



Università degli studi di Padova
Facoltà di Ingegneria
Laurea magistrale Ingegneria Elettrica

***CONTROLLO DIRETTO DI FLUSSO E DI COPPIA PER MOTORI
BRUSHLESS SPM/IPM***

Relatore: Ch.mo Prof. Silverio Bolognani

Laureando: Valeriano Veronese
Matricola n° 626430/IT

Anno Accademico: 2011

....Ad Emanuele, Mamma Daniela, Papà Franco...

INDICE

-Introduzione

1-Motore SPM

1.1-Introduzione

1.2-Principio di funzionamento

1.3-Equazioni e modello del motore SPM

2-Azionamento in corrente per motore SPM

2.1-Introduzione

2.2-Controllo in corrente

3-Un nuovo metodo di controllo: il DTC

3.1-Introduzione

3.2-Il Direct Torque Control

3.3-DTC switching table-self control

4-Formulazione del DTC per un motore SPM

4.1-Introduzione

4.2-Equazioni SPM

4.3-Principi teorici

4.4-DTC per SPM principio di funzionamento

4.5-Adattamento criteri di stabilità per SPM

4.6-Calcolo della derivata della coppia

4.7-Calcolo della derivata del modulo del flusso di statore

5-Simulazioni

5.1-Introduzione

5.2-Presentazione motore

5.3-Controllo puro

5.4-Controllo di velocità

5.5-Controllo dell'errore

5.6-Controllo stati inverter

6-Sperimentazione

6.1-Introduzione

6.2-Funzione unica

6.3-Controllo predittivo

6.4-Passaggio SPM-IPM

6.5-Implementazione in C

6.6-Commenti Simulato-Reale

Conclusioni

Bibliografia

INTRODUZIONE:

Se ci soffermiamo un attimo a valutare i tipi di motori impiegati in ambito industriale e civile notiamo subito che nella maggior parte dei casi sono di tipo elettrico, inoltre se consideriamo uno scenario internazionale notiamo che tutti i paesi industrializzati orientano la loro ricerca nell'impiego di questo tipo di motore anche nel settore trasporti sia di tipo pubblico che privato.

La macchina elettrica in genere presenta quindi un largo utilizzo destinato a crescere nel tempo. Di pari passo aumentano anche le prestazioni richieste in termini di potenza, di prontezza e precisione nel controllo.

Una tipologia di macchina che sta trovando un utilizzo sempre più diffuso è il motore BrushLess (letteralmente "senza spazzole") che viene brevemente introdotto di seguito.

E' una macchina sincrona che riceve in ingresso energia elettrica e in uscita fornisce energia meccanica presenta varie caratteristiche che la rendono appetibile a partire dalla massima rapidità di ciclo: un servomotore BL (inteso come motore e azionamento) può invertire il senso di rotazione in pochi millesimi di secondo. Presenta inoltre un'ottima controllabilità, un'alta precisione nel posizionamento, marcia regolare a basse velocità, rapidissime accelerazioni o decelerazioni, ricordiamo anche, derivante dal nome: la mancanza di contatti striscianti che diventa garanzia di lunga durata e distanza notevolmente la periodicità dell'assistenza programmata. Consideriamo anche che il BL ha una buona capacità di sovraccarico, alta coppia, può sopportare una forte sollecitazione impulsiva ed è di ingombri ridotti.

Tutto questo però a scapito di costi abbastanza elevati e della presenza indispensabile dell'azionamento; in quanto un BL deve sempre essere accompagnato dal relativo apparato di controllo.

In questa trattazione andremo a studiare un motore BL per trazione ferroviaria, il quale, sottoposto a correnti e tensioni relativamente ingenti comporterà delle difficoltà nell'applicare il consueto controllo in corrente, si dovrà quindi adoperare un nuovo controllo di coppia, chiamato Direct Torque Control, applicabile sia a macchine con rotore a magneti interni (Interior Permanent Magnet: IPM) che a magneti superficiali (Surface-mounted Permanent Magnet: SPM).

La macchina e la struttura dell'azionamento sono stati modellizzati tramite il tool di matlab simulink, per poi simulare il comportamento dinamico e le risposte del sistema complessivo. Successivamente si è passati alla fase di sperimentazione pratica per analizzare nella realtà le caratteristiche del controllo.

1-IL MOTORE SINCRONO SPM: (SURFACE-MOUNTED PERMANENT MAGNET)

1.1-Introduzione:

In questo capitolo andremo ad analizzare la struttura e il principio di funzionamento di questo tipo di macchina, in modo da ricavare delle equazioni che ne descrivono il comportamento in termini dinamici, per poi ottenere un modello rappresentativo del motore, che ben si presta ad essere simulato ed implementato.

1.2-Principio di funzionamento

Il principio di funzionamento si basa, analogamente a qualsiasi macchina elettrica di tipo elettromagnetico, sulle interazioni tra campi magnetici e correnti creati da, o aventi sede, su rotore e- o statore. La sostanziale peculiarità consiste nel fatto di avere dei magneti permanenti (PM: materiali ferromagnetici duri, caratterizzati da un ciclo di isteresi ampio e alta magnetizzazione residua) nel rotore che creano il flusso magnetico a traferro.

Lo statore sostanzialmente è costruito nello stesso modo di quello di un motore asincrono, con i conduttori dentro le cave che costituiscono l'avvolgimento, il quale, percorso da corrente, crea il campo magnetico di statore (la cui direzione è data dalle solite regole elettrotecniche) che sommato con quello creato dai magneti, interagisce con le correnti stesse e fa nascere delle forze sui conduttori, e pertanto momenti torcenti, che mettono in rotazione il rotore.

Il rotore dove sono inseriti i PM può essere istropo e quindi il campo magnetico creato da una fase non viene influenzato dalla posizione del rotore, (il quale non ha alcuna direzione preferenziale per il flusso), anisotropo se invece siamo nel caso opposto e quindi il rotore influenza con la sua posizione, il campo creato dalle fasi statoriche.

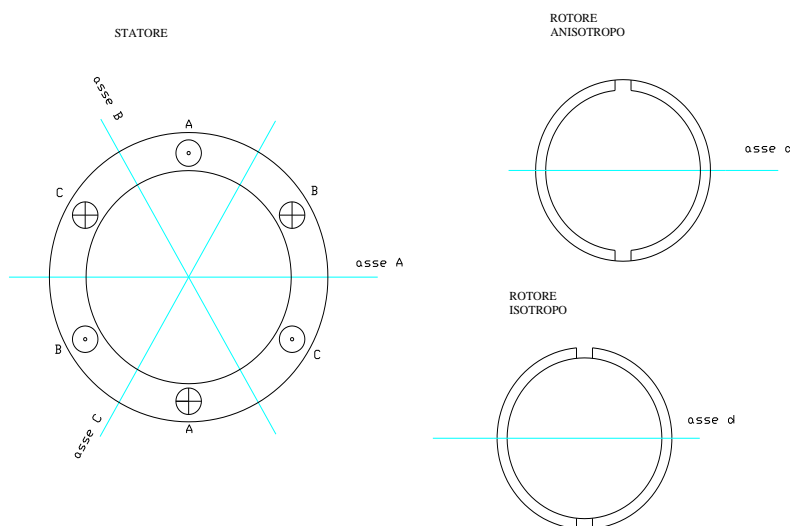


Fig 1.1 rappresentazione schematica di statore e rotore con esempi di rotore isotropo e anisotropo (2p)

La macchina che si prenderà in esame è costituita da un rotore di tipo isotropo, infatti i magneti sono incollati o fissati sulla circonferenza esterna del rotore (rotore SPM). In questo modo, essendo la permeabilità magnetica (differenziale) dei magneti circa uguale a quella dell'aria, il campo magnetico che viene a crearsi a traferro per effetto delle correnti statori che incontra una riluttanza magnetica circa costante al variare della posizione angolare.

1.3-Equazioni e modello del motore SPM:

Per capire meglio il funzionamento analizziamo ora le equazioni del motore, idealizzando la macchina imponendo assenza di correnti parassite e perdite nel ferro.

Sappiamo dall'elettrotecnica che la tensione prelevata ai capi di ciascun avvolgimento di fase (schematizzato come un' unica spira, es: A-A di fig 1.1) è data, con la convenzione di segno degli utilizzatori, da: caduta di tensione resistiva sugli avvolgimenti sommata alla variazione temporale del flusso concatenato dalla fase stessa (forza contro-elettromotrice):

$$u_a = Ri_a + \frac{d\lambda_a}{dt}$$

$$u_b = Ri_b + \frac{d\lambda_b}{dt}$$

$$u_c = Ri_c + \frac{d\lambda_c}{dt}$$

è una terna in cui manca la componente omopolare, usando quindi i vettori spaziali posso compattare:

$$\overline{u_{\alpha\beta}} = R\overline{i_{\alpha\beta}} + \frac{d\overline{\lambda_{\alpha\beta}}}{dt}$$

Le cui componenti sono:

$$u_\alpha = Ri_\alpha + \frac{d\lambda_\alpha}{dt}$$

$$u_\beta = Ri_\beta + \frac{d\lambda_\beta}{dt}$$

Il flusso però è dato da due contributi: flusso dei PM e flusso dato dalle correnti statoriche. I due interagiscono e si combinano tra di loro.

In particolare, applicando il principio di sovrapposizione degli effetti scriviamo l'andamento del flusso ipotizzando la sola presenza dei PM (a vuoto); il flusso concatenato su ciascuna fase avrà quindi un andamento cosinusoidale, in quanto il magnete, girando assieme al rotore presenta un asse polare (asse d) variabile nel tempo.

Infatti considerando fase a per esempio: quando l'asse polare del rotore è a 90° (θ_{me}) il flusso concatenato dalla fase è nullo, mentre quando l'asse polare del rotore è a 0° il flusso è massimo:

$$\lambda_{mg_a} = \Lambda_{mg} \cos \theta_{me}$$

$$\lambda_{mg_b} = \Lambda_{mg} \cos \left(\theta_{me} - \frac{2\Pi}{3} \right)$$

$$\lambda_{mg_c} = \Lambda_{mg} \cos \left(\theta_{me} - \frac{4\Pi}{3} \right)$$

Dove:

$$\vartheta_{me} = p\theta_m$$

Anche qui uso la notazione in vettore spaziale:

$$\overline{\lambda_{mg \alpha\beta}} = \Lambda_{mg} e^{j\theta_{me}}$$

Questo vettore spaziale rappresenta le grandezze in un sistema di riferimento α - β stazionario, l'osservatore è quindi fermo rispetto alle grandezze che variano nel tempo. Se invece adottiamo un sistema di riferimento diverso, poniamo per esempio l'osservatore sul rotore, ciò equivale a moltiplicare i termini delle equazioni per un operatore $e^{-j\theta_{me}}$ (da stazionario a rotante, da rotante a stazionario tolgo il -). In pratica non vedo più le grandezze sul rotore variare in modo cosinusoidale ma diventano costanti, sistema di riferimento d-q

Il flusso quindi:

$$\overline{\lambda_{dq}} = \overline{\lambda_{\alpha\beta}} e^{-j\theta_{me}} = \Lambda_{mg}$$

Ricaviamo quindi le equazioni di tensione della macchina scritte in precedenza per un sistema di riferimento stazionario, ora adattiamole per le corrispondenti coordinate rotanti utilizzando l'opportuno operatore matematico

$$\overline{u_{dq}} e^{j\theta_{me}} = R \overline{i_{dq}} e^{j\theta_{me}} + \frac{d(\overline{\lambda_{dq}} e^{j\theta_{me}})}{dt}$$

Sviluppando la derivata e dividendo tutto per $e^{j\theta_{me}}$

$$\overline{u_{dq}} = R \overline{i_{dq}} + \frac{d\overline{\lambda_{dq}}}{dt} + j\omega_{me} \overline{\lambda_{dq}}$$

Scomponendola in parte reale e parte immaginaria si ottiene:

$$u_d = R i_d + \frac{d\lambda_d}{dt} - \omega_{me} \lambda_q$$

$$u_q = R i_q + \frac{d\lambda_q}{dt} + \omega_{me} \lambda_d$$

Andiamo ora a considerare l'effetto delle sole correnti sugli avvolgimenti, ipotizzando dapprima un motore con rotore isotropo e circuito ferromagnetico lineare.

Con queste correnti nei tre avvolgimenti ho i classici auto e mutui concatenamenti delle linee di campo con le spire, esprimibile rispettivamente per ciascuna fase come:

$$\lambda_{a,i} = L_a i_a + L_{Mab} i_b + L_{Mac} i_c$$

$$\lambda_{b,i} = L_{Mba} i_a + L_b i_b + L_{Mbc} i_c$$

$$\lambda_{c,i} = L_{Mca} i_a + L_{Mcb} i_b + L_c i_c$$

Osserviamo i termini:

Date le ipotesi assunte, gli auto concatenamenti, rappresentati dai termini: L_a L_b L_c , non dipendono né dalle correnti, né dalla posizione angolare del rotore, possiamo quindi assumerli tutti e tre costanti e pari a L_s .

Se consideriamo i mutui concatenamenti, rappresentati dai termini: L_{Mab} L_{Mac} L_{Mba} L_{Mbc} L_{Mcb} L_{Mca} , data la disposizione ciclica e uniforme, e assumendo sempre rispettate le ipotesi precedenti, posso considerarli tutti uguali tra loro e pari a L_{Mss} . Ricordiamo però che questi mutui accoppiamenti, relativi a flussi di diverse fasi hanno segno negativo:

$$L_{Mss} = -|L_{Mss}|$$

Quindi scriviamo per la fase a:

$$\lambda_{a,i} = L_{ss} i_a + L_{Mss} (i_b + i_c)$$

Se non ho il neutro, caso usuale nella pratica:

$$i_a + i_b + i_c = 0 \quad \rightarrow \quad i_b + i_c = -i_a$$

Allora

$$\lambda_{a,i} = (L_{ss} - L_{Mss}) i_a = (L_{ss} + |L_{Mss}|) i_a = L i_a$$

$$\lambda_{b,i} = L i_b$$

$$\lambda_{c,i} = Li_c$$

Andiamo ora ad unire i due contributi e possiamo scrivere che una quota parte del flusso è dovuta alle correnti, mentre l'altra è data dai PM:

$$\lambda_{a,b,c} = Li_{a,b,c} + \lambda_{mg_{a,b,c}}$$

Quindi:

$$\lambda_a = Li_a + \Lambda_{mg} \cos \theta_{me}$$

$$\lambda_b = Li_b + \Lambda_{mg} \cos \left(\theta_{me} - \frac{2\Pi}{3} \right)$$

$$\lambda_c = Li_c + \Lambda_{mg} \cos \left(\theta_{me} - \frac{4\Pi}{3} \right)$$

Trovata l'espressione del flusso andiamo a sostituirla nelle equazioni iniziali che descrivono gli andamenti delle varie grandezze nelle tre fasi del nostro motore:

$$u_a = Ri_a + \frac{d}{dt} (Li_a + \Lambda_{mg} \cos \theta_{me})$$

$$u_b = Ri_b + \frac{d}{dt} \left(Li_b + \Lambda_{mg} \cos \left(\theta_{me} - \frac{2\Pi}{3} \right) \right)$$

$$u_c = Ri_c + \frac{d}{dt} \left(Li_c + \Lambda_{mg} \cos \left(\theta_{me} - \frac{4\Pi}{3} \right) \right)$$

Usando i vettori spaziali:

$$\overline{u_{\alpha\beta}} = \overline{Ri_{\alpha\beta}} + \frac{d\overline{\lambda_{\alpha\beta}}}{dt} = \overline{Ri_{\alpha\beta}} + L \frac{d\overline{i_{\alpha\beta}}}{dt} + \frac{d\overline{\lambda_{mg\alpha\beta}}}{dt}$$

Osserviamo che c'è una f.e.m. dovuta alla variazione della corrente e una dovuta alla variazione dei flussi in particolare chiamiamo $e_{a,b,c}$ quest'ultima componente:

$$e_a = \frac{d\lambda_{mg,a}}{dt} = \omega_{me} \Lambda_{mg} \cos \left(\theta_{me} + \frac{\Pi}{2} \right)$$

$$e_b = \frac{d\lambda_{mg,b}}{dt} = \omega_{me} \Lambda_{mg} \cos \left(\theta_{me} + \frac{\Pi}{2} - \frac{2\Pi}{3} \right)$$

$$e_c = \frac{d\lambda_{mg,c}}{dt} = \omega_{me}\Lambda_{mg} \cos\left(\theta_{me} + \frac{\pi}{2} - \frac{4\pi}{3}\right)$$

Compattando con i vettori spaziali:

$$\overline{e_{\alpha\beta}} = \frac{d\overline{\lambda_{mg\alpha\beta}}}{dt} = j\omega_{me}(\Lambda_{mg}e^{j\theta_{me}}) = j\omega_{me}\overline{\lambda_{mg\alpha\beta}}$$

Quindi l'equazione iniziale:

$$\overline{u_{\alpha\beta}} = R\overline{i_{\alpha\beta}} + L\frac{d\overline{i_{\alpha\beta}}}{dt} + \frac{d\overline{\lambda_{mg\alpha\beta}}}{dt} = R\overline{i_{\alpha\beta}} + L\frac{d\overline{i_{\alpha\beta}}}{dt} + \overline{e_{\alpha\beta}}$$

Le cui componenti sono:

$$u_{\alpha} = Ri_{\alpha} + L\frac{di_{\alpha}}{dt} + e_{\alpha}$$

$$u_{\beta} = Ri_{\beta} + L\frac{di_{\beta}}{dt} + e_{\beta}$$

Cambio sistema di riferimento vado in d-q, moltiplicando e dividendo come al solito per l'operatore di rotazione; ottengo:

$$\overline{u_{dq}} = R\overline{i_{dq}} + L\frac{d\overline{i_{dq}}}{dt} + j\omega_{me}L\overline{i_{dq}} + j\omega_{me}\Lambda_{mg}$$

Le componenti:

$$u_d = Ri_d + L\frac{di_d}{dt} - \omega_{me}Li_q + 0$$

$$u_q = Ri_q + L\frac{di_q}{dt} + \omega_{me}Li_d + \omega_{me}\Lambda_{mg}$$

Abbiamo quindi ottenuto le due equazioni cardine del motore, esse rappresentano, in termini dinamici, il comportamento e le variazioni nel tempo delle varie grandezze che interessano la macchina.

