}\lambda_d &= \lambda_{PM} + L_d \cdot i_d \\ \lambda_q &= L_q \cdot i_q\end{aligned}\quad (22)$$

dove $L_d = L_q = 0,32 * 10^{-3}[H]$ e $\lambda_{PM} = 0.0082[Vs]$
Si ottiene così una mappa di flusso come in 10.

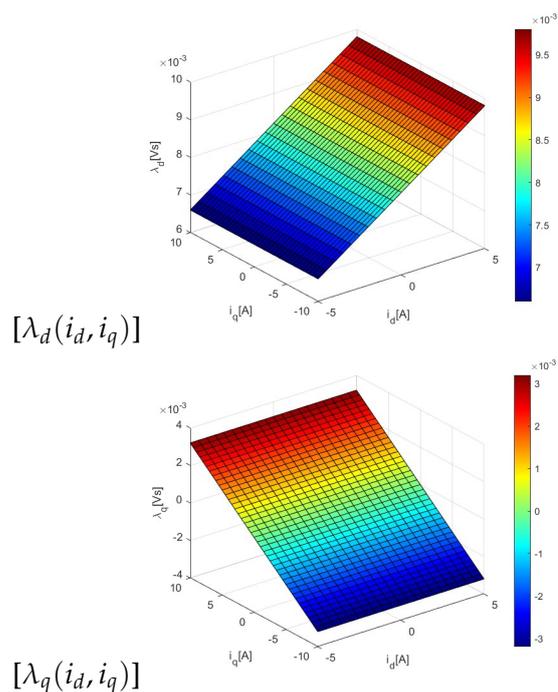


Figura 10: Relazioni dei flussi nel motore simulato

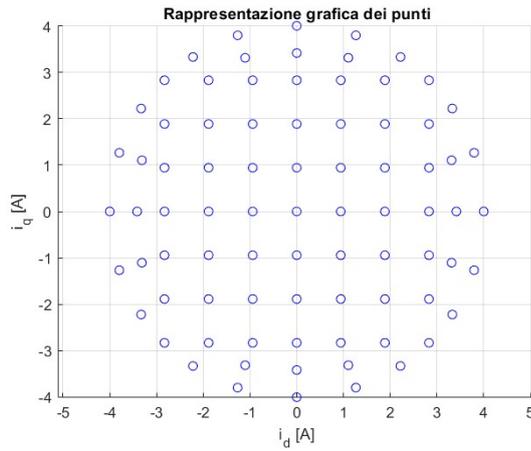


Figura 11: Mappa con i punti di lavoro della simulazione.

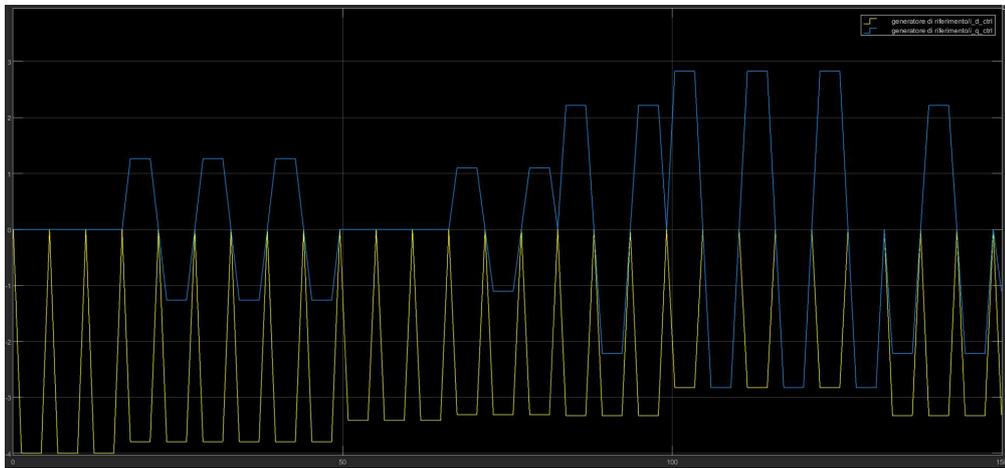


Figura 12: Andamento dei riferimenti dei punti di lavoro.

A questo punto la mappa comprende sei intervalli lungo i due assi del quadrato, un raggio di $4A$ in quanto la corrente nominale vale $4,5A$. In figura 11 si trova le coordinate mentre in figura 12 è possibile vedere l'andamento dei vari riferimenti.

Al termine della prova è possibile osservare come il flusso ottenuto dal campionamento è sovrapposto a quello impostato nel motore ad inizio test, si osserva dalle figure 13 14

Per avere un'ulteriore conferma del fatto che i due valori coincidono, ogni volta che il flusso veniva stimato è stato pure calcolato sfruttando la formula 22 per poi salvare in un vettore l'errore tra i due valori. Osservando dalle figure 15 16 gli errori lungo d e q hanno un errore massimo di $6 * 10^{-6}[Vs]$ che risulta totalmente ininfluenza sul flusso stimato.

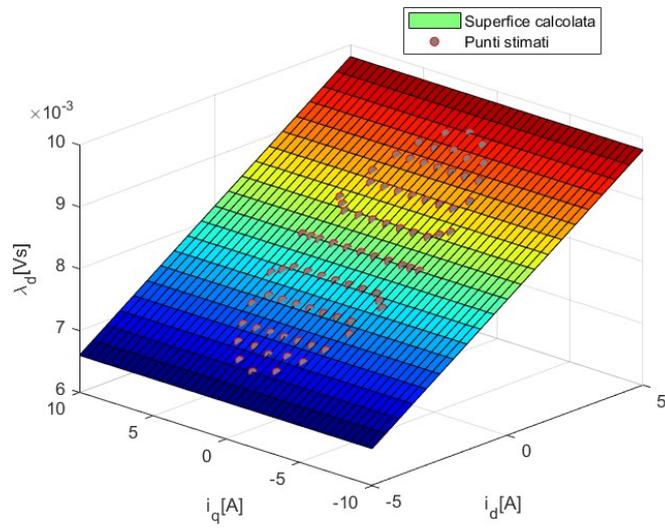


Figura 13: Confronto tra la stima lungo d e il calcolo.

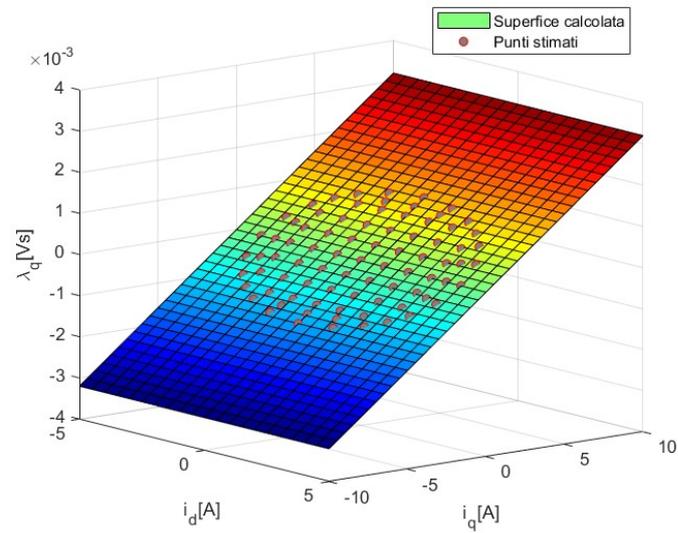


Figura 14: Confronto tra la stima lungo q e il calcolo.