Se rivediamo meglio le formule ricavate, applicando il principio di sovrapposizione degli effetti (solo magneti, poi solo d, successivamente solo q), osserviamo che:

su asse d del flusso corrispondente all'asse polare, abbiamo due contributi: quello dei PM stessi e quello creato da un corrente i_d generica che percorre avvolgimento fittizio (posto ovviamente con le spire sul piano individuato dall' asse q per creare flusso con asse magnetico lungo d): $Li_d + \Lambda_{mg}$

su asse q del flusso abbiamo esclusivamente il contributo di una corrente i_q generica su avvolgimento fittizio Li_q

Concludiamo quindi dicendo che con motore isotropo gli assi d e q vedono entrambi percorsi magnetici con la stessa riluttanza.

Moltiplicando da ambo le parti per la rispettiva componente di corrente ottengo un bilancio delle potenze:

$$u_d i_d + u_q i_q = R(i_d^2 + i_q^2) + i_d L \frac{di_d}{dt} + i_q L \frac{di_q}{dt} + \omega_{me} [(Li_d + \Lambda_{mg})i_q - Li_q i_d]$$

Ho quattro blocchi: potenza assorbita, perdite joule, variazione energia magnetica, potenza convertita.

Ricordando che la trasformazione adottata per il passaggio dal sistema di riferimento stazionario a quello sincrono non è invariante per la potenza, la quale risulta diminuita di $\frac{2}{3}$, possiamo eguagliare la quota parte relativa alla potenza elettrica convertita con la corrispondente grandezza meccanica:

$$m\omega = \frac{3}{2} \omega_{me} [\Lambda_{mg} i_q + (L - L) i_d i_q] = \frac{3}{2} \omega_{me} \Lambda_{mg} i_q$$

La coppia quindi diventa:

$$m = \frac{3}{2} p \Lambda_{mg} i_q$$

Soffermiamoci un attimo su questa espressione per fare ragionamenti che verranno utili nel seguito della trattazione.

Se scriviamo:

$$m = \frac{3}{2} p \Lambda_{mg} i_q = \frac{3}{2} p [\Lambda_{mg} i_q + (L - L) i_d i_q] = \frac{3}{2} p [(Li_d + \Lambda_{mg})i_q - Li_q i_d]$$

Introducendo le espressioni dei flussi:

$$\lambda_d = Li_d + \Lambda_{mg}$$

$$\lambda_q = Li_q$$

Otteniamo la formula:

$$m = \frac{3}{2} p (\lambda_d i_q - \lambda_q i_d)$$

la quale altri non è che l'espressione generale della coppia valida per qualsiasi tipo di macchina elettrica.

Abbiamo quindi ricavato per un motore SPM: sia le equazioni che rappresentano la dinamica elettrica, sia l'equazione della coppia elettromagnetica che mette in relazione grandezze elettriche con quella meccanica.

Queste sono equazioni differenziali nel dominio nel tempo, difficili da trattare matematicamente, per semplificare la problematica operiamo allora nel dominio della frequenza, consideriamo cioè nelle grandezze non il loro andamento nel tempo ma il loro spettro armonico in termini di fase e ampiezza. Per fare ciò adoperiamo le opportune trasformazioni matematiche, nel nostro caso Laplace.

Per operare questo passaggio matematico dobbiamo considerare ω_{me} costante, trascurando così la dinamica meccanica. Questa è una buona approssimazione in quanto solitamente i fenomeni elettrici del sistema si evolvono molto più rapidamente di quelli meccanici.

Così facendo le derivate e gli integrali si semplificano e le formule passano da differenziali ad algebriche:

$$u_d = Ri_d + sLi_d - \omega_{me}Li_q$$

$$u_q = Ri_q + sLi_q + \omega_{me}Li_d + \omega_{me}\Lambda_{mg}$$

Se poi prendiamo in esame anche l'equazione meccanica della coppia di carico:

$$m = B\omega + J \frac{d\omega}{dt} + m_l$$

la quale dice che la coppia creata dal motore deve essere in grado di controbilanciare: un termine di attrito $B\omega$ direttamente proporzionale alla velocità meccanica ω , un termine inerziale $J \frac{d\omega}{dt}$ legato alla variazione temporale di ω , e un termine che rappresenta la coppia di carico m_l generalmente funzione del tempo e del termine $B\omega$ stesso.

Anche qui passiamo nel dominio della frequenza con le L-trasformate:

$$m = B\omega + sJ\omega + m_l$$

Avendo allora identificato ciascuna equazione temporale nel suo equivalente in frequenza, possiamo analizzarne la dinamica, ricavando dai blocchi rappresentativi la funzione di trasferimento del sistema complessivo, rappresentabile in modulo e fase con i noti diagrammi di Bode.

Lo schema rappresentativo nel dominio della frequenza diventa quello in Fig.1.2, in esso compaiono elementi non lineari, quali i moltiplicatori.. Si noti l'accoppiamento incrociato degli assi d e q, espresso dalle quantità $\omega_{me}Li_d$ e $\omega_{me}Li_q$, che indicano come la tensione u_d , agendo su i_d , interferisca con l'azione di u_q nel determinare la corrente i_q e viceversa.

Se pensiamo ad un controllo in corrente per renderlo efficace dovremo pensare ad un disaccoppiamento tra i due canali di tensione. La cosa diventa invece inutile nel DTC visto che agisce direttamente su coppia e flusso e non su grandezze ad esse legate.

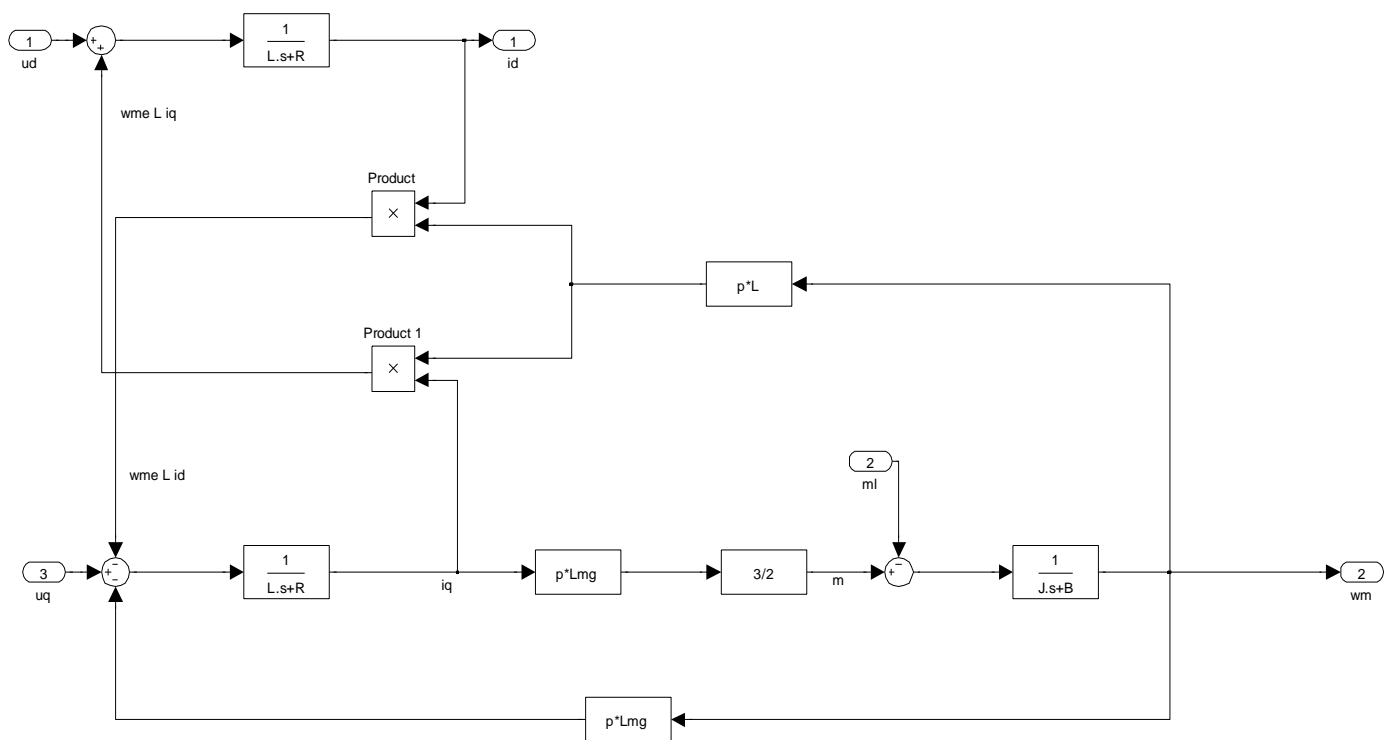


Fig 1.2 schema a blocchi del motore nel dominio della frequenza

2-AZIONAMENTO A CORRENTE IMPRESSA PER MOTORE SPM

2.1-Introduzione:

Dopo aver richiamato il motore e ottenute le sue equazioni di funzionamento andiamo ora a vedere com'è possibile comandare questa macchina, vogliamo cioè stabilire dei criteri e definire delle metodologie affinché il BrushLess risponda a delle specifiche richieste.

Procediamo dapprima con il classico metodo di controllo: l'azionamento a corrente impressa (o comandato in corrente).

2.2- Controllo in corrente

Come primo tentativo procediamo quindi costruendo l'azionamento nel solito modo seguendo la seguente filosofia: nella nostra applicazione andiamo a controllare le correnti le quali, essendo una legata alla coppia e l'altra al flusso, possiamo agevolmente regolare la grandezze meccaniche ed elettriche di nostro interesse.

Ricordiamo quindi come avviene questa procedura: la corrente in primo luogo viene trasdotta dalla macchina, in modo da avere informazioni da confrontare con la grandezza di riferimento da noi imposta. Comparando le due si ottiene un errore (nullo se e solo se la corrente della macchina sarà pari a quella da noi desiderata), bisogna a questo punto elaborare tramite un regolatore l'errore di corrente in modo da dare un segnale di riferimento di tensione all'inverter che agendo sulla tensione di alimentazione della macchina farà in modo che la corrente circolante diventi quella voluta.

Nota importante va fatta per quanto riguarda la limitazione delle grandezze in uscita dai regolatori, bisogna porre dei componenti limitatori per evitare che le correnti e le tensioni superino valori ammissibili dalla macchina ed inoltre non ci siano fenomeni fastidiosi di wind-up.

Lo schema di principio diventa quindi:

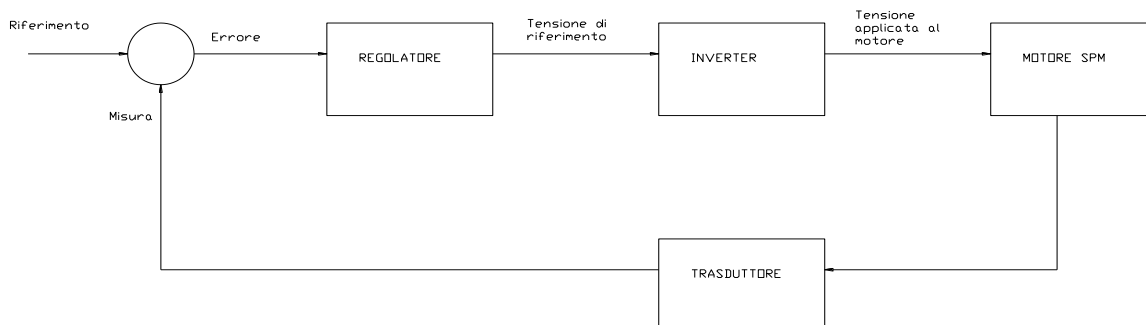


Fig.2.1 schema a blocchi dell'azionamento in corrente

Analizziamo ciascun componente un attimo più nel dettaglio, sempre con il fine di trovare una funzione di trasferimento che identifichi il rapporto ingresso-uscita nel dominio di Laplace.

Un trasduttore è un dispositivo destinato a “trasformare” una grandezza fisica alterandone in un segnale anche i valori numerici. Nella nostra applicazione il parametro di interesse è la corrente presente sulle fasi del motore, che va rilevata e trasmessa, eventualmente con valori differenti, al regolatore per essere confrontata con il valore di riferimento.

Il trasduttore può essere rappresentato in termini dinamici da un blocco che riceve in ingresso la corrente del motore, la trasduce, e la rende disponibile all’uscita, eventualmente con valori in scala.

Possiamo scrivere allora la funzione di trasferimento:

$$G_t = \frac{K_t}{1 + sT_t}$$

Dove K_t rappresenta il fattore di scala (guadagno) e T_t è la costante di tempo che rappresenta il ritardo della trasduzione. Possiamo però considerare quest’ultima quasi istantanea, senza ritardo, la f.d.t. diventa allora:

$$G_t = K_t$$

Il regolatore è il blocco dove possiamo intervenire per il controllo, solitamente è un classico PID che riceve l’errore di corrente in ingresso e in uscita dà un segnale di tensione che è la combinazione lineare dell’integrale, della derivata e di un addendo proporzionale dell’errore.

Analizzando un attimo più nel dettaglio queste tre componenti osserviamo che il PID regola l’uscita in base a: il valore del segnale d’errore (P), alla sua evoluzione negli istanti passati (I) e alla sua rapidità di variazione nel tempo (D).

In questo modo noi che andiamo a dimensionare il controllore riusciamo ad intervenire su queste 3 caratteristiche del segnale, determinando i vari coefficienti si modifica la risposta dinamica dell’intero sistema in modo da soddisfare a delle specifiche richieste.

La funzione di trasferimento di un PID generale è, operando in frequenza:

$$PID = R = K_p + \frac{K_i}{s} + sK_d$$

Con questo dispositivo operando in frequenza possiamo agire su parametri come banda passante e margine di fase in modo da avere poi nel tempo una risposta più o meno pronta, con una certa sovra elongazione ecc.

In questo caso occorre osservare che nella pratica si usano quasi sempre regolatori di tipo PI, per non amplificare l’errore che si presenta nei casi reali con il termine derivativo.

L’inverter PWM trifase è un convertitore di tensione da DC ad AC, ovvero il blocco che ricevendo un certo riferimento va a modificare le tensioni vere e proprie del motore in ampiezza e frequenza cambiando in modo opportuno gli stati degli interruttori bidirezionali di cui è composto.

Osserviamo lo schema di principio:

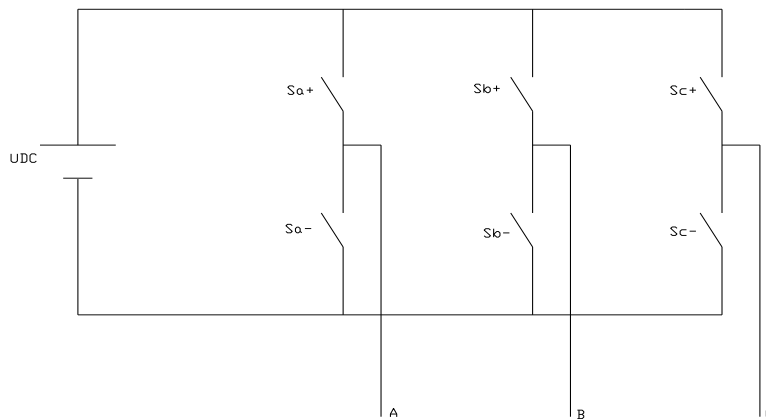


Fig 2.2 Schema di principio di un inverter trifase

Il principio di funzionamento si basa sulla commutazione degli interruttori statici bidirezionali presenti sui rami (o gambe) del dispositivo, questi cambi di stato determinano un livello di tensione positivo o negativo (rispetto alla mezzaria della tensione di alimentazione) presente sull'uscita della gamba, ottenuto collegando con un interruttore o l'altro il terminale del motore al polo positivo della sorgente di tensione continua o a quello negativo. Possiamo rappresentare lo stato di ciascuna gamba mediante uno stato logico, in particolare 1 se è chiuso Sa+ e 0 se è chiuso Sa-. Ne risulta quindi una sequenza binaria di tre cifre che identifica l'assetto dell' inverter.

Ciascun stato logico può essere abbinato da un vettore spaziale di tensione, di conseguenza, in funzione di quale e per quanto tempo è applicato si determina un valore specifico di tensione applicato sulle tre fasi A, B, C.

In termini dinamici possiamo rappresentare il tutto con un blocco che riceve in ingresso la tensione di riferimento, poi con un certo ritardo la elabora e produce in uscita la tensione con i parametri da noi voluti:

$$G_i = \frac{1}{1 + sT_i}$$

dove T_i rappresenta la costante di tempo che esprime il ritardo di generazione della tensione, dovuto alla natura discreta di controllo ottenuta tramite la commutazione degli interruttori statici.

Abbiamo ora le funzioni di trasferimento dei dispositivi, unendo il tutto con lo schema a blocchi del sistema motore-carico otteniamo:

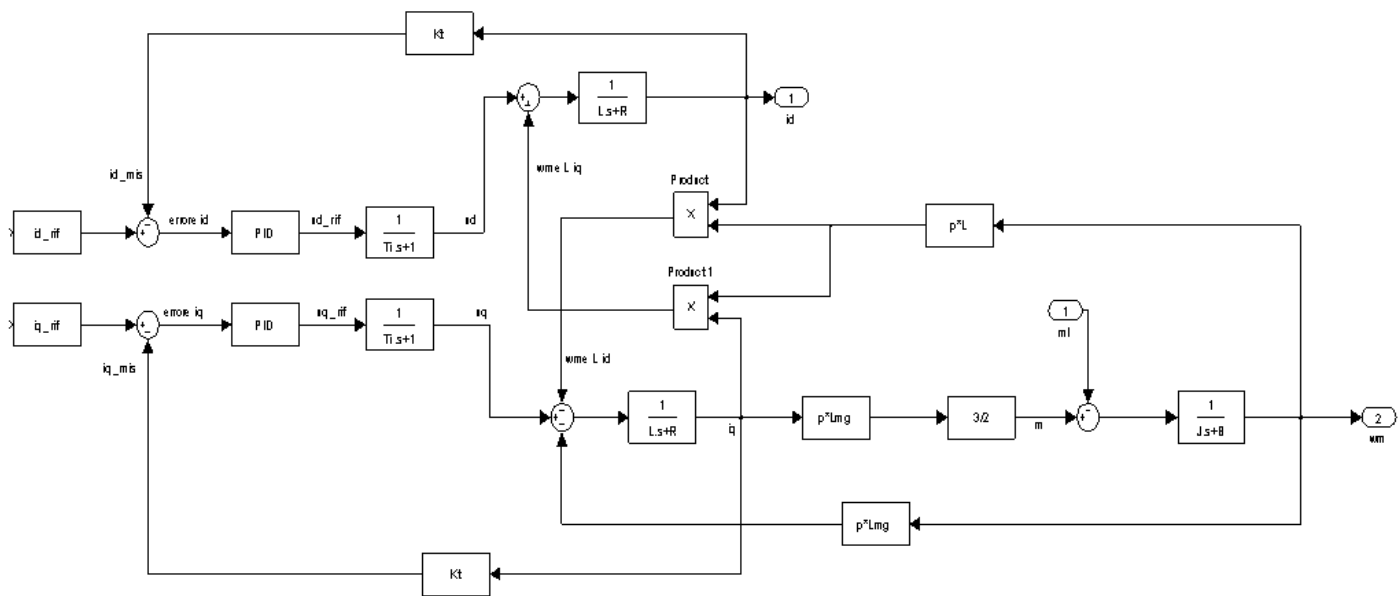


Fig 2.3 schema controllo in corrente

Osserviamo che per quanto riguarda il controllo e la regolazione delle grandezze in gioco possiamo intervenire esclusivamente sul PID, è lui che comanda l'inverter dando l'opportuna tensione di riferimento, i blocchi restanti fanno parte della parte di potenza, non possiamo modificare i parametri del motore o dell'inverter stesso perché sono propri del dispositivo.

Notiamo poi che le grandezze non sono tra di loro indipendenti, in particolar modo gli anelli di corrente: chiedo una variazione di i_q , cambia la ω_m , cambia u_d , cambia i_d , in altre parole una variazione del riferimento di i_q per esempio va a modificare anche il ramo i_d .

Possiamo ovviare a questo problema grazie alla seguente osservazione: essendo una macchina sincrona, la frequenza delle grandezze elettriche in gioco è strettamente legata alla velocità elettrica e di conseguenza meccanica.

A questo punto se consideriamo le grandezze elettriche del motore, caratterizzate da una dinamica lenta (visto che la velocità varia lentamente), e le confrontiamo con quelle attuate dall'inverter, notiamo subito che le seconde sono interessate da una costante di tempo breve, di conseguenza presentano una dinamica più veloce delle prime.

Questa diversa evoluzione dei parametri comporta che quelli interessati dalla velocità possano essere trattati come disturbi che possono essere facilmente compensati ed eliminati introducendo un opportuno segnale di tensione, abbiamo effettuato il cosiddetto disaccoppiamento degli assi

Dalle funzioni di trasferimento risultano poi spesso dei poli complessi coniugati, eliminabili tramite un ragionamento analogo al precedente, andando però ad agire sulla forza elettromotrice del motore e compensandola con un opportuno segnale.

In totale lo schema con le opportune compensazioni è il seguente:

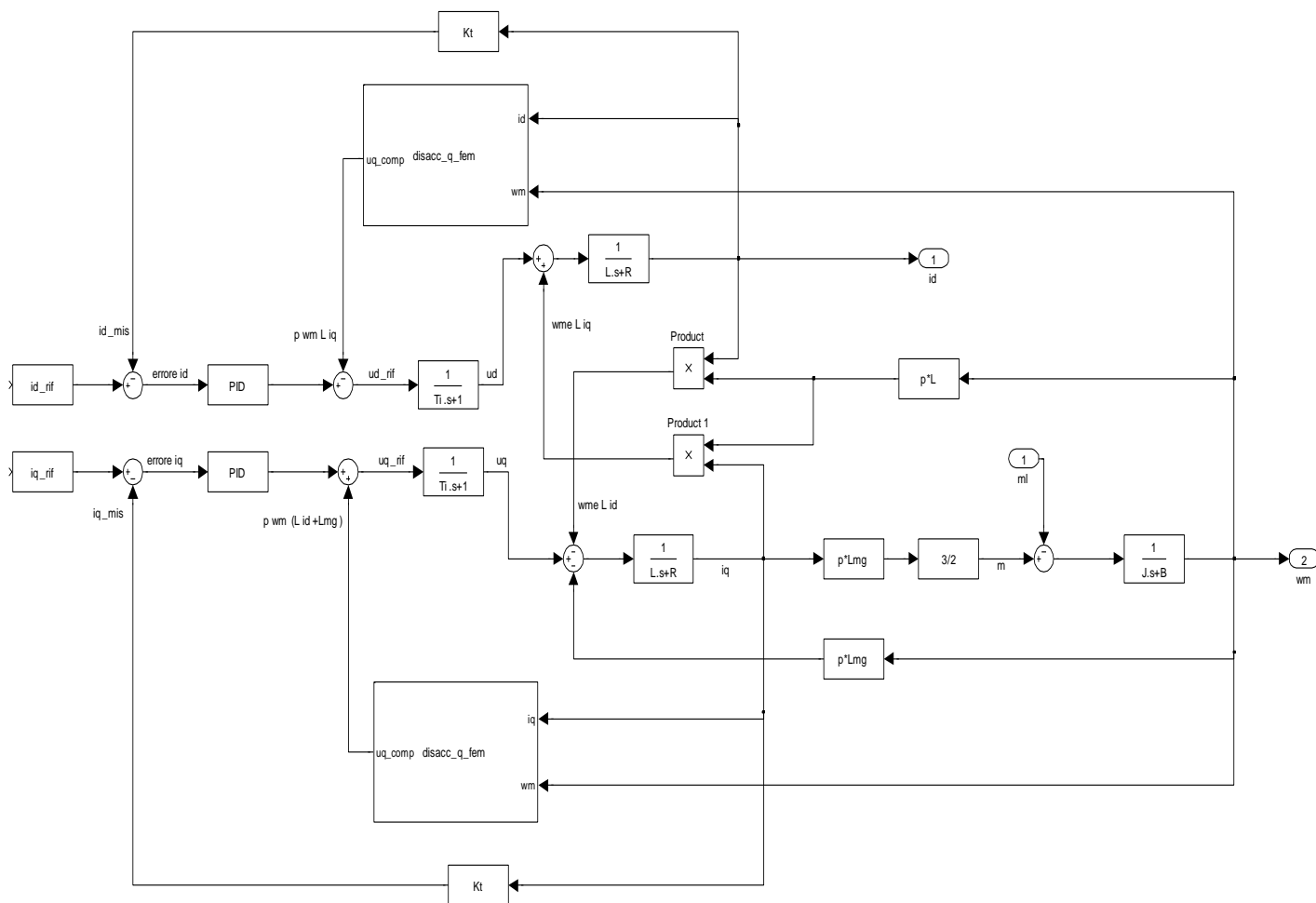


Fig 2.4 Schema complessivo

Ottenendo in tal modo lo schema completo che possiamo utilizzare come base per la simulazione con il programma simulink.

Ultima nota vogliamo farla per quel che riguarda la cosiddetta MTPA, usualmente infatti è conveniente far lavorare la macchina in modo che con una data corrente si abbia la massima coppia disponibile, la Max Torque Per Ampère appunto. Per quanto riguarda un SPM la coppia dipende solo dalla componente q della corrente, di conseguenza regolando esclusivamente la i_q abbiamo la coppia massima disponibile.

3-UN NUOVO METODO DI CONTROLLO: IL DTC (DIRECT TORQUE AND FLUX CONTROL)

3.1-Introduzione:

In questo capitolo andremo ad enunciare i principi generali del nuovo tipo di controllo, nato in origine per i motori asincroni trifasi e poi adattato ai motori brushless di nuova generazione, particularizzando il modo adeguato le equazioni e le formule a seconda del caso.

Accenneremo quindi alle varie strategie esistenti, comunque provenienti tutte da una base comune che andremo ad analizzare in modo da capire come funziona e qual è l'idea di base che porta il DTC.

3.2-Il Direct Torque Control:

Un azionamento elettrico presenta alcuni tratti caratteristici comuni, in particolare si pensa solitamente a costruire un controllo che soddisfi a due fondamentali specifiche: controllo del campo di induzione, e, di conseguenza controllo d'armatura per regolare la coppia.

Solitamente si perseguono questi due obiettivi cercando di riprodurre il comportamento dinamico di una macchina a corrente continua, lo si fa perché è relativamente semplice, e con due grandezze si riesce a comandare indipendentemente sia la coppia che il flusso di induzione. Questo è quanto si è pensato di fare infatti nei motori asincroni dove si ha l'azionamento ad orientamento di campo e nei BL, nei quali viene effettuato il disaccoppiamento degli assi delle componenti della corrente, che permette di controllare: con la componente diretta il campo e con quella in quadratura la coppia.

Alla metà degli anni '80 iniziano ad essere pubblicati vari studi su una nuova metodologia di controllo: Depenbrock in Germania e Takahashi e Noguchi in Giappone per esempio, sviluppano una nuova filosofia che sta alla base del controllo diretto di flusso e di coppia, o più semplicemente controllo diretto di coppia.

L'obiettivo che si erano proposti i ricercatori del periodo era: partendo da un motore asincrono, con relativo controllo FOC, riuscire a ridurre la complessità dei tipici algoritmi, in modo da ottenere un controllo di coppia e flusso più semplice e preciso. Per riuscire nella loro idea hanno pensato di cambiare totalmente ragionamento: invece di tentare di riprodurre il comportamento elettromeccanico di una macchina a corrente continua, hanno cercato di sfruttare al massimo la capacità di produrre coppia e flusso di un motore alimentato da un inverter.

La nuova tecnica di controllo risultò fin da subito caratterizzata da semplicità, buone prestazioni e robustezza, inoltre, altra caratteristica importante del controllo diretto di coppia è che si riesce a ottenere un buon controllo delle grandezze senza l'uso di alcun trasduttore posto sull'albero della macchina. Stiamo quindi per andare ad analizzare una tecnica di controllo di tipo sensorless con tutti i vantaggi che ne derivano: l'assenza di sensori garantisce un minor costo del dispositivo, inoltre visto che non sono previsti, i trasduttori non possono guastarsi, migliorando l'affidabilità dell'azionamento.

Ovviamente questo comporta un aumento della complessità del controllo che dovrà provvedere alla stima delle grandezze necessarie, in questo caso la coppia e il modulo del flusso di statore, tramite algoritmi stimatori di tipo deterministico o stocastico.

Andiamo ora a studiare il principio di funzionamento del controllo diretto di flusso e di coppia, o meglio le idee su cui si basa.

L'obiettivo è sempre quello di riuscire a controllare la coppia del motore e il flusso di induzione, considerando l'equazione generale della coppia vista al paragrafo 1.3 scritta per un sistema di riferimento rotante (comunque estendibile a qualsiasi sistema di riferimento) valida per tutti i tipi di motore elettrico:

$$m = \frac{3}{2}p(\lambda_d^r i_q^r - \lambda_q^r i_d^r) = \frac{3}{2}p|\lambda^r \times i^r| = \frac{3}{2}p|\lambda^r||i^r|\sin \alpha^r$$

La coppia è quindi proporzionale al modulo del flusso di statore e al seno dell'angolo compreso tra il vettore rappresentativo del flusso e il vettore rappresentativo della corrente visti da un sistema di riferimento rotante per esempio alla velocità di sincronismo.

Possiamo quindi osservare come la coppia possa essere modificata cambiando uno dei fattori presenti nella formula ossia: angolo, flusso, correnti.

Se ci spostiamo un attimo sulla regolazione del flusso notiamo che esso dipende dalla tensione applicata dalla macchina, in particolare se prendiamo l'equazione del motore nel sistema di riferimento stazionario, trascurando la caduta di tensione resistiva:

$$\overline{u^s} = \frac{d\overline{\lambda^s}}{dt}$$

Integrando da ambo le parti ottengo:

$$\overline{\lambda^s} = \int \overline{u^s} dt$$

Passando al sistema di riferimento rotante da noi usato:

$$\overline{\lambda^r} e^{j\theta_{me}} = \int \overline{u^r} e^{j\theta_{me}} dt$$

Quindi:

$$\overline{\lambda^r} = \frac{\int \overline{u^r} e^{j\theta_{me}} dt}{e^{j\theta_{me}}}$$

Anche il flusso nel sistema di riferimento rotante dipende dalla tensione applicata alla macchina, ovvero quella che dà l'inverter.

Come è evidenziato dalle formule si deduce quindi che modificando la tensione della macchina andiamo ad intervenire sul flusso e di conseguenza sulla coppia fornita dal motore.

Osserviamo che è possibile quindi intervenire sul flusso applicando una tensione piuttosto che un'altra, o meglio, ragionando con i vettori spaziali (indicati anche con v. s. nel seguito della trattazione): possiamo cambiare o meno il v.s del flusso applicando il corretto v.s. attivo in uscita dall'inverter (tra i 6 disponibili)

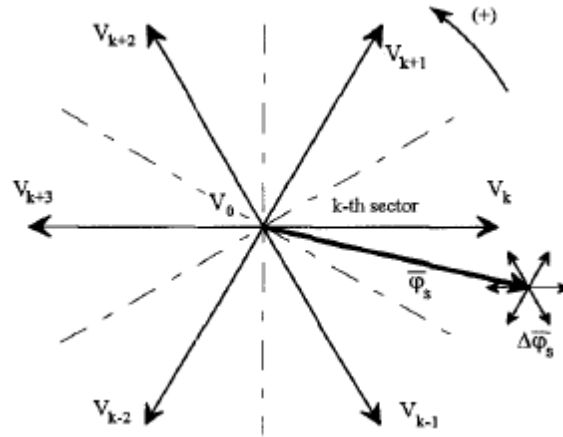


Fig. 3.1 Variazioni del flusso applicando diversi v.s.

Dalla figura osserviamo infatti che se per esempio applichiamo un v.s. contrario come $k + 3$ il flusso cala.

Ora ci si potrebbe chiedere l'uso degli stati nulli, per rispondere a questo quesito pensiamo di essere sul sistema di riferimento d-q rotante; a regime le grandezze ruoteranno alla velocità di sincronismo e appariranno costanti e ferme, poi applicando diversi v.s. di tensione si spostano, ma se pensiamo di applicare il vettore nullo delle tensioni, il vettore spaziale del flusso non cambia la sua ampiezza si sposta solamente, inizia a ruotare modificando l'angolo α^r , di conseguenza anche senza variare le ampiezze riesco comunque a comandare la coppia.

Visti i caratteri generali della filosofia di base di questo sistema di controllo passiamo ad analizzare mediante l'ausilio degli schemi a blocchi lo mschema generale di un DTC:

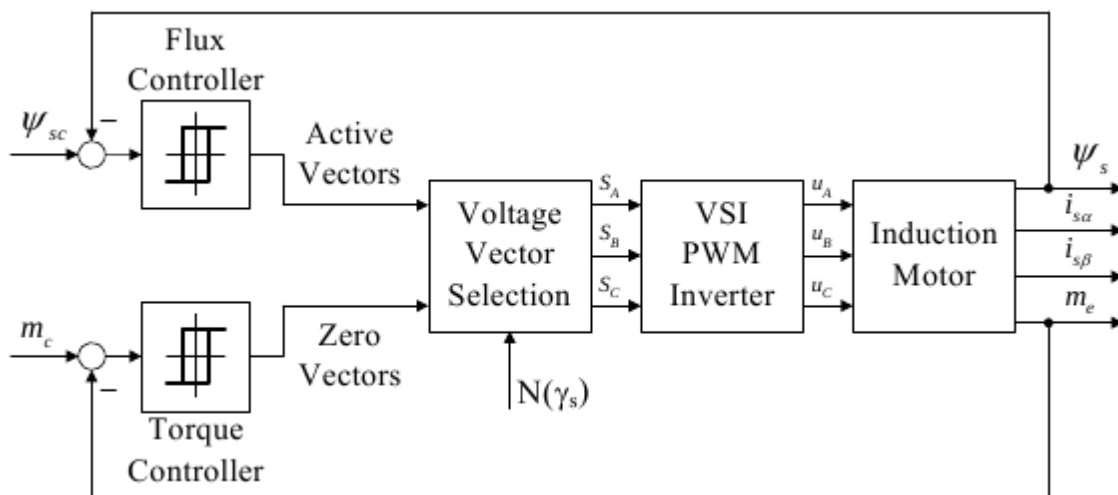


Fig. 3.2 schema a blocchi base di un DTC

Imponiamo i riferimenti di coppia e di flusso, li confrontiamo con i valori ricavati, per esempio tramite uno stimatore, dalle correnti. Abbiamo allora l'errore che viene valutato e, se maggiore di una certa soglia di tolleranza, entra in azione il controllo: ovvero ricevendo in ingresso gli errori di coppia e flusso, a seconda dei casi, l'algoritmo decide la selezione degli stati dell'inverter e di conseguenza le tensioni al motore

Queste idee brevemente descritte stanno alla base di due tipi di controllo: il DTC switching table e il DTC self-control, per la nostra applicazione viene utilizzata una ulteriore variante del DTC, che comunque poggia sulla base comune di controllare il flusso e di conseguenza la coppia mediante una scelta opportuna di valori di tensione forniti dall' inverter.

3.3-Switching table e self-control

Il controllo diretto di coppia basato su strategia switching table prevede che la selezione del vettore spaziale di tensione che l'inverter deve applicare al motore avvenga leggendo il valore mediante una tabella precaricata, memorizzata su una memoria ROM.

La costruzione della tabella si fa valutando dapprima la posizione del vettore di flusso e poi a seconda dell'entità dell'errore si scelgono gli stati che più rapidamente aumentano o calano i valori di coppia e/o flusso.

Il self-control invece sempre tramite lo stimatore fornisce i segnali di riferimento i quali vengono elaborati da un blocco modulatore che opera fornendo un vettore spaziale di tensione che garantisca al flusso di stare dentro una traiettoria esclusivamente esagonale.

4-FORMULAZIONE DEL DTC PER UN MOTORE SPM

4.1 Introduzione:

In questo capitolo andremo ad esporre una nuova formulazione del controllo diretto di flusso e di coppia per motori SPM e non IPM.