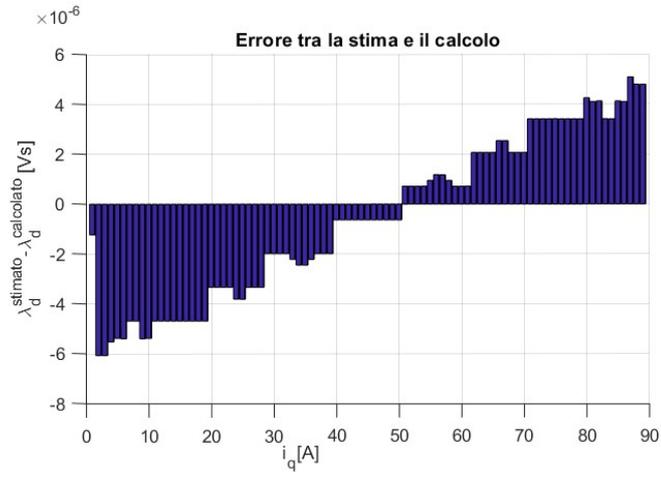


Figura 15: Errore tra la stima e il calcolo lungo d.

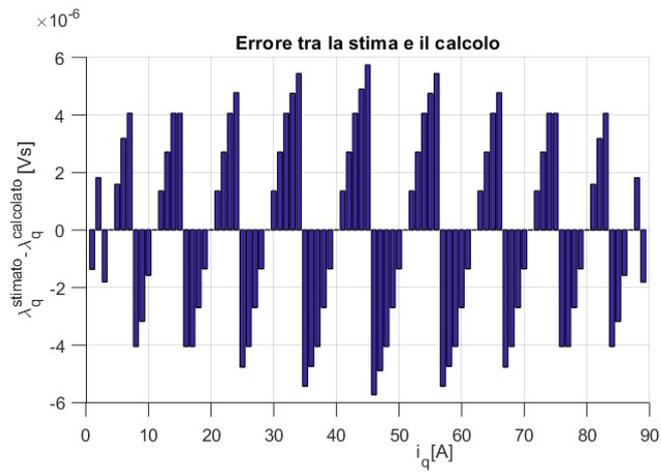


Figura 16: Errore tra la stima e il calcolo lungo q.

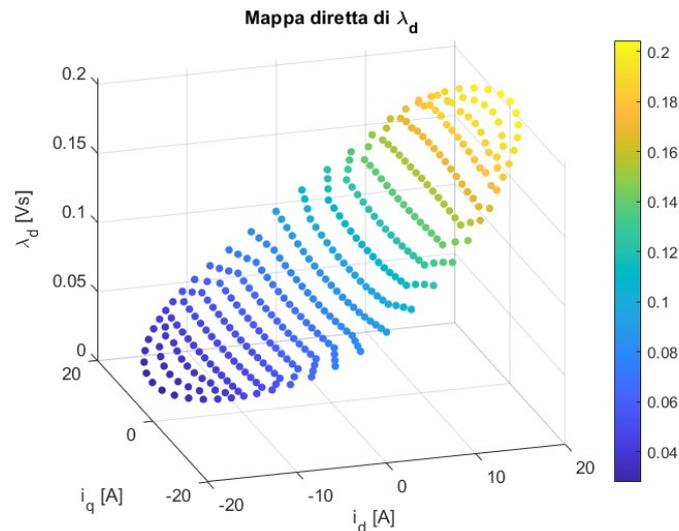


Figura 17: Mappa sperimentale diretta di λ_d .

4.0.2 Risultati ottenuti dalla prova sperimentale

I dati ottenuti dalla prova sperimentale sono due vettori contenenti i punti di lavoro forniti come ingresso e due vettori della stessa dimensione contenenti i flussi λ_d e λ_q . Quello che ci si aspetta da questo esperimento è una mappa circolare dove la distribuzione equivale a quella visualizzata nella simulazione per cui non omogenea. Il risultato ottenuto è un insieme di punti in quanto non possibile visualizzare una mappa contenente il valore NaN dovuto alla forma circolare dei punti, per questo abbiamo dei vettori con i soli valori effettivi. Il risultati della relazione diretta sono [17](#) e [18](#).

Per ottenere una relazione utile ai fini pratici è necessario creare una superficie di lavoro. L'obiettivo è trovare una relazione anche in dei punti di lavoro dove il valore del flusso non è stato ricavato per via sperimentale ma ottenuto attraverso un'interpolazione. L'interpolazione è una tecnica matematica utilizzata per stimare valori incogniti di una funzione a partire da un insieme finito di dati. Nel nostro caso, vogliamo ricostruire una funzione continua partendo dai punti campionati durante il processo spiegato precedentemente. La scelta della tecnica di interpolazione dipende fortemente dalla distribuzione dei dati e dalle specifiche esigenze computazionali e applicative. Ho confrontato tre metodi di interpolazione presenti in Matlab che potevano essere utilizzati con i dati a disposizione. I metodi sono:

- *scatteredInterpolant*: è utilizzata principalmente per dei dati disposti in modo non omogeneo, sfrutta metodi di interpolazione come la triangolazione di Delaunay. Le principali proprietà sono: la flessibilità in quanto non è necessario una griglia regolare, l'interpolazione di spazi a più dimensioni. L'aspetto negativo è il tempo

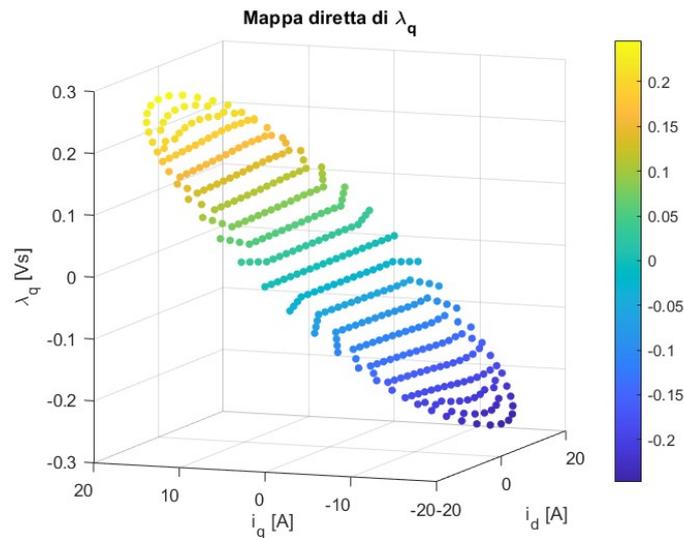


Figura 18: Mappa sperimentale diretta di λ_q .

di calcolo rispetto a tecniche più semplici infatti per grandi quantità di dati il tempo di elaborazione è un aspetto da considerare. Un ulteriore aspetto critico della funzione si presenta nel momento in cui l'implementazione del range dentro il quale stimare i flussi risulta più ampia dei dati stessi quindi in quei punti possono presentarsi delle anomalie nella stima dei flussi, lo si vede dalla figura 19 .

- *griddatan*: è molto simile alle caratteristiche di *scatteredInterpolant* in quanto si utilizza per dati su griglie irregolari. La caratteristica principale è che stima i nuovi valori solo all'interno dei campioni dati in ingresso alla funzione, come mostrato in figura 20 i valori stimati sono all'interno del cerchio.
- *fit*: offre vari modelli di adattamento, come polinomi o modelli spline. Questo permette a seconda dei dati presenti di utilizzare una vasta possibilità di modelli così da adattarsi ad diversi casi. L'aspetto più critico risulta principalmente l'accuratezza che dipende fortemente dalla scelta del modello e che può non essere sempre intuitiva. Offre un compromesso tra precisione e continuità delle derivate, risultando adatta per modelli che richiedono transizioni fluide. Dalla figura 21 si può vedere come proprio per questo motivo non crea un'ottima interpolazione.