Anche se il motore analizzato è di tipo anisotropo procediamo adottando questa semplificazione che risulta utile in quanto lo studio della tipologia di motori a rotore isotropo ben si presta a considerazioni e valutazioni analitiche che potremo poi estendere al caso di nostro interesse.

Dopo questo passaggio verranno enunciati i principi teorici su cui si basa questo tipo di controllo, la cui validità ovviamente è generale. Dovremo allora adattare la struttura per la nostra tipologia di motore, spiegandone dapprima il funzionamento in termini descrittivi, e poi ricavando le equazioni che garantiscono la stabilità del sistema, con il conseguente annullamento dell'errore.

4.2 Equazioni fondamentali SPM:

Per avere le equazioni di un motore SPM basta riprendere la:

$$\overline{u_{dq}} = R\overline{i_{dq}} + L\frac{d\overline{i_{dq}}}{dt} + j\omega_{me}L\overline{i_{dq}} + j\omega_{me}\Lambda_{mg}$$

Scritta per un riferimento rotante sincrono con il rotore.

Le sue componenti sono:

$$u_d = Ri_d + L\frac{di_d}{dt} - \omega_{me}Li_q + 0$$

$$u_q = Ri_q + L\frac{di_q}{dt} + \omega_{me}Li_d + \omega_{me}\Lambda_{mg}$$

Oppure conoscendo il legame flussi correnti, supposto lineare:

$$\begin{cases} \lambda_q = Li_q \\ \lambda_d = Li_d + \Lambda_{mg} \end{cases}$$

otteniamo la ben nota:

$$\begin{cases} u_d = Ri_d + \frac{d}{dt}\lambda_d - \omega_{me}\lambda_q \\ u_q = Ri_q + L\frac{d}{dt}\lambda_q + \omega_{me}\lambda_d \end{cases}$$

Sono queste le formule che useremo come punto di partenza per sviluppare le equazioni che saranno necessarie, in quanto sono quelle che descrivono completamente la dinamica del motore.

4.3 Principi teorici

Introduciamo ora i concetti di teoria su cui si basa la nuova formulazione del controllo diretto di coppia e di flusso studiata in questa tesi.

Partiamo considerando il caso piú semplice possibile: un sistema con relativo controllore ed un'unica grandezza di controllo in retroazione, come in figura:

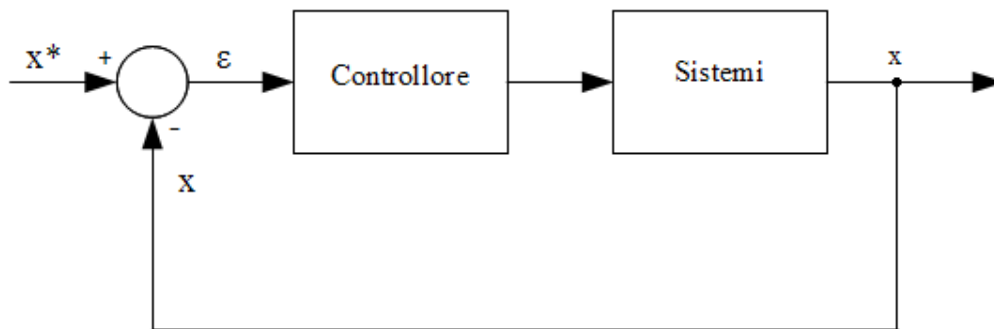


Fig. 4.1 schema generale del controllo

Confrontando con il segnale di riferimento la misura della grandezza che interessa il sistema, otteniamo l'errore, che rappresenta l'entità del distacco tra misura retro azionata e riferimento. Supponiamo che l'errore, indicato con ε (sottointendiamo la dipendenza dal tempo per tutti i termini)

$$\varepsilon = x^* - x$$

Sia minore di zero, quindi il riferimento x^* è minore della grandezza misurata, presente nel sistema, indicata con x . Il nostro obiettivo è ridurre l'errore, farlo tendere a zero, l'errore deve allora crescere nel tempo, o meglio, avere derivata positiva.

Analogamente, se avessi $\varepsilon > 0$ per annullarsi deve calare e quindi avere derivata negativa.

Possiamo quindi ricavare la condizione generale sufficiente e necessaria affinché un sistema sia stabile ed il relativo errore tenda ad annullarsi:

$$\varepsilon \frac{d\varepsilon}{dt} = \varepsilon \varepsilon' < 0$$

Prodotto tra errore e relativa derivata deve essere minore di zero, ovvero i due devono essere di segno opposto. In questo modo il controllore condizionerà il sistema in modo da ridurre la differenza tra grandezza controllata e il suo valore di riferimento.

Osserviamo ora la necessità di avere un controllo ad isteresi: un controllore quindi che opera valutando se l'errore che riceve in ingresso rientra o meno in una banda predefinita, intervenendo solo se il valore è esterno a quest'ultima.

A causa delle inerzie del sistema infatti, non è possibile ottenere un controllo esatto dell'errore, o meglio non si riesce a stabilizzare l'errore ad un valore pari a zero, se non imponendo dinamiche molto lente. Non è quindi consigliabile rispettare questa specifica, in quanto richiederebbe un continuo intervento del controllore sottoponendo i componenti a maggiori stress, determinandone un precoce invecchiamento.

In pratica utilizzando un controllore come quello descritto, non si interviene appena la grandezza controllata supera il riferimento in un verso o nell'altro, ma quando il valore desiderato viene superato di una certa quantità che definisce la banda di tolleranza dell'errore.

Naturalmente, più ampia è la banda di isteresi, quindi l'intervallo di valori in cui l'errore si considera accettabile, minori saranno gli interventi del sistema di controllo e viceversa.

Visto il principio generale in linea teorica, precisiamo che nel nostro controllo non abbiamo una sola grandezza ma bensì due, possiamo rielaborare il tutto definendo un'unica variabile di tipo complesso che permette di tener conto di entrambi i fattori, compattando la trattazione (sottintendiamo sempre la dipendenza dal tempo):

$$\bar{z} = x + jy$$

E considera le generiche grandezze misurate x e y .

Invece se analizziamo le grandezze di riferimento generiche x^* e y^* :

$$\bar{z}^* = \bar{x}^* + j\bar{y}^*$$

In analogia a quanto visto nel caso ad una sola variabile definiamo l'errore come differenza tra riferimento e valore misurato:

$$\bar{\varepsilon} = \bar{z}^* - \bar{z} = (\bar{x}^* + j\bar{y}^*) - (x + jy) = (x^* - x) + j(y^* - y) = \varepsilon_{Re} + j\varepsilon_{Im}$$

Abbiamo quindi associato ad ε_{Re} l'errore della grandezza controllata x , analogamente con ε_{Im} associamo l'errore della grandezza y .

Si può dimostrare che la condizione necessaria e sufficiente affinché l'errore tenda ad annullarsi è la seguente:

$$Re(\bar{\varepsilon}'/\bar{\varepsilon}) < 0$$

Indicando con $\bar{\varepsilon}'$ la derivata temporale della variabile complessa che rappresenta l'errore mentre con $\check{\varepsilon}$ il suo complesso coniugato.

Esplicitando i due termini:

$$\check{\varepsilon} = \varepsilon_{Re} - j\varepsilon_{Im}$$

$$\bar{\varepsilon}' = \frac{d}{dt}\bar{\varepsilon} = \frac{d}{dt}(\varepsilon_{Re} + j\varepsilon_{Im}) = \frac{d}{dt}\varepsilon_{Re} + \frac{d}{dt}\varepsilon_{Im} = \varepsilon'_{Re} + j\varepsilon'_{Im}$$

Con

$$\varepsilon'_{Re} = \frac{d}{dt}\varepsilon_{Re} = \frac{d}{dt}(x^* - x) = \frac{d}{dt}x^* - \frac{d}{dt}x = x^{*'} - x'$$

$$\varepsilon'_{Im} = \frac{d}{dt}\varepsilon_{Im} = \frac{d}{dt}(y^* - y) = \frac{d}{dt}y^* - \frac{d}{dt}y = y^{*'} - y'$$

Sostituendo:

$$\begin{aligned} Re(\bar{\varepsilon}'\check{\varepsilon}) &= Re[(\varepsilon'_{Re} + j\varepsilon'_{Im})(\varepsilon_{Re} - j\varepsilon_{Im})] = Re[\varepsilon'_{Re}\varepsilon_{Re} + \varepsilon'_{Im}\varepsilon_{Im} + j(\varepsilon'_{Im}\varepsilon_{Re} - \varepsilon'_{Re}\varepsilon_{Im})] \\ &= \varepsilon'_{Re}\varepsilon_{Re} + \varepsilon'_{Im}\varepsilon_{Im} < 0 \end{aligned}$$

La condizione sopra scritta e' sicuramente verificata se vale:

$$\begin{cases} \varepsilon'_{Re}\varepsilon_{Re} < 0 \\ \varepsilon'_{Im}\varepsilon_{Im} < 0 \end{cases}$$

Condizione sufficiente ma non necessaria in quanto altre combinazioni nel sistema possono verificare la formula di cui sopra.

Interpretiamo ora il sistema di equazioni, precisando che dobbiamo controllare due grandezze in maniera separata, come nel caso monodimensionale. Se utilizziamo due controllori ad isteresi, uno per la variabile x ed uno per la variabile y , abbiamo rispettivamente una banda posta sull'asse reale ed una sull'asse immaginario:

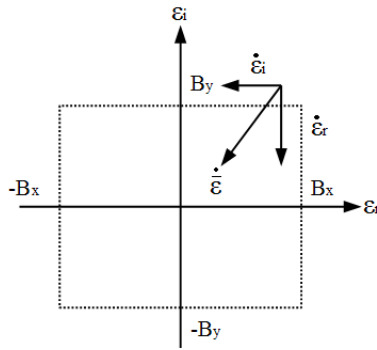


Fig. 4.2 Interpretazione banda isteresi in due variabili

Se quindi l'errore delle due variabili ha coordinate equivalenti nel piano complesso ϵ_{Re} , ϵ_{Im} , il controllore forza il sistema imponendo i giusti valori delle relative derivate, rappresentabili con un vettore di verso tale da portare l'errore verso il valore nullo. Nell'esempio se l'errore ha componenti entrambe maggiori di zero, il vettore delle derivate ha componenti entrambe minori di zero.

4.4 DTC per SPM principio di funzionamento:

Come scritto nei precedenti paragrafi, l'idea di base di questo tipo di controllo è riuscire a comandare le grandezze rappresentative di ogni azionamento: coppia del motore e flusso di statore (in analogia ad esempio al controllo di armatura e di eccitazione del motore a corrente continua o del FOC per il motore asincrono).

Per fare ciò, come già visto nei richiami teorici, andiamo a modificare la velocità di rotazione e il modulo del vettore flusso di statore intervenendo sulla tensione di alimentazione del motore in maniera opportuna essendo quest'ultima capace di controllare sia il flusso di statore che la coppia, come si evince dalle formule.

Il dispositivo che permette di modificare la tensione di alimentazione della macchina è l'inverter, nel quale, in questo particolare modo di comando, andiamo a comandare direttamente gli stati dei 6 interruttore statici (3-f) costituenti il dispositivo.

Precisiamo che per lo studio del sistema utilizzeremo lo strumento matematico definito dalle trasformazioni di Clarke e Park: i vettori spaziali; riusciamo così a rappresentare le terne trifasi in gioco con un vettore che nel caso specifico della tensione in uscita dell'inverter assume sei diverse posizioni ai vertici di un esagono.

Nasce allora spontanea la domanda seguente: con quali criteri vengono determinati i vettori della tensione da imporre sulle tre fasi in modo da raggiungere il valore del flusso e della coppia di riferimento, o, del tutto equivalente: come facciamo ad imporre la giusta successione di vettori spaziali per ottenere questi valori desiderati.

Per rispondere a questa domanda applichiamo la teoria vista, che si basa sulle classiche regole dell'algebra dei sistemi. Dal confronto tra riferimento e valore misurato il controllore ad isteresi decide o meno di intervenire, a seconda che l'errore sia contenuto o meno in una banda di tolleranza. Quando interviene, perché una delle due grandezze (coppia o flusso) o entrambe, sono eccessivamente differenti dal riferimento, vengono valutate le possibili derivate della coppia e del flusso per ognuno dei vettori attivi applicabili dall'inverter alla macchina. Così facendo si ha un'informazione dell'andamento della grandezza se si applicasse in questo istante il corrispondente vettore attivo, e

quindi se la misura è maggiore del riferimento l'andamento dovrà essere decrescente per raggiungere il valore voluto, ovviamente si può fare il discorso duale.

Il passo successivo consiste nel determinare quale vettore di tensione adottare tra gli otto disponibili (due nulli più sei attivi); il criterio di scelta sta nell'applicare quello i cui valori appena calcolati, sottratti alla derivata del riferimento, rispondono alla condizione sufficiente ma non necessaria che il prodotto tra errore della grandezza in esame e rispettiva derivata sia minore di zero.

Applicando allora questa specifica tensione l'errore va a ridursi, se quest'ultimo è ancora elevato il ciclo si ripete e viene valutato un altro vettore, che può anche essere lo stesso del passo precedente. La procedura continua fino a che l'errore non rientra nella banda di tolleranza.

Per agevolare lo studio del funzionamento di questo tipo di controllo possiamo rappresentare la situazione con l'aiuto dei vettori spaziali e del piano complesso: i riferimenti di coppia e flusso sono rappresentati da un vettore con corrispondente parte reale ed immaginaria (vettore "riferimento"), i valori misurati di coppia e flusso hanno anche loro la rispettiva rappresentazione sul piano complesso (vettore "misurato"), le bande di isteresi di flusso e coppia trovano il loro equivalente in una circonferenza limite, che determina o meno l'intervento a seconda che l'errore, la differenza tra riferimento e valore misurato, stia o no entro tale limite.

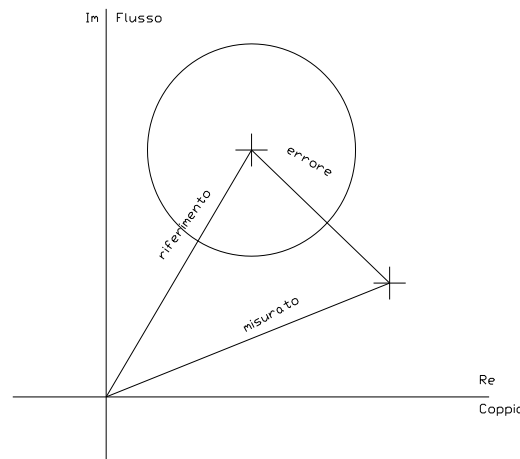


Fig. 4.3 rappresentazione grandezze su piano complesso

Valutiamo adesso per ciascun vettore in uscita dell'inverter la derivata della coppia e la derivata del flusso, queste due trovano la corrispondente rappresentazione sul piano complesso come parte reale ed immaginaria di un vettore "derivata", precisiamo che le sei tensioni applicabili sono relative ad un unico valore di coppia e flusso misurato, poi la variazione sarà diversa a seconda delle grandezze imposte dall'inverter.

Supponendo che il riferimento sia costante, non cambia quindi nel tempo, ho la situazione rappresentata in figura:

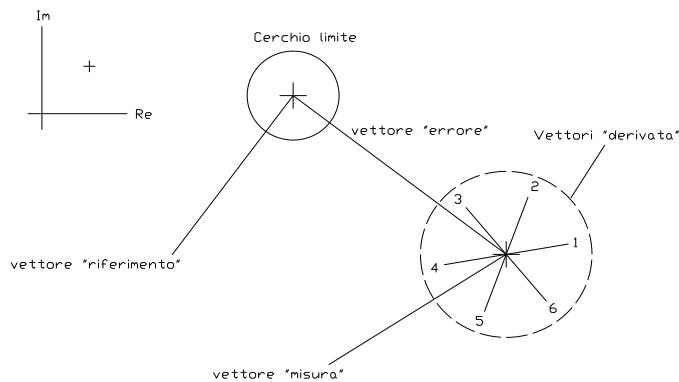


Fig. 4.4 rappresentazione derivate su piano complesso

Osserviamo che, tra i sei vettori “derivata”, applicandone uno con parte reale ed immaginaria di segno contrario a quelle del vettore “errore”(il che equivale a rispettare la condizione sufficiente ma non necessaria) sicuramente riduciamo l’errore, cioè riportiamo il vettore “misura” più vicino al vettore “riferimento”.

Notiamo subito che potrei avere più di un vettore che soddisfa le richieste (in figura è solo il vettore 3), oppure che l’errore potrebbe ridursi anche con un vettore con parte reale e/o immaginaria dello stesso segno dell’errore (in figura è il 4). Ci chiediamo quindi che criterio adottare per procedere: in un primo momento vogliamo solo ridurre l’errore nel più breve tempo possibile per portarci nella maniera più sicura dentro il cerchio, scegliendo allora il vettore “derivata” che sia, in termini semplicistici, il più opposto possibile al vettore “errore” (è il 3 nel nostro caso), cioè che rispetta la condizione già ripetuta.

Successivamente, arrivati alla stabilità, per affinare il controllo passiamo al livello successivo: tra i vettori che soddisfano le richieste potremo discriminare quello per esempio che presenta meno commutazioni oppure che diminuisce i tempi di intervento.

4.5 Adattamento criteri di stabilità per SPM

Ora, spiegato in termini descrittivi il principio di funzionamento, possiamo particularizzare le equazioni che garantiscono la stabilità del controllo per il nostro motore SPM. Individuiamo in questo modo le condizioni che devono rispettare le grandezze vettoriali in gioco per far sì che l’errore tenda ad annullarsi

Dovendo controllare flusso e coppia del motore definiamo la seguente variabile complessa:

$$\bar{z} = m + j|\lambda_s^r|$$

dove la coppia è la parte reale e il flusso la parte immaginaria.

In maniera analoga definiamo il riferimento con un'altra variabile complessa:

$$\bar{z}^* = m^* + j|\lambda_s^r|^*$$

Osservazioni:

Facendo una semplice analisi dimensionale notiamo che gli ordini di grandezza delle due componenti sono generalmente diverse. Inoltre, matematicamente parlando sto operando con grandezze le cui unità di misura sono differenti: Nm e Vs.

Per ovviare a questi problemi uso una diversa notazione, normalizzando le grandezze con i rispettivi valori nominali:

$$\bar{z} = \frac{m}{M_N} + j \frac{|\lambda_s|}{\Lambda_N}$$

Inoltre da risultati ed osservazioni sperimentali la velocità di variazione della componente di flusso dovuta a ciascun vettore spaziale ottenibile dall'inverter è molto più contenuta rispetto alla variazione della componente di coppia; si adotta quindi moltiplicare la parte immaginaria per un coefficiente $a > 0$ in modo che il campo di variazione venga "amplificato".

Ottingo quindi:

$$\bar{z} = \frac{m}{M_N} + ja \frac{|\lambda_s|}{\Lambda_N}$$

$$\bar{z}^* = \frac{m^*}{M_N} + ja \frac{|\lambda_s|^*}{\Lambda_N}$$

Possiamo calcolare adesso l'errore della variabile complessa come:

$$\bar{\varepsilon} = \bar{z}^* - \bar{z} = \left(\frac{m^*}{M_N} - \frac{m}{M_N} + ja \left(\frac{|\lambda_s|^*}{\Lambda_N} - \frac{|\lambda_s|}{\Lambda_N} \right) \right)$$

Distinguiamo parte reale, rappresentativa dell'errore di coppia:

$$\varepsilon_{Re} = Re\{\bar{\varepsilon}\} = \frac{m^*}{M_N} - \frac{m}{M_N} = \frac{m^* - m}{M_N}$$

E parte immaginaria, rappresentativa dell'errore di flusso:

$$\varepsilon_{Im} = Im\{\bar{\varepsilon}\} = a \frac{|\lambda_s|^*}{\Lambda_N} - a \frac{|\lambda_s|}{\Lambda_N} = a \frac{|\lambda_s|^* - |\lambda_s|}{\Lambda_N}$$

In questo modo riusciamo a compattare le informazioni che riguardano gli errori che si commettono per il controllo sia di flusso che di coppia, in un vettore che può essere studiato sul piano complesso, semplificando la trattazione matematica del problema.

Procedendo quindi riprendiamo le condizioni necessarie e sufficienti a garantire che l'evoluzione delle grandezze di controllo nella dinamica del sistema rimangano all'interno dell'area di tolleranza.

Precisando che con l'apice ' indiciamo la derivata prima, deve essere:

$$\begin{aligned}
Re\{\bar{\varepsilon}'\tilde{\varepsilon}\} &= Re\{(\varepsilon'_{Re} + j\varepsilon'_{Im}) + (\varepsilon_{Re} - j\varepsilon_{Im})\} = Re\{\varepsilon'_{Re}\varepsilon_{Re} - j\varepsilon'_{Re}\varepsilon_{Im} + j\varepsilon'_{Im}\varepsilon_{Re} + \varepsilon'_{Im}\varepsilon_{Im}\} \\
&= \varepsilon'_{Re}\varepsilon_{Re} + \varepsilon'_{Im}\varepsilon_{Im} \\
&= \frac{1}{M_N} \left((m^* - m) \frac{d}{dt} (m^* - m) \right) + \frac{a}{\Lambda_N} \left((|\lambda_s|^* - |\lambda_s|) \frac{d}{dt} (|\lambda_s|^* - |\lambda_s|) \right) < 0
\end{aligned}$$

Da questa possiamo ricavare la condizione sufficiente ma non necessaria, in quanto la formula sopra è sicuramente verificata se vale:

$$\begin{cases} \varepsilon'_{Re}\varepsilon_{Re} = \frac{1}{M_N} \left((m^* - m) \frac{d}{dt} (m^* - m) \right) < 0 \\ \varepsilon'_{Im}\varepsilon_{Im} = \frac{a}{\Lambda_N} \left((|\lambda_s|^* - |\lambda_s|) \frac{d}{dt} (|\lambda_s|^* - |\lambda_s|) \right) < 0 \end{cases}$$

Ma questa non è l'unica condizione per cui la formula sopra è verificata.

Per comprendere meglio il significato delle equazioni che garantiscono la convergenza a zero dell'errore nel caso di un controllo DTC per un motore SPM andiamo a considerarle in maniera separata.

Consideriamo (costanti a parte) la coppia, e supponiamo che in un primo approccio il riferimento non vari sensibilmente, di conseguenza avrà derivata nulla o comunque trascurabile.

Supponiamo poi che la coppia prodotta sia troppo alta, di conseguenza, essendo maggiore del riferimento, l'errore ha un valore negativo; per avere la prima condizione rispettata dovremo avere il termine derivato di segno positivo, per cui la derivata temporale della coppia avendo il meno davanti dovrà essere negativa anch'essa.

Se

$$m > m^* \Rightarrow \varepsilon_{Re} < 0$$

Per avere

$$\varepsilon_{Re}\varepsilon'_{Re} < 0 \Rightarrow \varepsilon'_{Re} > 0 \Rightarrow -\frac{dm}{dt} > 0 \Rightarrow \frac{dm}{dt} < 0$$

Tutto ciò si traduce nel fatto che il motore, producendo più coppia di quella desiderata dovrà per forza diminuirla nel tempo.

Analogo ragionamento se l'errore è positivo, cioè la macchina produce una coppia troppo bassa la derivata sarà positiva perché dovrà aumentarla.

Se

$$m < m^* \Rightarrow \varepsilon_{Re} > 0$$

Per avere

$$\varepsilon_{Re} \varepsilon'_{Re} < 0 \Rightarrow \varepsilon'_{Re} < 0 \Rightarrow -\frac{dm}{dt} < 0 \Rightarrow \frac{dm}{dt} > 0$$

Stessi identici passaggi possono essere estesi alla componente del flusso:

Se

$$|\lambda_s| > |\lambda_s|^* \Rightarrow \varepsilon_{Im} < 0$$

Per ottenere

$$\varepsilon_{Im} \varepsilon'_{Im} < 0 \Rightarrow \varepsilon'_{Im} > 0 \Rightarrow -\frac{d|\lambda_s|}{dt} > 0 \Rightarrow \frac{d|\lambda_s|}{dt} < 0$$

Se invece:

$$|\lambda_s| < |\lambda_s|^* \Rightarrow \varepsilon_{Im} > 0$$

Per ottenere

$$\varepsilon_{Im} \varepsilon'_{Im} < 0 \Rightarrow \varepsilon'_{Im} < 0 \Rightarrow -\frac{d|\lambda_s|}{dt} < 0 \Rightarrow \frac{d|\lambda_s|}{dt} > 0$$

E queste rappresentano le condizioni sufficienti ma non necessarie da rispettare quando i riferimenti sono costanti nel tempo o variano molto lentamente

Analizziamo ora il caso in cui le grandezze di comando varino nel tempo, precisando però che nella realtà il caso in cui il riferimento di flusso sia variabile nel tempo si presenta più raramente di quello della coppia, a meno di non voler deflussare la macchina.

Ipotizziamo, sempre con le costanti a parte, che l'errore, riferito alla parte reale della variabile complessa sia maggiore di zero, la macchina quindi crea una coppia non sufficiente ed inferiore al riferimento, quest'ultimo però cambia e potrebbe calare o crescere nel tempo diventando anche inferiore alla coppia creata dal motore. Andiamo quindi a definire meglio i criteri da rispettare in questa particolare situazione.

Se

$$m^* > m \Rightarrow \varepsilon_{Re} > 0$$

Per rispettare:

$$\varepsilon_{Re} \varepsilon'_{Re} < 0 \Rightarrow \varepsilon'_{Re} < 0 \Rightarrow \frac{d}{dt}(m^* - m) < 0 \Rightarrow \frac{d}{dt}m^* - \frac{d}{dt}m < 0 \Rightarrow \frac{d}{dt}m > \frac{d}{dt}m^*$$

Per riportare l'errore all'interno della banda di tolleranza la derivata della coppia del motore deve essere maggiore di quella del riferimento, in altre parole per ridurre l'errore al motore deve essere imposta una dinamica maggiore di quella del riferimento, nel tempo deve variare molto più veloce. Analoghi ragionamenti si possono applicare negli altri casi di coppia e flusso, interpretando ovviamente nella maniera corretta le informazioni date dalle equazioni. Abbiamo quindi ricavato la condizione che devono avere le derivate degli errori: di essere di segno opposto agli errori stessi

$$\begin{cases} \text{sign}(\varepsilon'_{Re}) = -\text{sign}(\varepsilon_{Re}) \\ \text{sign}(\varepsilon'_{Im}) = -\text{sign}(\varepsilon_{Im}) \end{cases}$$

Sostituendo:

$$\begin{cases} \text{sign} \left[\frac{d}{dt} (m^* - m) \right] = -\text{sign}(m^* - m) \\ \text{sign} \left[\frac{d}{dt} (|\lambda_s|^* - |\lambda_s|) \right] = -\text{sign}(|\lambda_s|^* - |\lambda_s|) \end{cases}$$

Ricavo per la coppia che:

$$\begin{cases} \text{se } m^* > m \Rightarrow \frac{d}{dt} (m^* - m) < 0 \Rightarrow \frac{d}{dt} m > \frac{d}{dt} m^* \\ \text{se } m^* < m \Rightarrow \frac{d}{dt} (m^* - m) > 0 \Rightarrow \frac{d}{dt} m < \frac{d}{dt} m^* \end{cases}$$

Analogamente per il flusso

$$\begin{cases} \text{se } |\lambda_s|^* > |\lambda_s| \Rightarrow \frac{d}{dt} |\lambda_s| > \frac{d}{dt} |\lambda_s|^* \\ \text{se } |\lambda_s|^* < |\lambda_s| \Rightarrow \frac{d}{dt} |\lambda_s| < \frac{d}{dt} |\lambda_s|^* \end{cases}$$

Da queste possiamo finalmente capire come effettuare il controllo diretto di coppia e di flusso. Il nostro obiettivo è comandare queste due grandezze, abbiamo quindi le informazioni necessarie sia per quanto riguarda il riferimento, visto che è impostato da noi, sia per quanto riguarda le grandezze che interessano la macchina, visto che possiamo usare degli stimatori per valutarle con una buona precisione. Da queste osservazioni si capisce che il controllo di certo non può essere previsto per il riferimento e la rispettiva derivata, andremo quindi a comandare l'alimentazione del motore in modo che le grandezze del motore e le rispettive derivate rispettino le condizioni ricavate

4.6 Calcolo della derivata della coppia

Si tratta ora di calcolare la derivata della coppia a partire dalla conoscenza delle grandezze che interessano la macchina e su cui possiamo intervenire, quali: tensione di alimentazione, correnti e parametri caratteristici come resistenza, induttanza, flusso dei magneti permanenti, che non possono essere modificati in quanto caratteristiche proprie del motore.

Svilupperemo le equazioni nel sistema di riferimento rotante d-q sincrono con il rotore, omettendo l'apice r per semplificare la trattazione.

A partire dall'espressione generale della coppia, valida per ogni motore;

$$m = \frac{3}{2}p(\lambda_d i_q - \lambda_q i_d)$$

Possiamo calcolare la derivata temporale:

$$\frac{d}{dt}m = \frac{3}{2}p \left(\frac{d\lambda_d}{dt} i_q + \frac{di_q}{dt} \lambda_d - \frac{d\lambda_q}{dt} i_d - \frac{di_d}{dt} \lambda_q \right)$$

Dobbiamo adesso ricavare le relazioni che legano flussi, correnti e rispettive derivate con le grandezze di alimentazione, quali tensione, flussi e correnti stesse.

Dal sistema di equazioni che descrive il comportamento dinamico in generale di un motore:

$$\begin{cases} u_d = R i_d + \frac{d}{dt} \lambda_d - \omega_{me} \lambda_q \\ u_q = R i_q + \frac{d}{dt} \lambda_q + \omega_{me} \lambda_d \end{cases}$$

Possiamo ottenere:

$$\begin{cases} \frac{d\lambda_d}{dt} = u_d - R i_d + \omega_{me} \lambda_q \\ \frac{d\lambda_q}{dt} = u_q - R i_q - \omega_{me} \lambda_d \end{cases}$$

E dal sistema di equazioni che esprime il legame tra flussi e correnti per un motore SPM:

$$\begin{cases} \lambda_q = L i_q \\ \lambda_d = L i_d + \Lambda_{mg} \end{cases}$$

Derivando:

$$\begin{cases} \frac{d\lambda_d}{dt} = L \frac{di_d}{dt} \\ \frac{d\lambda_q}{dt} = L \frac{di_q}{dt} \end{cases}$$

Quindi:

$$\begin{cases} \frac{di_d}{dt} = \frac{1}{L} \frac{d\lambda_d}{dt} = \frac{1}{L} (u_d - Ri_d + \omega_{me}\lambda_q) \\ \frac{di_q}{dt} = \frac{1}{L} \frac{d\lambda_q}{dt} = \frac{1}{L} (u_q - Ri_q - \omega_{me}\lambda_d) \end{cases}$$

Possiamo allora sostituire nell'espressione generale della derivata della coppia:

$$\begin{aligned} \frac{d}{dt} m &= \frac{3}{2} p \left(i_q (u_d - Ri_d + \omega_{me}\lambda_q) + \lambda_d \frac{1}{L} (u_q - Ri_q - \omega_{me}\lambda_d) \right. \\ &\quad \left. - i_d (u_q - Ri_q - \omega_{me}\lambda_d) - \lambda_q \frac{1}{L} (u_d - Ri_d + \omega_{me}\lambda_q) \right) \\ &= \frac{3}{2} p \left(u_d i_q - Ri_d i_q + \omega_{me}\lambda_q i_q + \frac{1}{L} u_q \lambda_d - \frac{R}{L} i_q \lambda_d - \frac{\omega_{me}}{L} \lambda_d \lambda_d \right. \\ &\quad \left. - u_q i_d + Ri_q i_d + \omega_{me}\lambda_d i_d - \frac{1}{L} u_d \lambda_q + \frac{R}{L} i_d \lambda_q - \frac{\omega_{me}}{L} \lambda_q \lambda_q \right) \end{aligned}$$

Possiamo isolare i termini:

$$A = u_d i_q - u_q i_d$$

$$B = R(i_d i_q - i_d i_q)$$

$$C = \omega_{me}(\lambda_q i_q + \lambda_d i_d) = \omega_{me}((Li_q)i_q + (Li_d + \Lambda_{mg})i_d) = \omega_{me}(Li_q^2 + Li_d^2 + \Lambda_{mg}i_d)$$

$$\begin{aligned} D &= \frac{1}{L} u_q \lambda_d - \frac{1}{L} u_d \lambda_q = \frac{1}{L} (u_q \lambda_d - u_d \lambda_q) = \frac{1}{L} (u_q (Li_d + \Lambda_{mg}) - u_d (Li_q)) \\ &= \frac{1}{L} (u_q Li_d + u_q \Lambda_{mg} - u_d Li_q) = u_q i_d + u_q \frac{\Lambda_{mg}}{L} - u_d i_q \end{aligned}$$

$$E = -\frac{R}{L}(i_q\lambda_d - i_d\lambda_q) = -\frac{R}{L}(i_q(Li_d + \Lambda_{mg}) - i_d(Li_q)) = -\frac{R}{L}(Li_qi_d + i_q\Lambda_{mg} - Li_di_q) \\ = -\frac{R}{L}(\Lambda_{mg}i_q)$$

$$F = -\frac{\omega_{me}}{L}(\lambda_d^2 + \lambda_q^2) = -\frac{\omega_{me}}{L}(\lambda_d(Li_d + \Lambda_{mg}) + \lambda_q(Li_q)) \\ = -\frac{\omega_{me}}{L}((Li_d + \Lambda_{mg})(Li_d + \Lambda_{mg}) + (Li_q)(Li_q)) \\ = -\frac{\omega_{me}}{L}(L^2i_d^2 + 2Li_d\Lambda_{mg} + \Lambda_{mg}^2 + L^2i_q^2) = -\omega_{me}\left(Li_d^2 + 2i_d\Lambda_{mg} + \frac{\Lambda_{mg}^2}{L} + Li_q^2\right)$$

Osservazioni sui vari termini:

$$B = 0$$

$$C + F = \omega_{me}\left(Li_q^2 + Li_d^2 + \Lambda_{mg}i_d - Li_d^2 - 2i_d\Lambda_{mg} - \frac{\Lambda_{mg}^2}{L} - Li_q^2\right) \\ = \omega_{me}\left(\Lambda_{mg}i_d - 2i_d\Lambda_{mg} - \frac{\Lambda_{mg}^2}{L}\right) = -\omega_{me}\Lambda_{mg}\left(i_d + \frac{\Lambda_{mg}}{L}\right)$$

$$A + D = i_q - u_qi_d + u_qi_d + u_q\frac{\Lambda_{mg}}{L} - u_di_q = u_q\frac{\Lambda_{mg}}{L}$$

Quindi la derivata di coppia unendo i vari termini diventa:

$$\frac{d}{dt}m = \frac{3}{2}p(B + C + F + A + D + E) = \frac{3}{2}p\left(0 - \omega_{me}\Lambda_{mg}\left(i_d + \frac{\Lambda_{mg}}{L}\right) + u_q\frac{\Lambda_{mg}}{L} - \frac{R}{L}(\Lambda_{mg}i_q)\right) \\ = \frac{3}{2}p\left(-\frac{R}{L}(\Lambda_{mg}i_q) + u_q\frac{\Lambda_{mg}}{L} - \omega_{me}\Lambda_{mg}\left(i_d + \frac{\Lambda_{mg}}{L}\right)\right)$$

Questa equazione mette in relazione la derivata della coppia con le grandezze che interessano la macchina, in particolare osserviamo che contiene i parametri caratteristici del motore: induttanza di macchina L , resistenza di fase R , il flusso dei magneti permanenti Λ_{mg} e la velocità ω_{me} e le correnti e tensioni che interessano il nostro BL SPM.

La grandezza che andiamo a modificare per intervenire sul controllo è la tensione, e sarà quella che riduce l'errore.

Un'osservazione finale va' fatta a riguardo del carattere generale della trattazione, infatti l'espressione della derivata della coppia avremmo potuto calcolarla esplicitando direttamente l'equazione iniziale, senza ripercorrere i passaggi usuali.

4.7 Calcolo della derivata del modulo del flusso di statore:

Analogamente a quanto visto nel calcolo della derivata della coppia, andiamo a cercare una formula che metta in relazione la derivata del flusso con grandezze caratteristiche della macchina, su cui possiamo anche intervenire per il controllo.