L'interpolazione che rispecchia al meglio le nostre richieste è *griddatan* ovvero la figura 20 perchè interpola seguendo le non linearità in modo preciso nei punti in cui vogliamo operare. A questo punto si visualizzano le non linearità presenti lungo d e q attraverso le figure 22 23, in più si può vedere l'effetto del cross-coupling in quanto sono

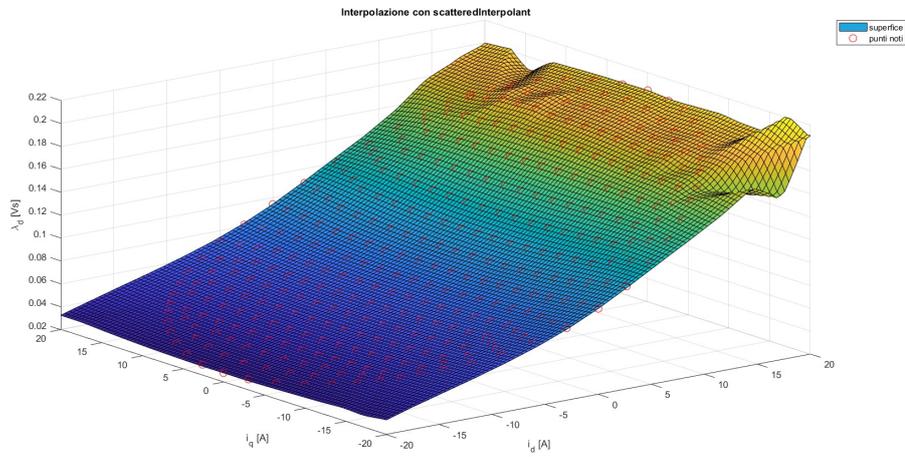


Figura 19: Interpolazione con scatteredInterpolant.

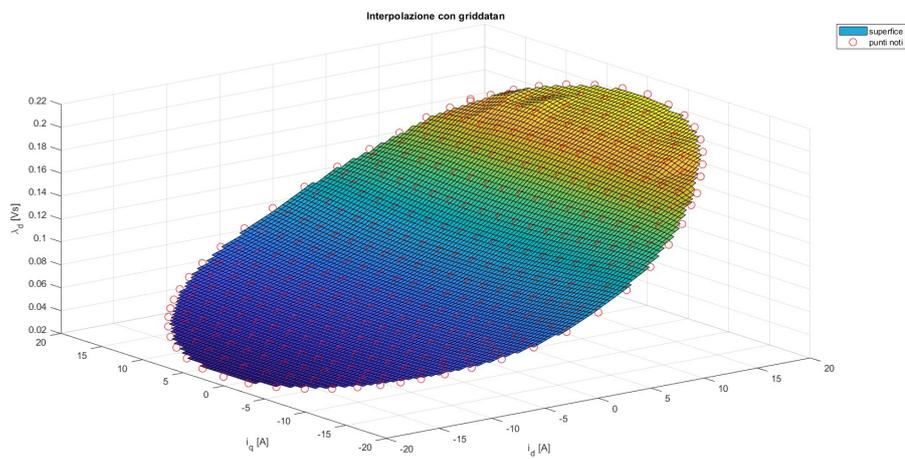


Figura 20: Interpolazione con griddatan.

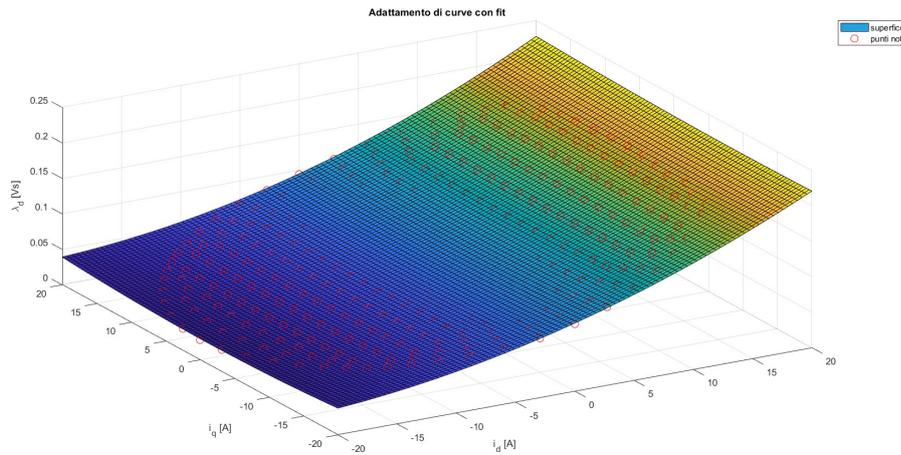


Figura 21: Adattamento di curve con fit.

presenti diverse linee di livello che mostrano come entrambi i flussi sono influenzati da entrambe le correnti.

4.0.3 Inversione mappe di flusso

L'inversione delle mappe di flusso ovvero il passaggio dalle relazioni $\lambda_d = f(i_d, i_q)$ e $\lambda_q = g(i_d, i_q)$ a delle nuove funzioni $i_d = h(\lambda_d, \lambda_q)$ e $i_q = k(\lambda_d, \lambda_q)$. La conoscenza di queste relazioni servono per creare uno schema del motore tanto più simile alla realtà di qualsiasi polinomio. In più queste nuove mappe servono per il controllo da inserire ogni qual volta che si richiede un punto di corrente diverso ottengo λ_d e λ_q . Per invertire le mappe di flusso si sfrutta lo stesso processo di interpolazione utilizzato precedentemente in quanto l'idea alla base è quella che ogni valore di i_d genera una λ_d e una λ_q allo stesso modo vale per i_q . Per cui è possibile vedere la relazione al contrario ovvero una coppia conosciuta λ_d λ_q ottengo una i_d conosciuta e con la stessa coppia di flussi corrisponde una i_q . Dal punto di vista operativo avrò tante coppie λ_d e λ_q quante la dimensione dei vettori di corrente. Questi punti non saranno disposti in modo omogeneo nello spazio per cui si crea una griglia omogenea e più fitta dove ricavare per altre coppie di flusso le rispettive correnti. Il nuovo dominio è composto da tutte le coppie di flusso note come mostrato in figura 24.

A questo punto si sfrutta lo stesso principio di interpolazione utilizzato per creare la superficie della mappa diretta. Si utilizza quindi la funzione *griddatan* così da lavorare solo nelle zone conosciute di punti. I risultati ottenuti sono

Queste relazioni sono poi caricate all'interno di LUT (Lookup Table) presenti nel controllo del banco prova dove si richiede in ingresso il punto di lavoro in corrente e restituisce in uscita i rispettivi flussi.