Dall'equazione della derivata del flusso nel sistema di riferimento rotante:

$$\begin{cases} \frac{d\lambda_d}{dt} = u_d - Ri_d + \omega_{me}\lambda_q \\ \frac{d\lambda_q}{dt} = u_q - Ri_q - \omega_{me}\lambda_d \end{cases}$$

Moltiplicando da ambo le parti per la corrispondente componente di flusso:

$$\begin{cases} \frac{d\lambda_d}{dt} \lambda_d = u_d \lambda_d - Ri_d \lambda_d + \omega_{me} \lambda_q \lambda_d \\ \frac{d\lambda_q}{dt} \lambda_q = u_q \lambda_q - Ri_q \lambda_q - \omega_{me} \lambda_d \lambda_q \end{cases}$$

Sommando le due:

$$\begin{aligned} \frac{d\lambda_d}{dt} \lambda_d + \frac{d\lambda_q}{dt} \lambda_q &= u_d \lambda_d - Ri_d \lambda_d + \omega_{me} \lambda_q \lambda_d + u_q \lambda_q - Ri_q \lambda_q - \omega_{me} \lambda_d \lambda_q \\ &= u_d \lambda_d - Ri_d \lambda_d + u_q \lambda_q - Ri_q \lambda_q \end{aligned}$$

Ma osserviamo che:

$$\frac{d\lambda_d}{dt} \lambda_d + \frac{d\lambda_q}{dt} \lambda_q = \frac{1}{2} \frac{d}{dt} \lambda_d^2 + \frac{1}{2} \frac{d}{dt} \lambda_q^2 = \frac{1}{2} \frac{d}{dt} (\lambda_d^2 + \lambda_q^2) = \frac{1}{2} \frac{d}{dt} |\lambda^r|^2 = |\lambda^r| \frac{d}{dt} |\lambda^r|$$

Quindi:

$$\frac{d}{dt} |\lambda| = \frac{1}{|\lambda|} \left((u_d \lambda_d + u_q \lambda_q) - R(i_d \lambda_d + i_q \lambda_q) \right)$$

Abbiamo anche in questo caso calcolato la derivata del flusso di statore in funzione di grandezze che interessano la macchina: i parametri, le correnti che percorrono le fasi e le tensioni su cui possiamo intervenire per il comando.

Osservazioni:

andiamo ad esporre alcune interessanti considerazioni; il termine:

$$u_d \lambda_d + u_q \lambda_q = |\lambda| u_d^\lambda$$

Ove u_d^λ e' la componente del vettore tensione nella direzione del flusso, avendo definito un sistema di riferimento solidale con quest'ultimo.

Analogamente si puo' fare per la corrente e quindi;

$$\frac{d}{dt}|\lambda| = u_d^\lambda - Ri_d^\lambda$$

Ove u_d^λ e', in genere, poco diverso da u_d^r

5-SIMULAZIONI

5.1-Introduzione

In questo capitolo andremo ad applicare in un ambiente di simulazione la teoria vista nei precedenti paragrafi, usando il tool Simulink di Matlab.

Partendo dalle formule ricavate otterremo degli schemi a blocchi nel riferimento rotante d-q rappresentativi non solo del sistema motore-carico ma di tutta la regolazione.

Siamo partiti dal controllo DTC più semplice possibile, il più “puro”, che ha presentato subito varie problematiche poi risolte in step successivi. Andiamo quindi ad analizzare i vari passaggi.

5.2-Presentazione del motore utilizzato

La macchina da noi analizzata è un motore BL SPM pensato per la trazione ferroviaria.

Il nostro obiettivo è comandare la coppia e il flusso desiderati ma prima abbiamo bisogno di conoscere il motore. Da misure effettuate in precedenza ricaviamo i parametri della macchina che ci interessano, li scriviamo poi in un listato il quale una volta compilato fa assumere alle variabili presenti nel modello simulink i valori scritti.

```
%Valori di riferimento del controllo
```

```
m_rif=5000
```

```
fl_rif=4.8
```

```
%valori nominali della macchina
```

```
Mn=5000
```

```
Fn=4.8
```

```
%parametri elettrici motore SPM
```

```
R_m= 0.0325
```

```
L_m= 0.0046
```

```
Lmg_m= 1.93
```

```
p=2
```

```
%parametri meccanici motore SPM
```

```
B=0.15;
```

```
J=2.9
```

```
n=1200;
```

5.3-Controllo DTC “puro”

Come prima cosa abbiamo costruito il modello del motore SPM in un riferimento sincrono d-q seguendo quanto indicato nel capitolo 1 (Fig. 5.2). Le grandezze che andiamo ad attuare sono le due tensioni u_d e u_q rappresentative proprio della terna di tensioni trifase presenti ai morsetti della macchina. Pensiamo quindi di trasdurre le correnti, in modo da avere comodamente un'informazione sul flusso, implementando un semplice stimatore, e sulla coppia (che in ambito di simulazione computerizzata viene comodamente rilevata nel modello del motore e non stimata).

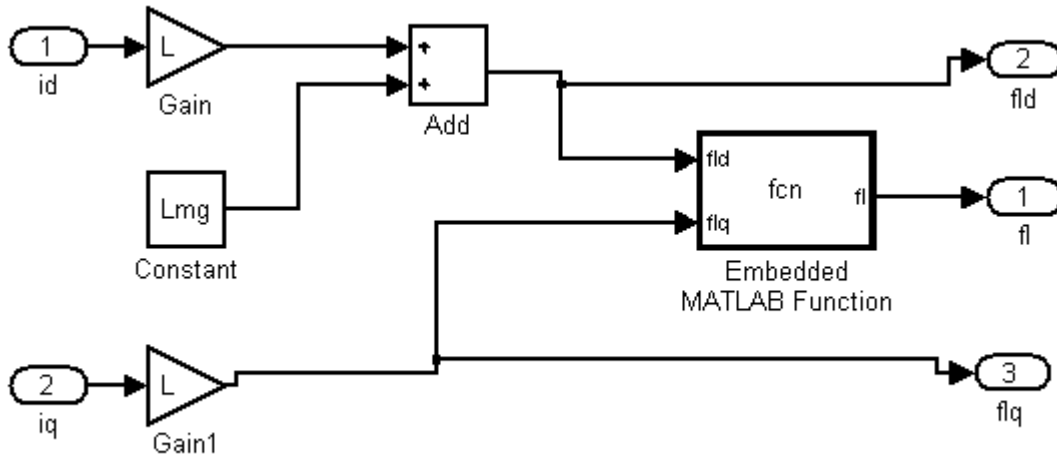


Fig 5.1 Stimatore delle componenti di flusso dalla misura delle correnti calc_flusso

Funzione nella Matlab function:

```
function fl = fcn(fld,flq)
fl=sqrt((fld*fld)+(flq*flq));
```

Il motore viene invece simulato con il seguente schema a blocchi:

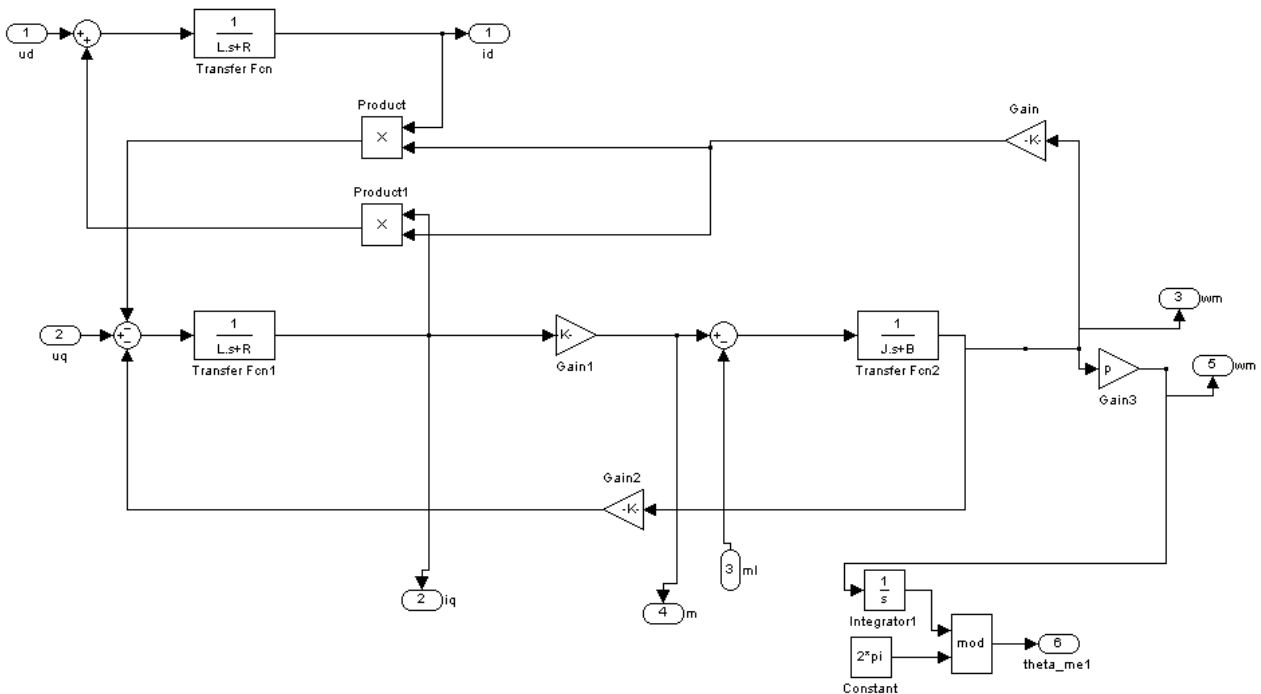


Fig. 5.2 schema a blocchi del motore SPM in un riferimento d-q SPM_dq

Il passo successivo consiste nel creare l'errore di coppia e l'errore di flusso: abbiamo fissato i riferimenti m_{rif} e fl_{rif} , confrontandoli entrambi con i valori di flusso stimato e di coppia trasdotta otteniamo rispettivamente errore di coppia E_m ed errore di flusso E_f

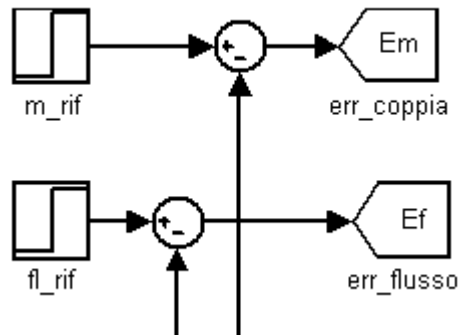


Fig. 5.3 errori

Osserviamo ora le variabili note: correnti trasdotte, flussi stimati, e quelle proprie della macchina: resistenza delle fasi R , induttanza sincrona L , flusso dei magneti permanenti Lmg , coppie polari p . Se vogliamo implementare il calcolo della derivata della coppia e del flusso mancano le sette tensioni attuabili dall'inverter: sei vettori attivi ed un vettore nullo. Con un blocco ulteriore andiamo allora a simulare le tensioni d-q in uscita dal dispositivo riuscendo così ad associare ad ogni vettore spaziale di tensione il corrispondente vettore spaziale di derivata.

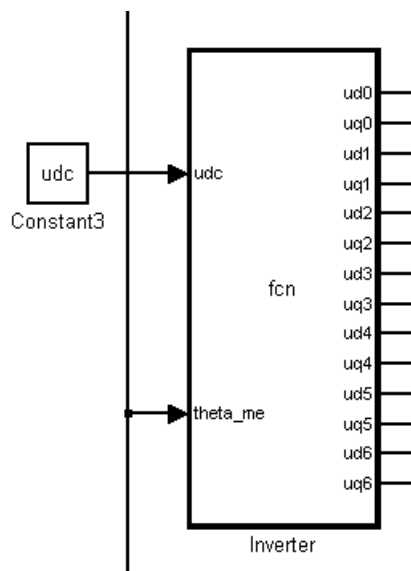


Fig. 5.4 blocco inverter

Il blocco in matlab contiene:

```
function [ud0,uq0,ud1,uq1,ud2,uq2,ud3,uq3,ud4,uq4,ud5,uq5,ud6,uq6]=
fcn(udc,theta_me)

ud0=0;
uq0=0;
u_alfa=0;
u_beta=0;

u_alfa1=(2/3)*udc;
u_beta1=0;
ud1=u_alfa1*cos(theta_me)+u_beta1*sin(theta_me);
```

```

uq1=-u_alfa1*sin(theta_me)+u_beta1*cos(theta_me);

u_alfa2=(2/3)*udc*cos(pi/3);
u_beta2=(2/3)*udc*sin(pi/3);
ud2=u_alfa2*cos(theta_me)+u_beta2*sin(theta_me);
uq2=-u_alfa2*sin(theta_me)+u_beta2*cos(theta_me);

u_alfa3=(2/3)*udc*cos(2*(pi/3));
u_beta3=(2/3)*udc*sin(2*(pi/3));
ud3=u_alfa3*cos(theta_me)+u_beta3*sin(theta_me);
uq3=-u_alfa3*sin(theta_me)+u_beta3*cos(theta_me);

u_alfa4=(2/3)*udc*cos(3*(pi/3));
u_beta4=(2/3)*udc*sin(3*(pi/3));
ud4=u_alfa4*cos(theta_me)+u_beta4*sin(theta_me);
uq4=-u_alfa4*sin(theta_me)+u_beta4*cos(theta_me);

u_alfa5=(2/3)*udc*cos(4*(pi/3));
u_beta5=(2/3)*udc*sin(4*(pi/3));
ud5=u_alfa5*cos(theta_me)+u_beta5*sin(theta_me);
uq5=-u_alfa5*sin(theta_me)+u_beta5*cos(theta_me);

u_alfa6=(2/3)*udc*cos(5*(pi/3));
u_beta6=(2/3)*udc*sin(5*(pi/3));
ud6=u_alfa6*cos(theta_me)+u_beta6*sin(theta_me);
uq6=-u_alfa6*sin(theta_me)+u_beta6*cos(theta_me);

```

Simulando così le tensioni ottenibili dal dispositivo nel riferimento d-q.

Possiamo ora implementare le formule viste al capitolo quattro ed ottenere i sette valori di derivata della coppia $dm0$ $dm1$ $dm2$ $dm3$ $dm4$ $dm5$ $dm6$ e i corrispondenti di flusso $df0$ $df1$ $df2$ $df3$ $df4$ $df5$ $df6$, ciascuno dei quali associato ad un ben definito v.s. di tensione attuabile dall'inverter.

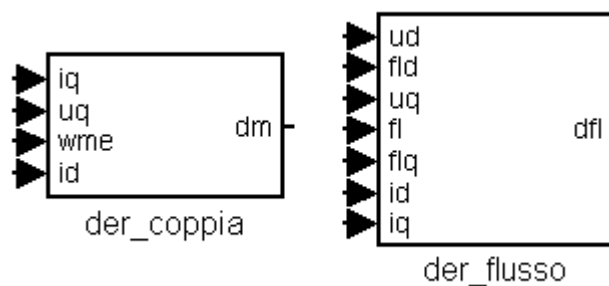


Fig. 5.5 blocchi derivata der_flusso der_coppia

Implementiamo le equazioni con:

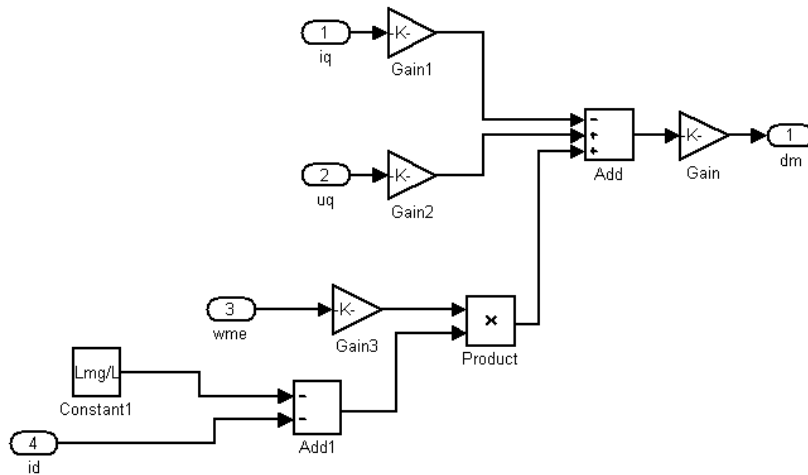


Fig. 5.6 derivata coppia

Per quanto riguarda il flusso invece:

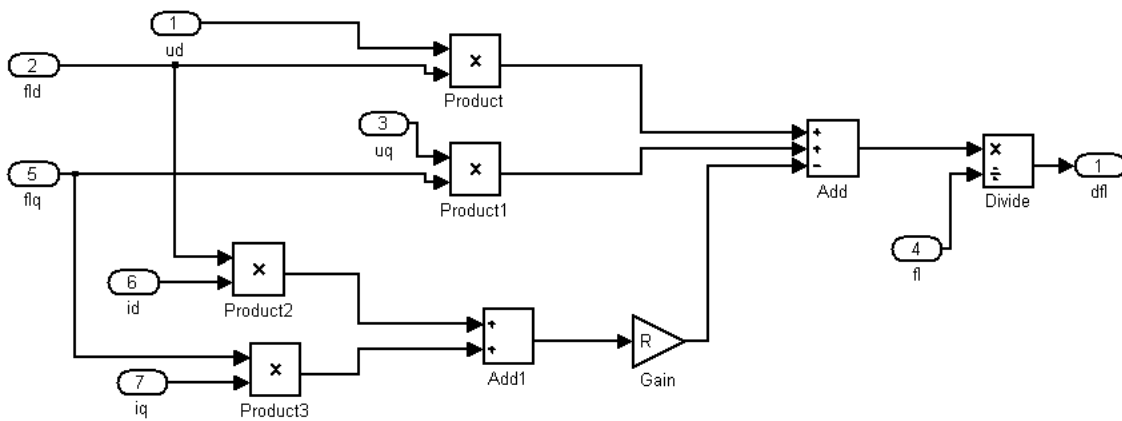


Fig. 5.7 derivata flusso

Arriviamo alla parte centrale del controllo, possiamo definirlo il “cervello” del DTC ossia l’algoritmo che decide i vettori da applicare a seconda del criterio da noi adottato.

Rappresentato con il seguente blocco:

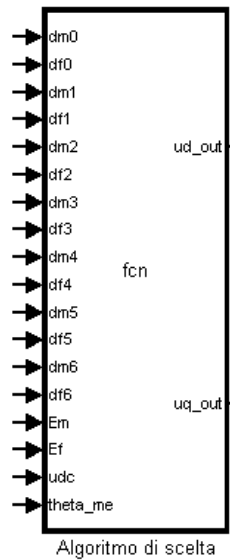


Fig. 5.8 blocco Algoritmo di scelta

Analizziamo quindi la prima procedura utilizzata: il blocco che incorpora le funzioni di matlab riceve in ingresso gli errori E_m E_f le derivate dm_0 dm_1 dm_2 dm_3 dm_4 dm_5 dm_6 e df_0 df_1 df_2 df_3 df_4 df_5 df_6 e l'angolo ϑ_{me} per fare le opportune trasformazioni di Park. Organizziamo in array i valori di derivata, successivamente, avendo i riferimenti costanti, seguiamo quanto detto al paragrafo 4.5 : per avere la convergenza a zero dell'errore derivata ed errore devono avere segno opposto, di conseguenza il loro prodotto deve essere negativo. Moltiplichiamo quindi il corrispondente errore per ogni valore disponibile di derivata; avremo così due array da valutare: uno per la coppia val_m e uno per il flusso val_f che presentano in ogni cella il prodotto tra la derivata e l'errore.

Possiamo osservare la rappresentazione seguente:

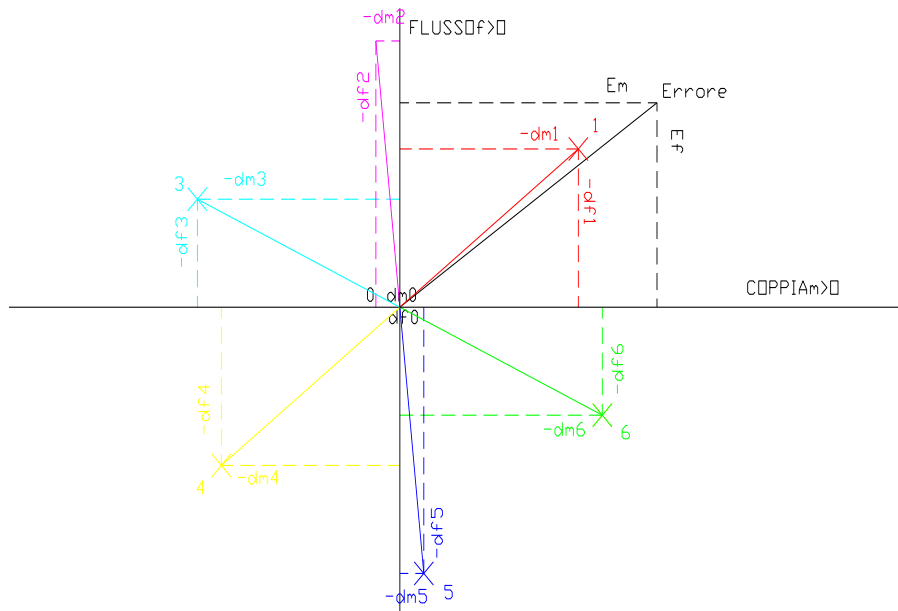


Fig. 5.9 Associazione sul piano complesso dei valori di derivata

Dove abbiamo numerato i vettori derivata da 0 a 6, le cui componenti rappresentano i vari dm df , sono presi con il segno meno in accordo a quanto detto al paragrafo 4.5 : se gli errori Em Ef sono positivi, per abbassarli la derivata della grandezze stesse dovranno rappresentare un calo ed essere quindi negative.

I vari val_m val_f sono dati dal prodotto errore-derivata.

Siamo ora pronti per eseguire la scelta vera e propria, pensiamo quindi ad una iterazione che valuta ogni elemento i –esimo degli array e calcola una funzione di costo pari alla somma dei termini trovati, memorizzando ogni valore su un nuovo vettore c . Il valore minimo di c rappresenta il valore più negativo possibile che posso ottenere dalle tensioni dell’inverter, individuiamo quindi i valori che più si oppongono all’errore e ne permettono la più rapida convergenza a zero.

Facendo riferimento alla figura il “minore in assoluto” è 1 infatti è quello che più si oppone all’errore e presenta i prodotti Em $dm1$ ed Ef $df1$ più negativi.

A questo punto, avendo associato ad ogni posizione dell’array c un ben determinato valore di derivata, ed essendo quest’ ultima conseguenza di un ben preciso v.s. di tensione; riusciamo, mediante una struttura switch, a determinare quali sono le ud uq da attuare sul motore affinché l’errore si riduca.

L’ algoritmo diventa quindi:

```
function [ud_out,uq_out]=
fcn(dm0,df0,dm1,df1,dm2,df2,dm3,df3,dm4,df4,dm5,df5,dm6,df6,Em,Ef,udc,theta_me)

%valori delle derivate in entrata memorizzate su array
der_m=[-dm0,-dm1,-dm2,-dm3,-dm4,-dm5,-dm6];
der_f=[-df0,-df1,-df2,-df3,-df4,-df5,-df6];

%prodotto errore-derivata
val_m=Em*der_m;
val_f=Ef*der_f;

%assegnazione vettore di costo
c=zeros(1,7);
```

```

%variabile di costo
for i= 1:7
    c(i)=val_m(i)+val_f(i);
end

%individuo il minimo
[z,i]=min(c);
k=i;
u_alfa=0;
u_beta=0;

%associo il minimo al vettore di tensione da attuare in uscita
switch k
    case 1 %vettore (0,0,0 o 1,1,1)
        u_alfa=0;
        u_beta=0;
    case 2 %vettore (1,0,0)
        u_alfa=(2/3)*udc;
        u_beta=0;
    case 3 %vettore (1,1,0)
        u_alfa=(2/3)*udc*cos(pi/3);
        u_beta=(2/3)*udc*sin(pi/3);
    case 4 %vettore (0,1,0)
        u_alfa=(2/3)*udc*cos((2*pi)/3);
        u_beta=(2/3)*udc*sin((2*pi)/3);
    case 5 %vettore (0,1,1)
        u_alfa=(2/3)*udc*cos((3*pi)/3);
        u_beta=(2/3)*udc*sin((3*pi)/3);
    case 6 %vettore(0,0,1)
        u_alfa=(2/3)*udc*cos((4*pi)/3);
        u_beta=(2/3)*udc*sin((4*pi)/3);
    case 7 %vettore (1,0,1)
        u_alfa=(2/3)*udc*cos((5*pi)/3);
        u_beta=(2/3)*udc*sin((5*pi)/3);
    otherwise
        -1;
end

ud_out=u_alfa*cos(theta_me)+u_beta*sin(theta_me);
uq_out=-u_alfa*sin(theta_me)+u_beta*cos(theta_me);

```

Lo schema complessivo è:

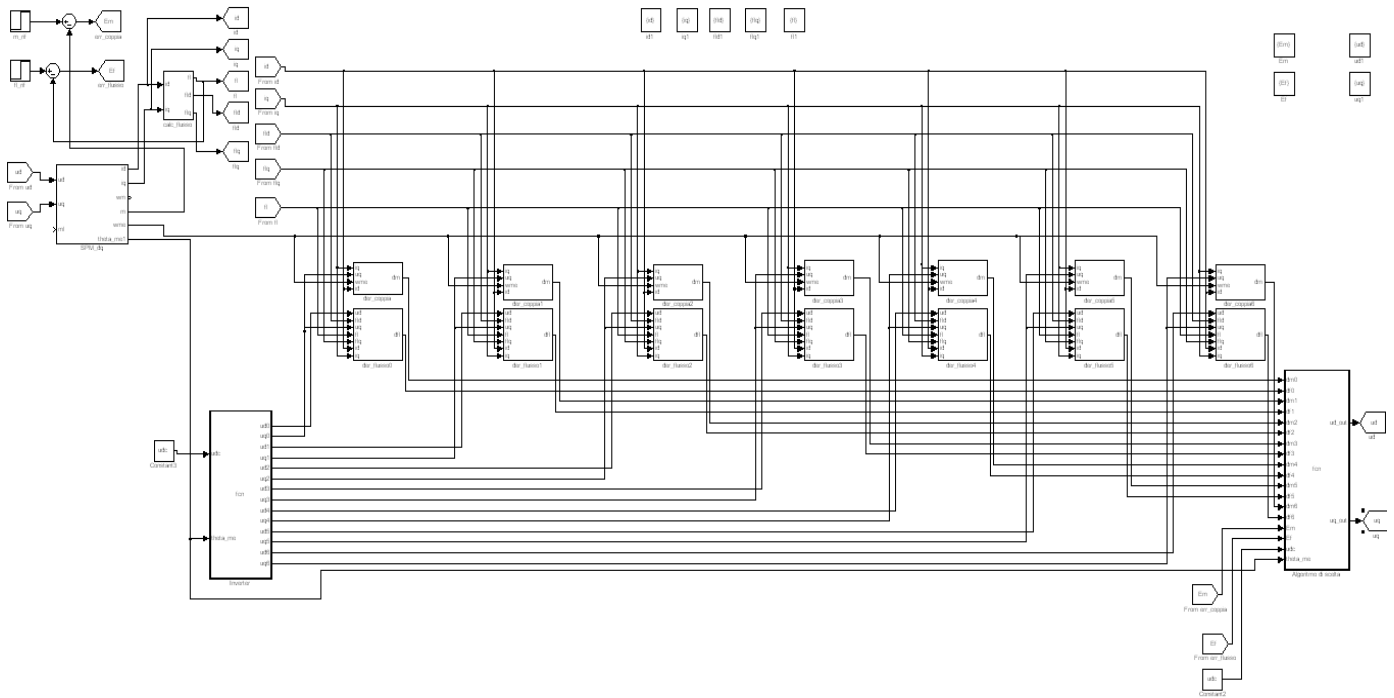


Fig. 5.10 Schema complessivo

Prima di partire con la simulazione impostiamo opportunamente i parametri di configurazione, impostando un passo fisso e relativamente breve per il tempo di simulazione, in modo da ottenere una buona precisione di calcolo: $1e^{-5}sec$.

Otteniamo quindi i seguenti risultati per un tempo di simulazione pari a 0.25 s.

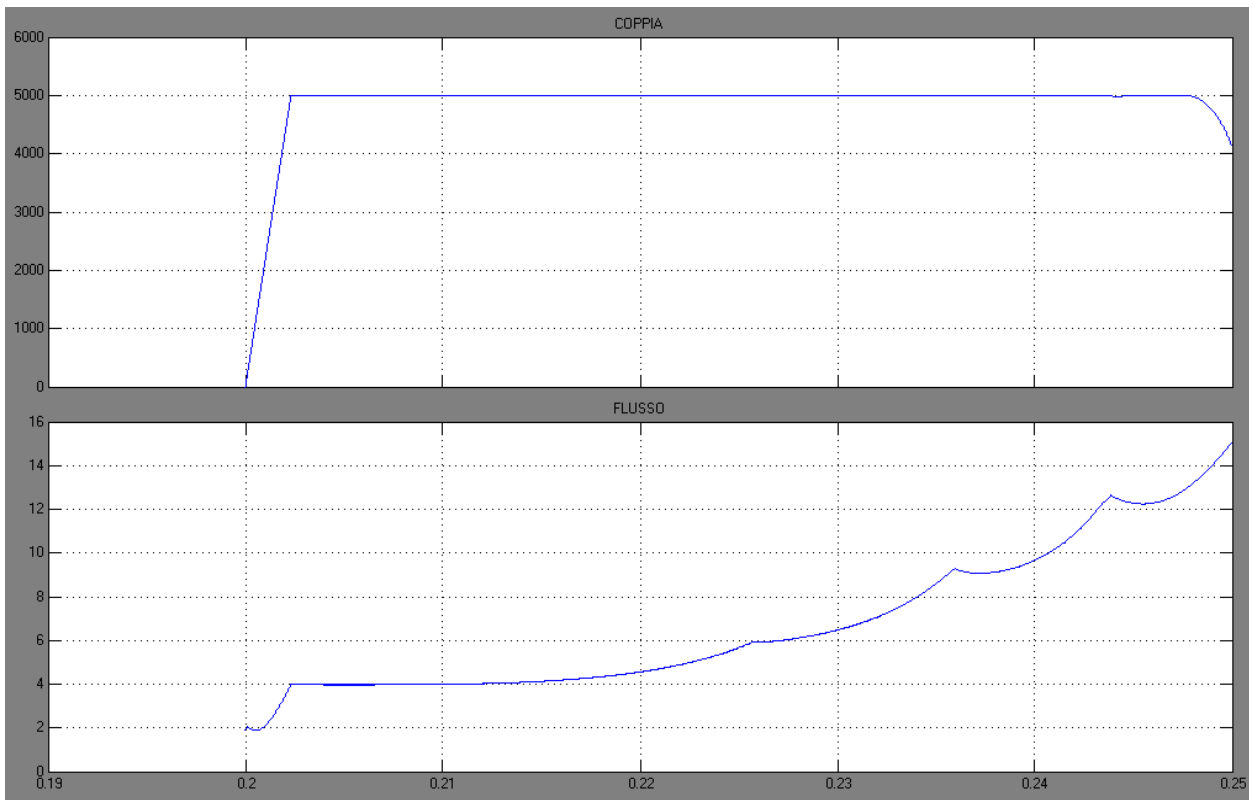


Grafico 5.1 Andamenti Coppia e Flusso controllo base

Come si evince dai grafici il controllo agisce per brevissimo tempo poi il flusso e di conseguenza la coppia non seguono più i riferimenti da noi impostati.

Andiamo allora a ricercare e risolvere le prime due problematiche che si presentano.

In primo luogo osserviamo che i valori della derivata di flusso differiscono di diversi ordini di grandezza rispetto alle derivate di coppia, di conseguenza il vettore di minimo costo non è ben valutato: i valori di df diventano trascurabili rispetto a dm , potrebbero allora presentarsi casi in cui anche se un valore di df è maggiore di un altro in termini di valore assoluto, però il corrispondente valore di coppia dell'ultimo è più elevato, determinando in tal modo un valore minimo nel posto sbagliato.

Una prima soluzione rudimentale consiste nell' introdurre un coefficiente che amplifica i valori di df o meglio di val_f in modo che ogni componente di c venga valutato con il giusto peso.

La riga di codice che si modifica è molto semplice:

```
c(i)=val_m(i)+1e6*val_f(i);
```

I grafici migliorano leggermente in particolare quello del flusso, anche se allungiamo il tempo di simulazione fino a 0.35sec.

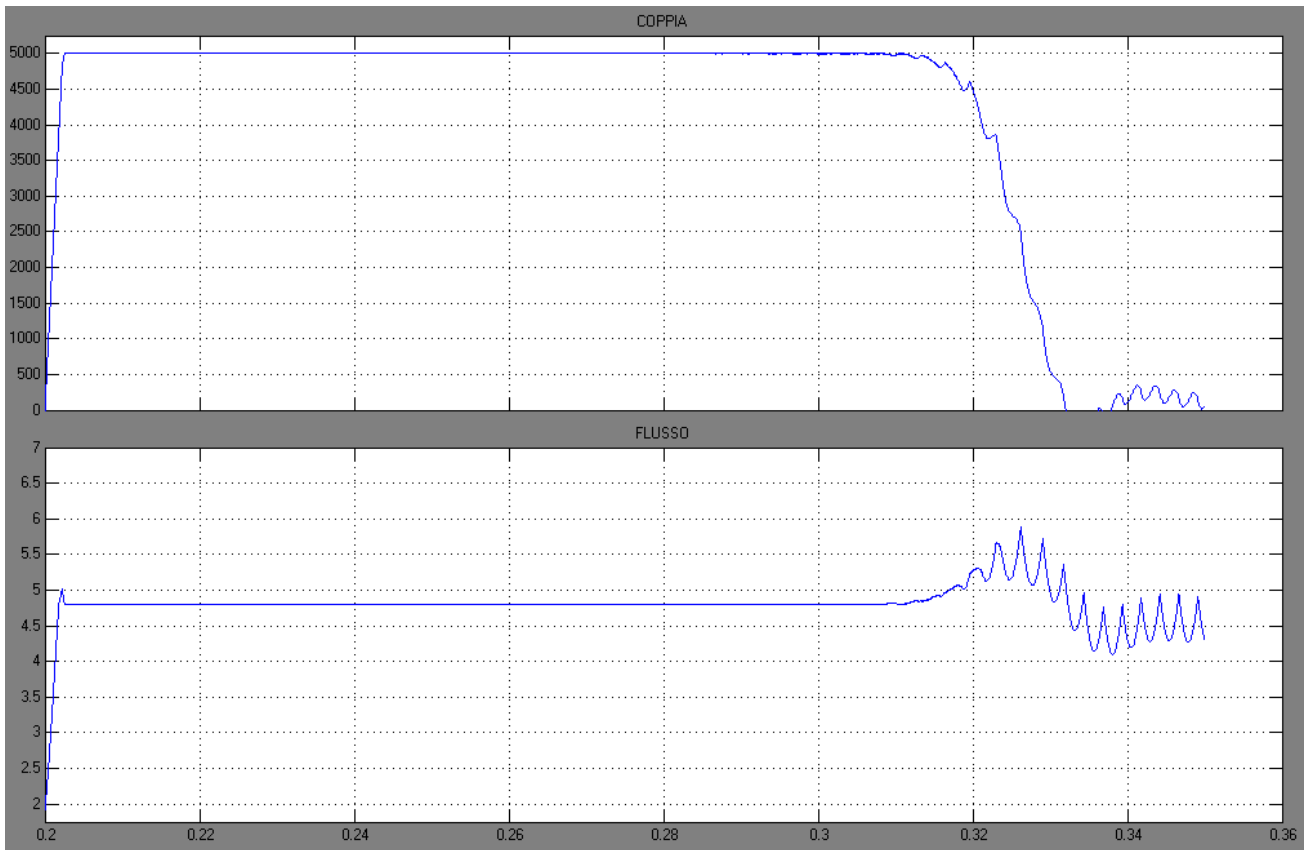


Grafico 5.2 Andamenti Coppia e Flusso controllo base più stringente sul flusso

Altro problema che dobbiamo superare è di natura elettrotecnica: il motore SPM è interessato da grandezze elettriche di una certa importanza, in particolare correnti e flussi. Questo determina una forza elettromotrice alta anche a velocità relativamente basse che il dc-link dell'inverter non è in grado di equilibrare.