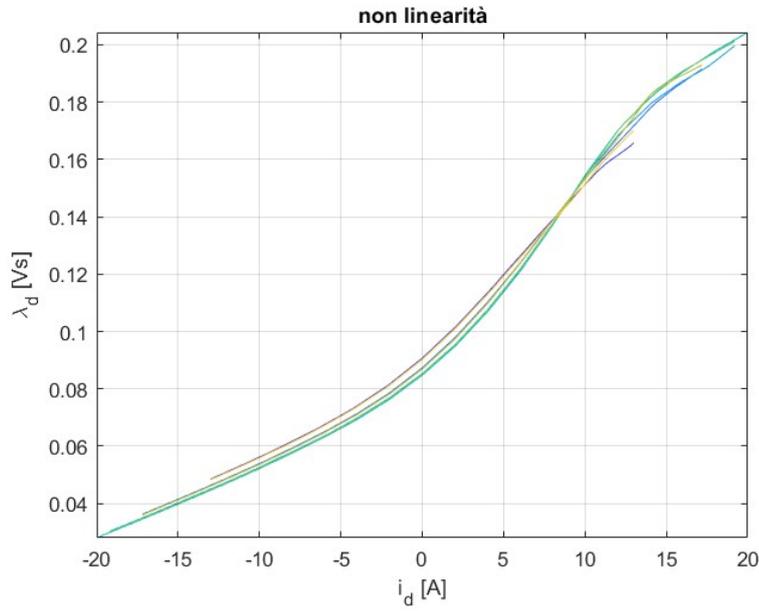


Figura 22: Presenza di non linearità lungo d.

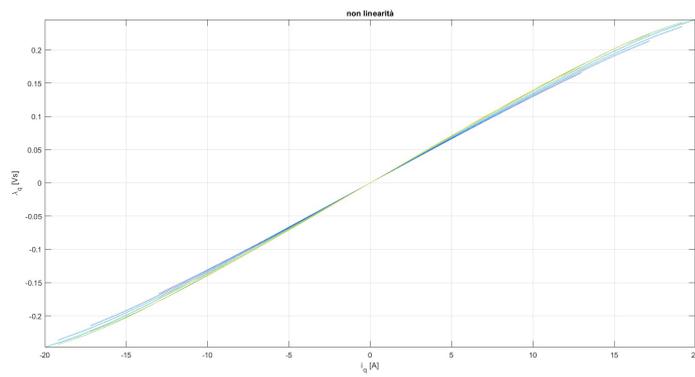


Figura 23: Presenza di non linearità lungo q.

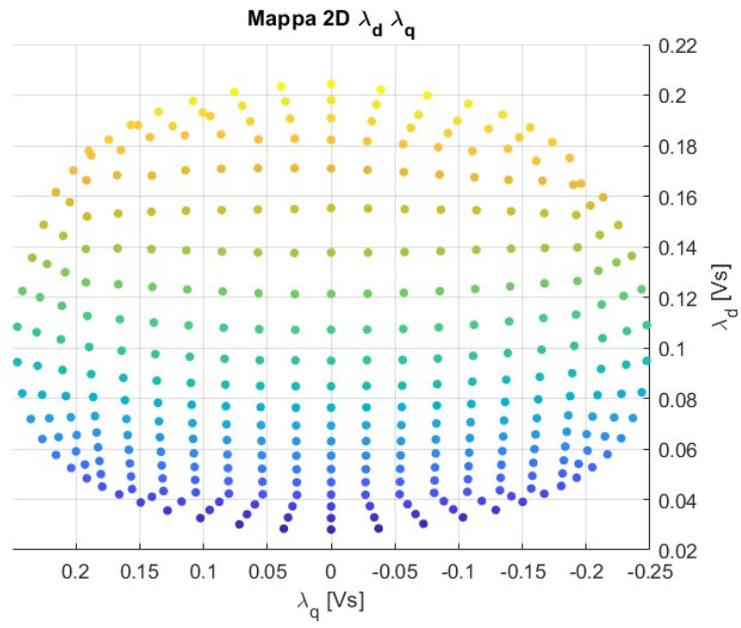


Figura 24: Coppie di flusso ricavate dall'esperienza.

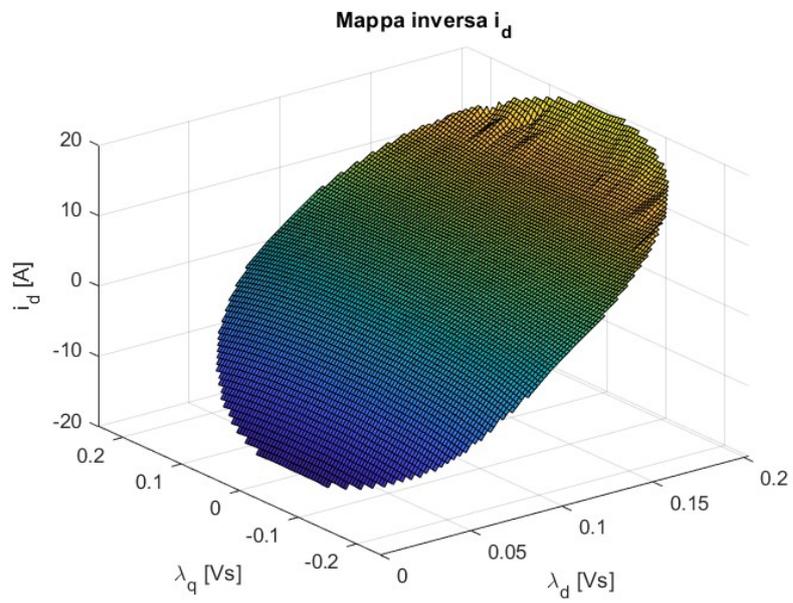


Figura 25: Relazione inversa $i_d = h(\lambda_d, \lambda_q)$.

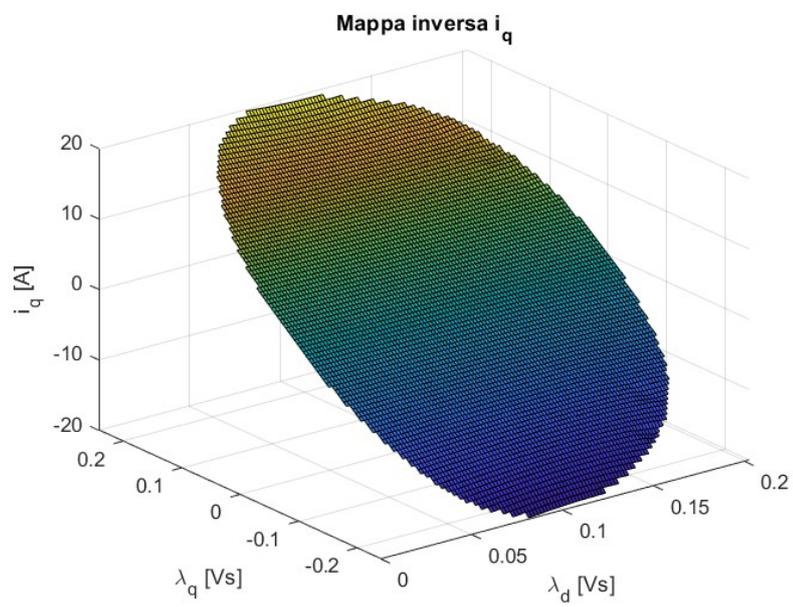


Figura 26: Relazione inversa $i_q = k(\lambda_d, \lambda_q)$.

MISURA DEL RENDIMENTO

Dopo aver ricavato le mappe di flusso, averle invertite e caricate nel controllo del banco prova, si è sviluppato un processo dove si misura l'efficienza del motore. Alla base di questo nuovo processo si utilizza oltre al banco prova anche un acquisitore dati. L'obiettivo di questa seconda parte è quello di creare una mappa di rendimento del motore ovvero una mappa dove si può vedere al variare della coppia e velocità quanto vale in un certo punto di lavoro. Il processo di sviluppo di una nuova procedura è stato molto simile al precedente in quanto prima si è creato un modello simulato e poi si è implementato quel processo al banco prova. Conoscere la mappa di efficienza permette di ottimizzare il funzionamento dell'azionamento, minimizzando le perdite.

5.0.1 Parte teorica

Alla base di questo processo ci sono diversi aspetti teorici da tenere in considerazione. In primis il concetto di efficienza in un motore elettrico è il rapporto tra potenza meccanica erogata all'albero e potenza elettrica assorbita dal motore [24](#)

$$\eta = \frac{P_{mecc}}{P_{elettr}} \quad (23)$$

dove:

- P_{mecc} è il prodotto tra la coppia misurata all'albero T e la velocità angolare meccanica dell'albero ω_m

$$P_{mecc} = T \cdot \omega_m \quad (24)$$

- P_{elettr} è la potenza elettrica in ingresso al motore (in uscita dall'inverter)

Il rendimento di un motore è influenzato anche da perdite di potenza durante la conversione dell'energia come:

- *Perdite Joule*: perdite dovute alla resistenza presente negli avvolgimenti
- *Perdite nel ferro*: dovute all'isteresi e alle correnti parassite presenti nel nucleo ferromagnetico del motore
- *Perdite meccaniche*: dovute all'attrito dei cuscinetti e a delle imperfezioni meccaniche

Ricapitolando, per misurare il rendimento sono necessari le misure di corrente, tensione, coppia, velocità. Un'ulteriore aspetto da tenere in considerazione è la misura di potenza elettrica in ingresso al motore in quanto si presenta con un segnale PWM (Pulse Width Modulation) perchè in uscita dall'inverter. Il segnale PWM è un segnale a onda quadra, quindi se lo si scompone attraverso la trasformata di Fourier è possibile vedere come è composto da una fondamentale e altre componenti ad alta tensione. L'obiettivo è quello di misurare la potenza elettrica che va al motore e per far ciò è necessario eliminare le componenti ad alta tensione attraverso un filtro passa basso.

5.0.2 *Procedura teorica*

Per generare questa nuova mappa di efficienza basta conoscere i termini che compongono la 24. Per far ciò è necessario ricavare i dati quando il processo è in steady state, questo è principalmente dovuto al campionamento di valori costanti mentre se il segnale è mobile risulta più complicato la corretta misura del valore. Per fare ciò è necessario quindi creare un zona di lavoro, cioè tutti i punti su cui operare. L'idea di base è creare una mappa rettangolare dove lungo l'ascissa è presente la velocità mentre lungo l'ordinata la coppia. Da tenere in considerazione anche il passaggio da un punto ad un altro, in quanto l'ordine è fondamentale questo perchè spostarsi prima lungo la coppia o prima lungo la velocità non è lo stesso. Lo spostamento in direzione delle ordinate, mantenendo costante la velocità per cambiare la coppia, risulta migliore per l'azionamento in quanto la dinamica di coppia è più veloce rispetto a quella dell'anello di velocità. Dopo aver riscontrato questo aspetto la scelta va sullo spostamento lungo la coppia per poi cambiare velocità quando gli intervalli di coppia sono terminati.

5.0.3 *Parte sperimentale*

La parte sperimentale è stata come detto precedentemente suddivisa in più parti come nel processo precedente. La suddivisione è stata la stessa ovvero prima la parte di simulazione poi la parte implementativa. La differenza rispetto al progetto precedente risulta nel momento in cui durante l'implementazione sul banco prova è stato necessario introdurre nel sistema un acquirente dati oltre alla dSpace già presente nel banco prova. Questo per avere un ulteriore riscontro dei dati ottenuti, con questo nuovo strumento la simulazione si limitava ad osservare gli andamenti dei riferimenti e dei vari trigger. Mentre l'implementazione per quanto riguarda l'aspetto pratico del banco prova in più si utilizza un acquirente dati il quale dovrà essere sincronizzato con tutto il resto del processo.

5.0.3.1 Simulazione del processo

La simulazione ha il compito di ricreare le condizioni che si sviluppano durante il processo reale. Per questo processo come nel precedente il compito è ricreare due motori accoppiati, dove uno è comandato in coppia mentre l'altro è comandato in velocità. Rispetto al precedente sono presenti delle modifiche:

- *velocità non costante*: mentre nel processo precedente la velocità era costante ora la velocità deve variare in modo tale da completare tutti i punti di lavoro della mappa
- *riferimento di coppia*: prima la matlab function produceva direttamente dei riferimenti di corrente mentre ora produce dei riferimenti di coppia che vanno poi convertiti attraverso le mappe di flusso in riferimenti di corrente.
- *presenza di un trigger*: un ulteriore output utilizzato per sincronizzare l'acquisitore dati con il processo

Anche in questo processo si è utilizzata una macchina a stati presente all'interno della matlab function in dove se possibile si riutilizzano le funzioni create per il processo precedente o con delle leggere modifiche. Il principio di funzionamento è molto simile, è suddiviso in due stati principali:

- *creazione della mappa + assegnazione del nuovo punto di lavoro*: come per il caso precedente la mappa verrà scelta dall'utilizzatore secondo i propri requisiti o valori nominali presenti nei dati di targa del motore. Questa operazione verrà chiaramente fatta una volta per poi a attraverso un'altra funzione avviene il passaggio dei punti di lavoro ogni volta che termina la misura.
- *generare il riferimento + generare il trigger*: in seguito allo stato precedente, l'obiettivo è raggiungere il punto di lavoro voluto. Questa operazione è svolta in modo tale che il raggiungimento del punto voluto non avvenga in contemporanea. Come detto in precedenza prima si raggiunge il punto di velocità e dopo averlo raggiunto ci si sposta lungo la coppia in modo separato. Questa separazione è un comportamento voluto appositamente per far sì che il motore non incrementi troppo la corrente per un breve periodo ma in modo graduale. Raggiunto il punto di lavoro avviene l'attivazione del trigger ovvero un segnale che passa da un livello basso ad alto così da attivare il campionamento da parte dell'acquisitore dei valori che servono per ricavare il rendimento. Terminato un certo periodo temporale utile all'acquisitore per ricavarsi un valore affidabile, dopodiché si attiva la richiesta di un nuovo punto di lavoro allo stato precedente così da far ripartire il ciclo fino all'esaurimento di tutti i punti.

Ora è il momento di spiegare in modo più preciso che compito hanno i vari case. A partire dal *case 1* dove si utilizza la funzione *creamappa-due*. La seguente funzione ha come ingressi:

- intervalli lungo l'asse d
- intervalli lungo l'asse q
- la coppia τ massima e la velocità ω massima

mentre in uscita:

- matrice dei riferimenti τ
- matrice dei riferimenti ω

L'obbiettivo è quello di creare una mappa rettangolare con tutti i punti di lavoro a seconda del numero di divisioni scelte come input che si vogliono creare lungo le due direzioni. Queste mappa prima di passare al al case successivo vengono riportate lungo dei vettori singoli per lo stesso motivo spiegato nel capitolo precedente. Nel *case 2* si trova la funzione *scanner-punti-due* che ha il compito di restituire ogni volta che termina l'acquisizione dati il nuovo punto. Segue il *case 3* il quale attraverso la funzione *rampa* incrementa la coppia τ dal punto di lavoro precedente a quello ricevuto dal *case 2*. Infatti come riportato precedentemente prima si incrementa la coppia poi la velocità. Tra l'incremento di coppia e quello di velocità ci si assicura di essere in steady state infatti questo è svolto dalla funzione rimasta invariata *steadystate*. Nel momento in cui questa condizione è confermata allora avviene l'incremento a forma di rampa pure della velocità. Nel momento in cui passo da un punto ad un altro dove la velocità rimane invariata questo case viene svolto solo con un tempo ciclo.

A questo punto del programma è presente una svolta rispetto al programma precedente il quale iniziava a campionare. In questo caso nel *case 8* il compito è attivare il trigger che avrà un duty cycle del 50%. Questa forma d'onda dovrà avere una durata temporale maggiore di 1[sec] infatti questo parametro lo si può cambiare dello script contenente tutti i parametri. La scelta è quella che il questo tempo duri 3[sec]. Il fatto di avere una certa durata di campionamento serve per sincronizzare l'acquisitore dati con il processo perchè solo dopo un secondo l'ingresso che legge il trigger risulta di nuovo attivo. Nel momento in cui questa durata è inferiore al secondo non avviene la sincronizzazione e di conseguenza non si campionano tutti i punti e la mappa risulta incompleta.

Il salvataggio dei dati non è un compito di questo programma in quanto sarà l'acquisitore a salvare in un file apposito tutti i valori. A questo punto del programma si ritorna al *case 2* dove avviene il passaggio del nuovo punto di lavoro. Questo processo si completa in modo ciclico fin tanto che non si conclude la mappa dei punti. Poi sono

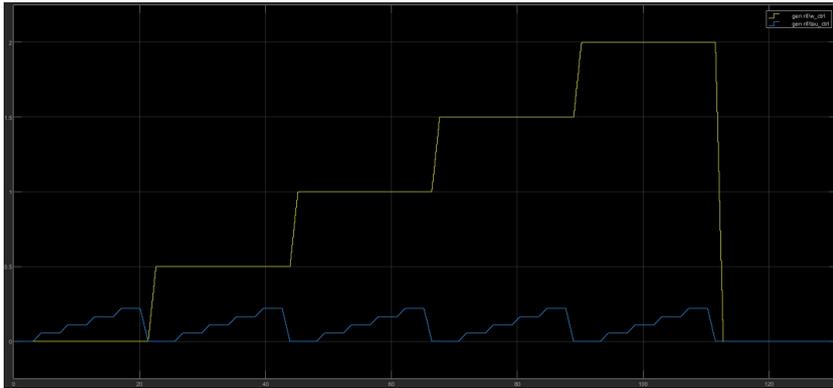


Figura 27: Andamento dei riferimenti di coppia e velocità.

presenti il *case 8* e il *case 9* attraverso delle rampe decrescenti a portare coppia e velocità a zero chiaramente in momenti separati.

I risultati ottenuti dalla simulazione dove sono presenti 4 intervalli lungo ω e τ con valori massimi rispettivamente di $2[\frac{rad}{s}]$ e $0,22[Nm]$ sono riportati in figura 27, dove si può notare che mentre il valore di velocità risulta fisso quello di coppia cambia.

5.0.3.2 Implementazione sul banco prova

Dal punto di vista applicativo rispetto alla simulazione sono presenti diverse aggiunte. Il progetto ha lo scopo di ricavarsi il rendimento ma attraverso un acquirente esterno al banco prova. Nonostante il banco prova possiede la possibilità di misurare tutte le grandezze utili alla misura del rendimento, l'utilizzo di un acquirente esterno ad alta precisione permette di confermare il risultato. La misura comprende diversi apparecchi: banco prova, dSpace, acquirente. Dal punto di vista applicativo:

- *dal computer alla dSpace*: come spiegato nel capitolo precedente il processo è lo stesso ovvero si carica la matlab function all'interno del controlDesk così da collegare visione real time con la dSpace che avrà il compito di generare i segnali presenti in uscita dalla funzione.
- *dalla dSpace al banco prova*: attraverso la dSpace è possibile selezionare quali segnali posso essere disponibili attraverso i DAC presenti all'interno del banco prova. In questo caso è necessario che i segnali di velocità, coppia e trigger disponibili in modo analogico.
- *acquirente*: è stato utilizzato un WT5000 della Yokogawa che dovrà combinare la misura di potenza che passa da inverter a motore e le misure di coppia e velocità presenti ai nei DAC del bancoprova. Tutto ciò è sincronizzato dal trigger che comanda l'acquisizione dei dati.

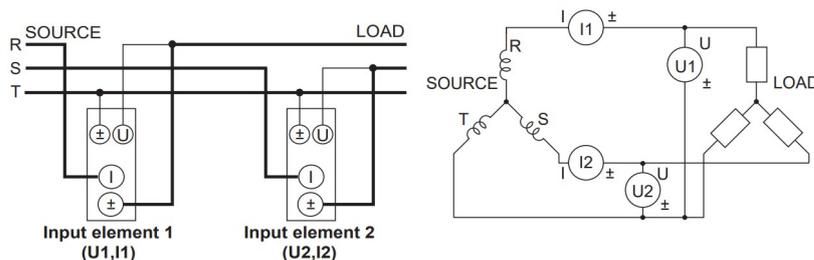


Figura 28: Schema di cablaggio.

Fondamentale è stato lo studio per capire il funzionamento del WT5000, caratteristiche di campionamento, le modalità di salvataggio dati, il significato dei vari menù. Infatti il WT5000 della Yokogawa è un wattmetro di precisione di alta gamma progettato per misurare con estrema accuratezza la potenza elettrica in vari contesti, come test di efficienza energetica di inverter, motori, apparecchiature industriali e sistemi di alimentazione. Lo schema di funzionamento è riportato in figura

Prima di iniziare la prova è necessario configurare il sistema:

1. *Configurare ogni elemento*: per elemento si intende quel dispositivo che svolge il compito di misurare tensione e corrente. Si interfacciano tra grandezza fisica e grandezza elettrica.
 - a) *impostare il sistema di cablaggio*: in uscita dall'inverter sono presenti tre fasi disposte a stella dalle quali è necessario misurare la potenza secondo lo schema presente in figura 28. Attraverso il menù *Wiring setting* è stato selezionato *3P3W* ovvero lo schema per l'inserzione Aaron.
 - b) *impostare i valori di range*: per ogni elemento è necessario impostare il range entro il quale lavorare. In tensione $600V$ in corrente $30A$.
 - c) *impostare i filtri di linea*: è stato attivato il filtro di Anti-aliasing dove la frequenza di taglio non si imposta ma è fissa a $1MHz$. Poi anche un filtro passa basso con frequenza di taglio di $300Hz$ in quanto il valore di velocità massima in questo esperimento è $4500rpm$ cioè $75Hz$ abbiamo una tempo ciclo dell'inverter di $10kHz$ per cui la frequenza di taglio del filtro è di $300Hz$ in quanto deve essere 4-5 volte la frequenza della fondamentale.
2. *Configurare l'aggiornamento dei campioni*: è possibile selezionare la sorgente del campionamento e i tempi di aggiornamento così da sincronizzare il riferimento al trigger.
 - a) *sorgente di campionamento*: dal menù *update mode* è possibile selezionare qual è la sorgente per aggiornare il valore della misura, in questo progetto è

stato utilizzato un segnale proveniente da un DAC il quale doveva essere un segnale da 0V a 5V con duty cycle del 50%.

- b) *impostazione del trigger*: la sorgente del trigger è l'external clock ovvero un ingresso apposito con attacco BNC e dove il segnale doveva avere le caratteristiche precedenti. In più si è selezionato che l'aggiornamento partiva nel fronte di salita del segnale.
 - c) *intervallo temporale di aggiornamento*: il valore minimo di aggiornamento del trigger è di 1s perciò è stato scelto un tempo di 3s tra un fronte ed un altro.
 - d) *salvataggio dati*: il salvataggio dei dati è avvenuto in una chiavetta dove è possibile caricare i campioni delle grandezze selezionate precedentemente in un altro menù.
3. *Configurare i sensori che provengono dal motore*: infine è stato necessario settare gli ingressi per i segnali provenienti dai sensori di coppia e velocità presenti nell'albero.
- a) *tipo di ingresso*: in questo caso entrambi i sensori producevano un'uscita analogica perchè generata dal DAC. Per cui bastava attivare la modalità *analog* così da attivare i due ingressi.
 - b) *range di lavoro e scalatura*: il range è per entrambi 10V mentre per convertire il segnale da tensione in V a valore numerico in *rpm* e allo stesso modo per il valore di coppia in *Nm* è stata necessaria una relazione lineare $y = Ax + B$ tra le due grandezze come mostrato in figura 29 e dove il valore di $B = 0$ per entrambi mentre il valore di A è rispettivamente $A = 495 \frac{rpm}{V}$ e $A = 1.175 \frac{Nm}{V}$.

5.0.3.3 Risultati ottenuti dalla prova sperimentale

La prova è stata sviluppata per 13 intervalli lungo entrambi gli assi, dove il valore massimo di coppia è 7.6Nm mentre di velocità è 4500rpm. I risultati ottenuti si trovano su una mappa di efficienza dove il valore massimo si trova a coppie e velocità elevate ovvero dove le perdite nel ferro sono basse come mostra la figura 30 ricavata dalla prova. Come mostrato dalla figura 30 è possibile notare quattro casi:

- *bassa coppia, bassa velocità*: le perdite nel ferro sono presenti ma in bassa parte mentre le perdite sul rame non dipendono dalla velocità perciò influenzano maggiormente le basse potenze in ingresso.

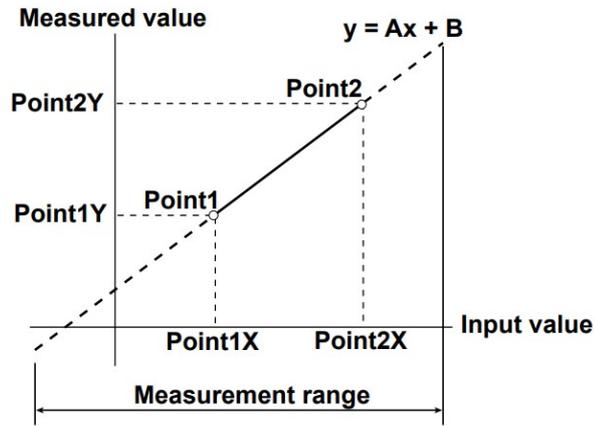


Figura 29: Relazione lineare per scalare il valore del sensore.

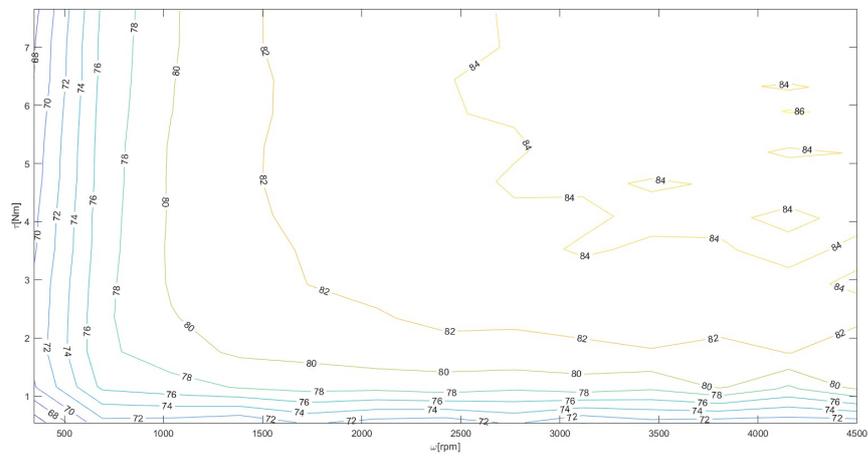


Figura 30: Mappa di efficienza.

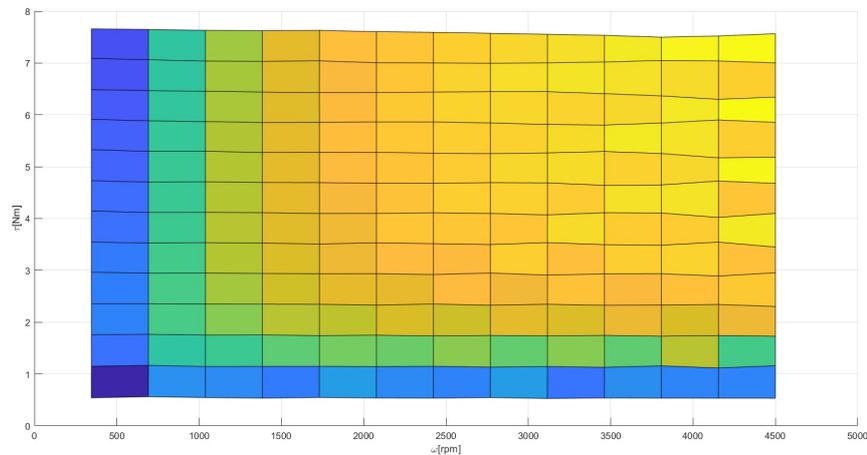


Figura 31: Mappa di efficienza con punti campionati.

- *bassa coppia, elevata velocità*: le perdite nel rame risultano meno influenti ma aumentano le perdite nel ferro in quanto proporzionali al quadrato della velocità. La potenza erogata è ancora bassa ma comunque influenzata dalle perdite sia per isteresi che per correnti parassite.
- *elevata coppia, bassa velocità*: l'effetto delle perdite nel ferro è basso ma sono presenti le perdite nel rame in quanto la coppia è elevata e quindi aumenta la richiesta di corrente.
- *elevata coppia, elevata velocità*: in questo caso il valore di efficienza aumenta in quanto entrambe le perdite saturano il loro valore ma aumenta la potenza in ingresso.

Un ulteriore aspetto da tenere in considerazione è l'errore di misura, dal momento che il valore di riferimento esce dalla matlab function la dSpace lo converte in segnale dove il controllore lo fa raggiungere. A questo punto il segnale viene campionato e convertito poi in valore effettivo. Si osserva dalla figura 31 che i valori campionati non creano una griglia perfettamente omogenea. L'errore ad alte velocità aumenta questo comporta che il punto di lavoro non si trovi esattamente nel punto voluto.

CONCLUSIONI

Questo metodo di caratterizzazione del motore risulta una semplice soluzione ad un problema che alle volte può risultare complesso. La sua forza sta proprio nella corretta stima dei flussi sfruttando un'attrezzatura modesta che non richiede particolari strumenti. Questo metodo è stato verificato confrontando i dati ottenuti attraverso delle prove compiute in precedenza con quelle attuali e il risultato è lo stesso. Un'ulteriore sviluppo può essere capire dopo una prima stima delle mappe dove sono le zone particolarmente non lineari e aumentare il numero di campioni proprio in quella zona. In modo tale da avere una migliore interpolazione in quei punti così poi da rendere più veritiera anche l'inversione.

APPENDICE



PROGRAMMI

Listing A.1: Codice generatore riferimenti di corrente.

```
1
2 function [i_d_ctrl, i_q_ctrl, flux_d, flux_q, flux_d_calc, flux_q_calc ] =
   generatore( Misura, intervallid, intervalliq, currmax, param, mot)
3
4 persistent stato_macchina
5
6 persistent en_scanner indice_trapezio indice_scanner i_d_ref i_q_ref
   vettore_riga_d_senzaNan vettore_riga_q_senzaNan
7
8 persistent i_d_ctrl_kml i_q_ctrl_kml indice_epsilon vett_velocita
   indice_vett_velocita
9
10 persistent misura_sum th_m_start indice_campiona valore_medio
11
12 persistent vettore_flusso_d vettore_flusso_q vettore_errore_flusso_d
   vettore_errore_flusso_q
13
14 %ingressi
15 misura.id = Misura(1);
16 misura.iq = Misura(2);
17 misura.th_m = Misura(3);
18 misura.w_m = Misura(4);
19 misura.tor = Misura(5);
20 misura.ud = Misura(6);
21 misura.uq = Misura(7);
22
23 %uscite
24 i_d_ctrl=0; i_q_ctrl=0; flux_d=0; flux_q=0; flux_d_calc = 0; flux_q_calc = 0;
25
26
27 termine_mappa = 0;
28 termine_rampa_d = 0;
29 termine_rampa_q = 0;
30 Tc=param.Tc;
31 T_rampa = param.T_rampa;
32
33 w_rif = mot.param.initialSpeed;
34 N_filtro = param.N_campioni_filtro;
```

```
35 epsilon = 1*w_rif/100;
36 ss = 0;
37
38 termine_misura = 0;
39
40
41
42
43 if isempty(stato_macchina)
44
45     stato_macchina = 1;
46
47     en_scanner = 1;
48     indice_trapezio = 1;
49     indice_scanner = 1;
50     i_d_ref = 0;
51     i_q_ref = 0;
52     vettore_riga_d_senzaNan = zeros();
53     vettore_riga_q_senzaNan = zeros();
54
55     vettore_errore_flusso_d = zeros();
56     vettore_errore_flusso_q = zeros();
57
58     i_d_ctrl_km1 = 0;
59     i_q_ctrl_km1 = 0;
60
61
62     indice_epsilon = 1;
63     vett_velocita = zeros(1, N_filtro);
64     indice_vett_velocita = 1;
65
66     misura_sum.id = 0;
67     misura_sum.iq = 0;
68     misura_sum.th_m = 0;
69     misura_sum.w_m = 0;
70     misura_sum.tor = 0;
71     misura_sum.ud = 0;
72     misura_sum.uq = 0;
73
74     indice_campiona = 1;
75
76     campiPrincipali = {'id', 'iq', 'w_m', 'tor', 'ud', 'uq'};
77     sottoCampi = {'x1', 'x2', 'x3'};
78     valori = 0;
79
80     valore_medio = struct();
81
```

```

82     for i = 1:length(campiPrincipali)
83         campoPrincipale = campiPrincipali{i};
84         for j = 1:length(sottoCampi)
85             sottoCampo = sottoCampi{j};
86             valore_medio.(campoPrincipale).(sottoCampo) = valori;
87         end
88     end
89
90     th_m_start = 0;
91
92     vettore_flusso_d = zeros();
93     vettore_flusso_q = zeros();
94
95 end
96
97
98
99
100 switch stato_macchina
101 case 1
102
103     [md, mq] = creamappa(intervallid, intervalliq, currmass );
104
105     vettore_riga_d = reshape(md, 1, []);
106     vettore_riga_q = reshape(mq, 1, []);
107
108     vettore_riga_d_senzaNan = rimuoviNaN(vettore_riga_d);
109     vettore_riga_q_senzaNan = rimuoviNaN(vettore_riga_q);
110
111     vettore_flusso_d = zeros(size(vettore_riga_d_senzaNan));
112     vettore_flusso_q = zeros(size(vettore_riga_q_senzaNan));
113
114     vettore_errore_flusso_d = zeros(size(vettore_riga_d_senzaNan));
115     vettore_errore_flusso_q = zeros(size(vettore_riga_d_senzaNan));
116
117     stato_macchina = 2;
118
119
120 case 2
121
122     [i_d_ref, i_q_ref, termine_mappa] = scanner_punti_due(vettore_riga_d_senzaNan,
123         vettore_riga_q_senzaNan, en_scanner, indice_trapezio, indice_scanner);
124
125     if(termine_mappa == 1)
126         plot3(vettore_riga_d_senzaNan, vettore_riga_q_senzaNan, vettore_flusso_d,
127             vettore_flusso_q);

```

```
127
128     stato_macchina = 8;
129
130 else
131
132     en_scanner = 0;
133
134     stato_macchina = 3;
135
136 end
137
138
139
140
141 case 3
142
143 [i_d_ctrl, termine_rampa_d]=rampa_due(T_rampa, 0, i_d_ref,i_d_ctrl_kml, Tc);
144 [i_q_ctrl, termine_rampa_q]=rampa_due(T_rampa, 0, i_q_ref,i_q_ctrl_kml, Tc);
145
146 if(termine_rampa_d == 1 && termine_rampa_q == 1)
147
148     i_d_ctrl = i_d_ref;
149     i_q_ctrl = i_q_ref;
150     stato_macchina = 4;
151
152 else
153
154     i_d_ctrl_kml = i_d_ctrl;
155     i_q_ctrl_kml = i_q_ctrl;
156
157 end
158
159
160 case 4
161
162 i_d_ctrl = i_d_ref;
163 i_q_ctrl = i_q_ref;
164
165 [ss, indice_epsilon, vett_velocita, indice_vett_velocita ] = steady_state_due(
    misura.w_m, w_rif, N_filtro, epsilon, indice_epsilon, vett_velocita,
    indice_vett_velocita );
166
167 if(ss == 1)
168
169     stato_macchina = 5;
170
171     th_m_start = misura.th_m;
```

```
172
173     indice_epsilon = 0;
174
175     indice_vett_velocita = 1;
176
177 else
178
179     stato_macchina = 4;
180
181 end
182
183 case 5
184
185 i_d_ctrl = i_d_ref;
186 i_q_ctrl = i_q_ref;
187
188 [termine_misura, valore_medio, misura_sum, indice_campiona ] = campiona_due(misura
    , misura_sum, th_m_start, indice_campiona, indice_trapezio, valore_medio);
189
190 if(termine_misura == 1)
191
192     stato_macchina = 6;
193
194 else
195
196     stato_macchina = 5;
197
198 end
199
200 case 6
201
202 [i_d_ctrl, termine_rampa_d]=rampa_due(T_rampa, i_d_ref, 0,i_d_ctrl_kml, Tc);
203 [i_q_ctrl, termine_rampa_q]=rampa_due(T_rampa, i_q_ref, 0,i_q_ctrl_kml, Tc);
204
205 if(termine_rampa_d == 1 && termine_rampa_q == 1)
206
207     i_d_ctrl = 0;
208     i_q_ctrl = 0;
209
210     stato_macchina = 7;
211
212     indice_trapezio = indice_trapezio + 1;
213
214 else
215
216     i_d_ctrl_kml = i_d_ctrl;
217     i_q_ctrl_kml = i_q_ctrl;
```

```
218
219 end
220
221
222
223 case 7
224
225 if(indice_trapezio == 4)
226
227     en_scanner = 1;
228
229     [flux_d, flux_q, flux_d_calc, flux_q_calc] = calcola_flusso(valore_medio,
230         i_d_ref, i_q_ref, mot);
231
232     vettore_flusso_d(indice_scanner) = flux_d;
233     vettore_flusso_q(indice_scanner) = flux_q;
234
235     err_d = flux_d - flux_d_calc;
236     err_q = flux_q - flux_q_calc;
237
238     vettore_errore_flusso_d(indice_scanner) = err_d;
239     vettore_errore_flusso_q(indice_scanner) = err_q;
240
241     indice_trapezio = 1;
242
243     indice_scanner = indice_scanner + 1;
244
245 end
246
247     stato_macchina = 2;
248
249 case 8
250
251 return;
252
253 end
254
255
256 end
```