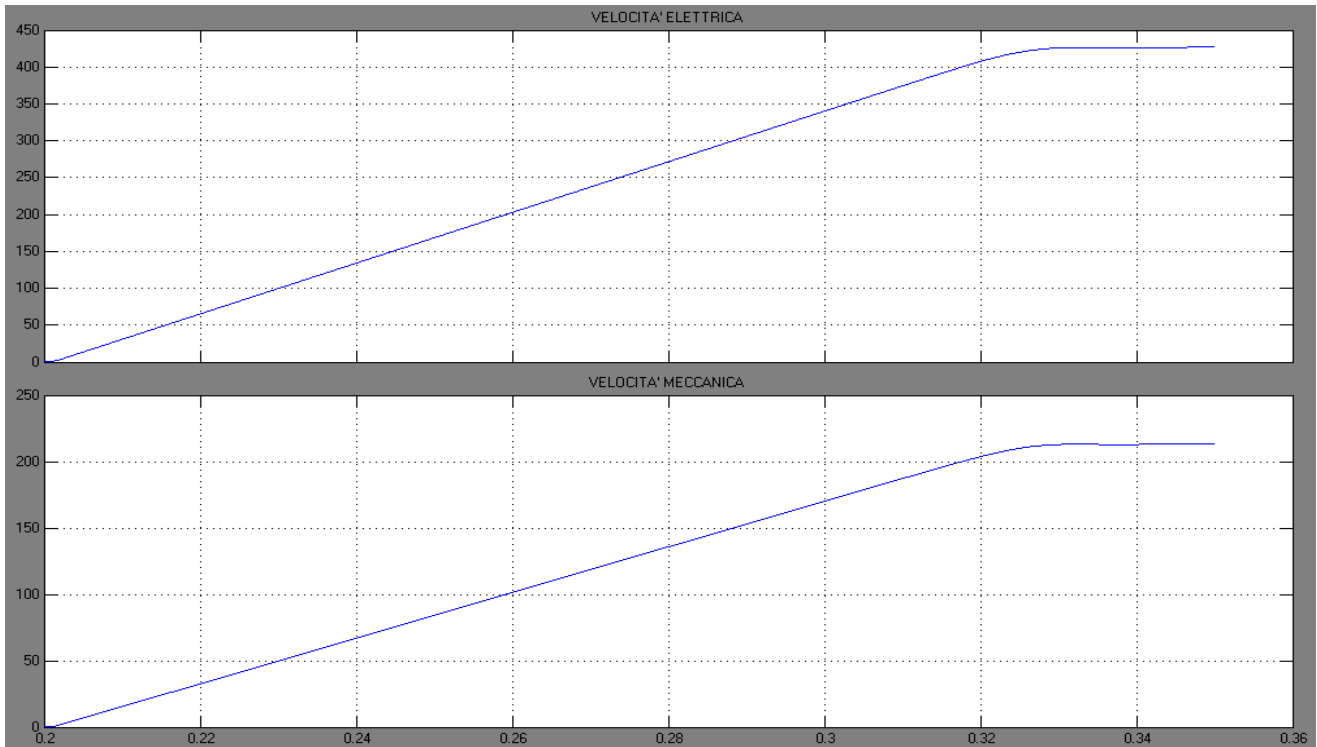


Grafico 5.3 Andamenti Velocità meccanica ed elettrica

Confrontando i grafici notiamo subito che la velocità limite in questione non sarà mai raggiunta da questo tipo di motore possiamo quindi pensare ad un controllo che stia sulla velocità nominale fortunatamente ben al di sotto di quella limite garantendo quindi il corretto funzionamento del DTC.

5.4-controllo DTC con limitazione in velocità

Andiamo ora ad implementare un controllo che ci permetta di svincolarci dalla limitazione della velocità, vogliamo infatti che la macchina funzioni correttamente, sviluppando la coppia di riferimento alla sua velocità nominale.

Osserviamo quindi:

$$n_N = 1200 \text{ rpm}$$

$$\Omega_{mN} = \frac{2\pi 1200}{60} = 125.66 \text{ rad/sec}$$

Di sicuro inferiore alla velocità limite .

Scegliamo allora di controllare la velocità elettrica:

$$\Omega_{meN} = p\Omega_{mN} = 2 \cdot 125.66 = 251.32 \text{ rad/sec}$$

Per agire sulla velocità pensiamo di utilizzare un regolatore di tipo PI che riceve il ingresso il riferimento di velocità da noi impostato e l'anello di retroazione della ω_{me}

In uscita otteniamo una coppia di riferimento M_e , limitata al valore della coppia nominale.

Per calcolare i guadagni proporzionale ed integrale operiamo nel modo classico approssimando la funzione di trasferimento coppia-velocità del sistema motore-carico- inverter con la:

$$\frac{\omega_{me}}{M_e} \cong p \frac{1}{sJ}$$

Essendo il termine inerziale quello predominante.

La funzione di trasferimento del PI la possiamo scrivere come:

$$PI = K_p + \frac{K_I}{s} = \frac{sK_p + K_I}{s}$$

assumendo una frequenza di attraversamento di 50 Hz calcoliamo la f.d.t del sistema complessivo:

$$f dt = \frac{p}{sJ} \frac{sK_p + K_I}{s}$$

Trascuriamo in un primo momento il termine K_I in quanto ipotizziamo che alla frequenza di attraversamento la parte integrale del PI non agisca più.

Alla frequenza di attraversamento il modulo della f.d.t. vale 1 possiamo quindi scrivere:

$$1 \cong \frac{p}{sJ} \frac{sK_p}{s} \Rightarrow K_p \cong \frac{sJ}{p} = 455$$

ed ipotizziamo un:

$$K_I = 50$$

Calcolati i due parametri li inseriamo nel blocco simulink che incorpora il regolatore PID, definendo anche i valori limite della coppia in uscita la quale altri non è che la coppia di riferimento che comanda il motore

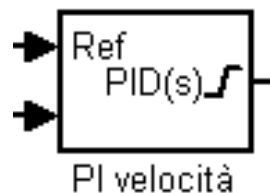


Fig. 5.11 PI controllo velocità

Il sistema di controllo è essenzialmente uguale al caso precedente con l'unica aggiunta del regolatore PI che riceve la ω_{me} di feedback la confronta con la ω_{me_rif} e fornisce la m_rif . Otteniamo allora:

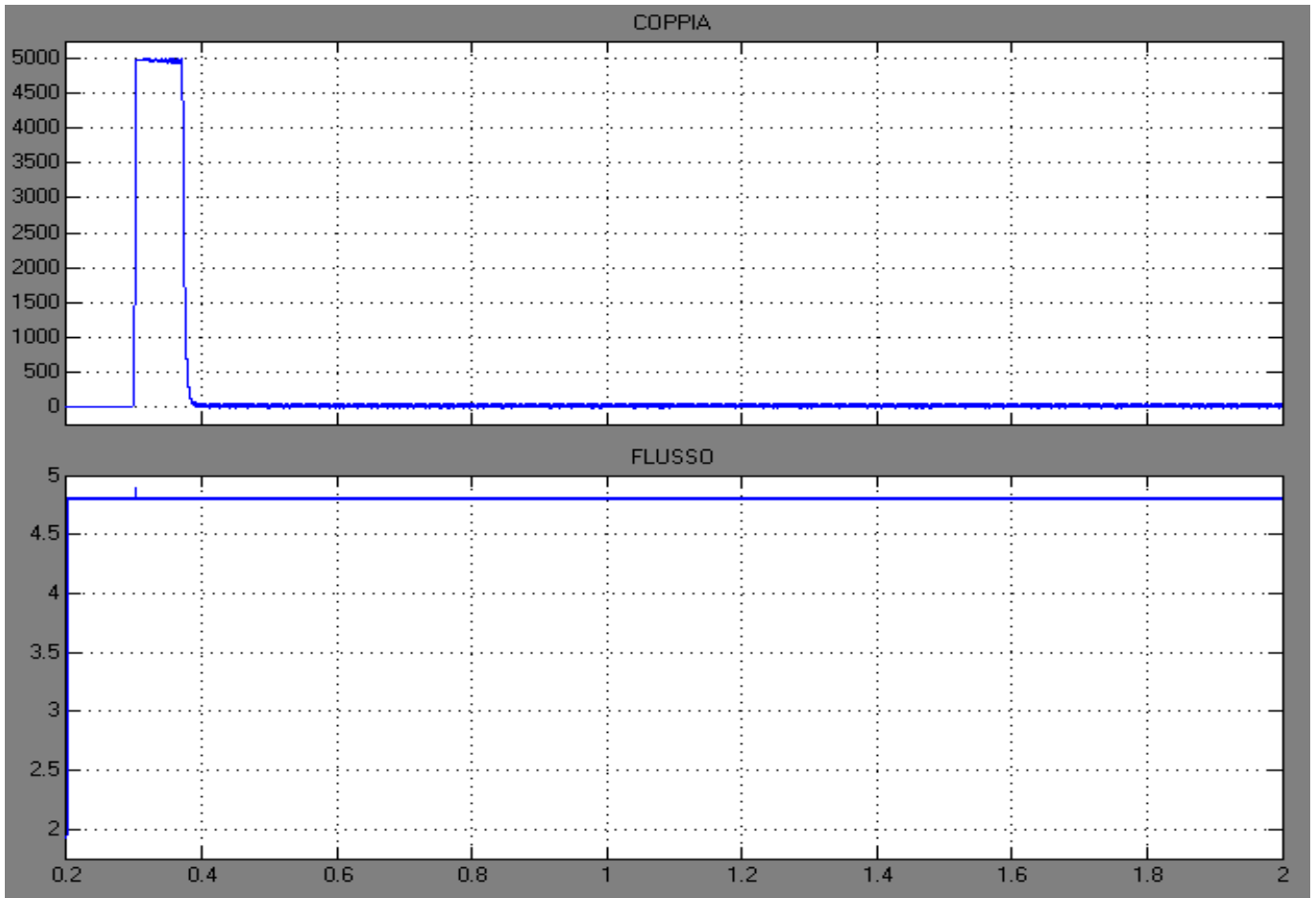


Grafico 5.4 Andamenti Coppia e Flusso controllo velocità senza coppia di carico

Le velocità:

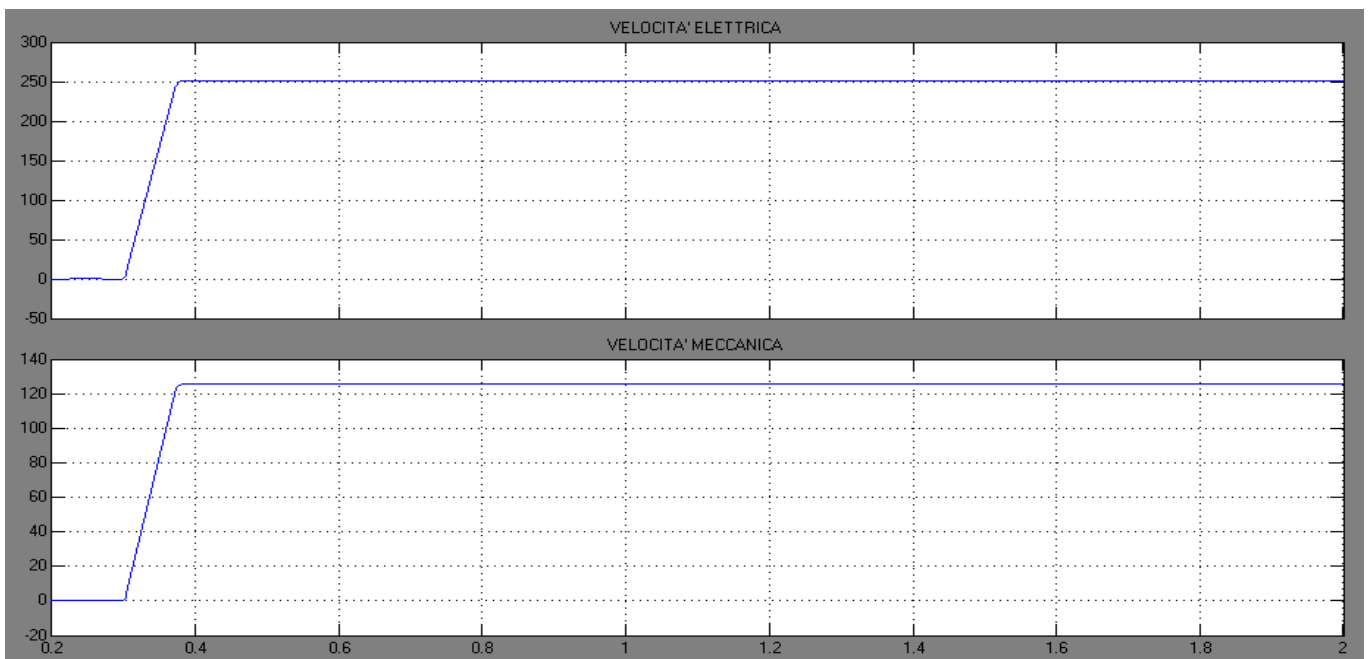


Grafico 5.5 Andamenti Velocità meccanica ed elettrica

Vogliamo ora simulare il caso reale relativo al carico del motore: costruiamo un ideale step di coppia positivo e negativo di valore $\pm ml$ che simula un carico connesso all'albero della macchina:

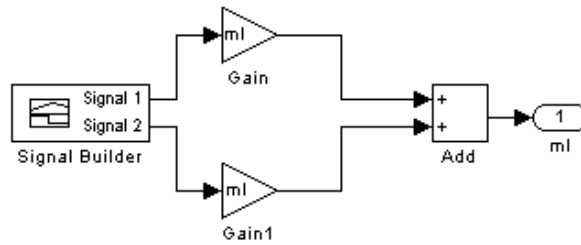


Fig. 5.12 costruzione step di coppia

il motore reagisce di conseguenza subendo un'accelerazione o una frenatura a seconda dei casi, ma comunque portandosi alla velocità nominale di funzionamento dopo un certo tempo, a seconda delle costanti di tempo in gioco:

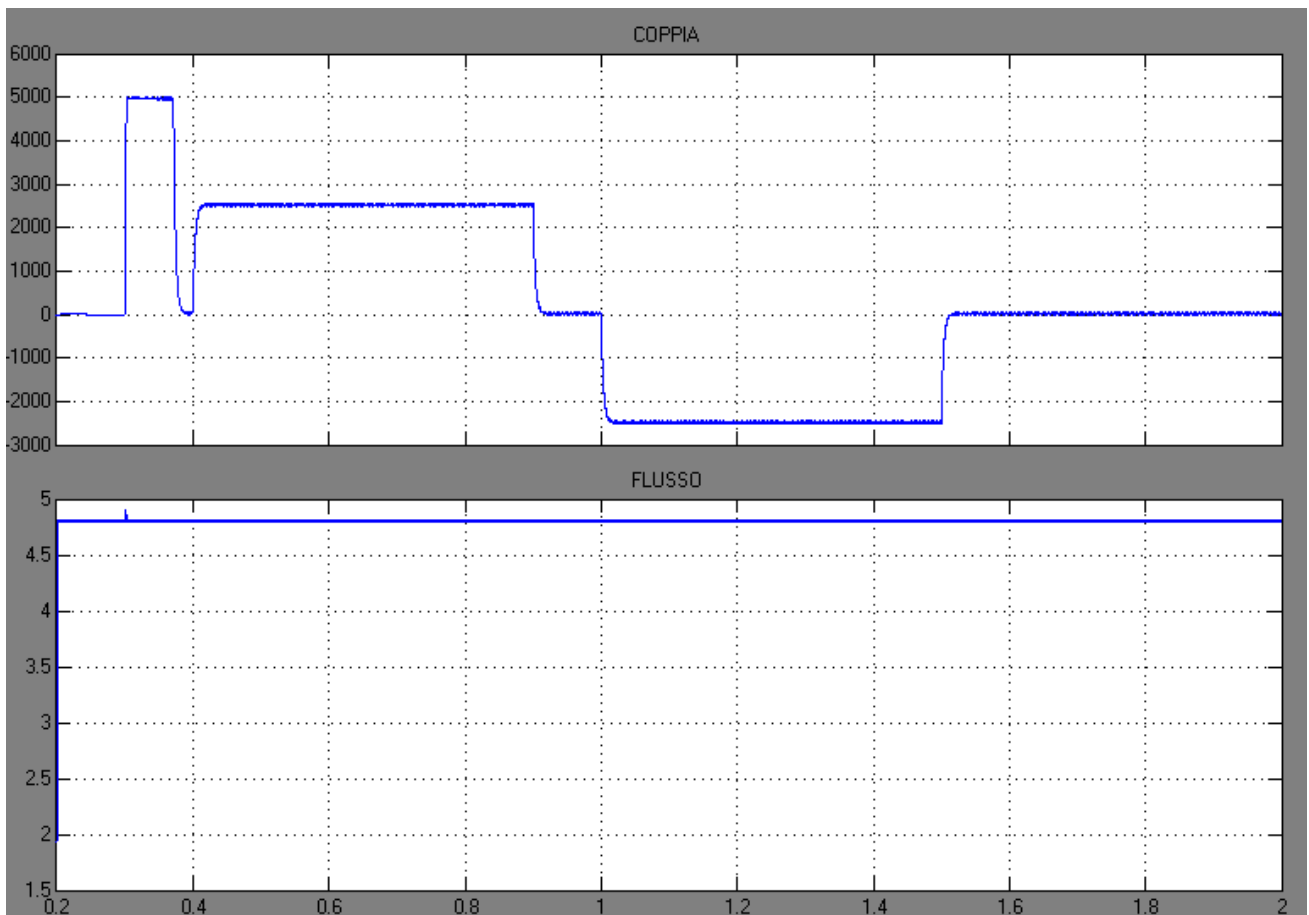


Grafico 5.6 Andamenti Coppia e Flusso controllo velocità con coppia di carico $ml=2500 \text{ Nm}$

Le velocità invece:

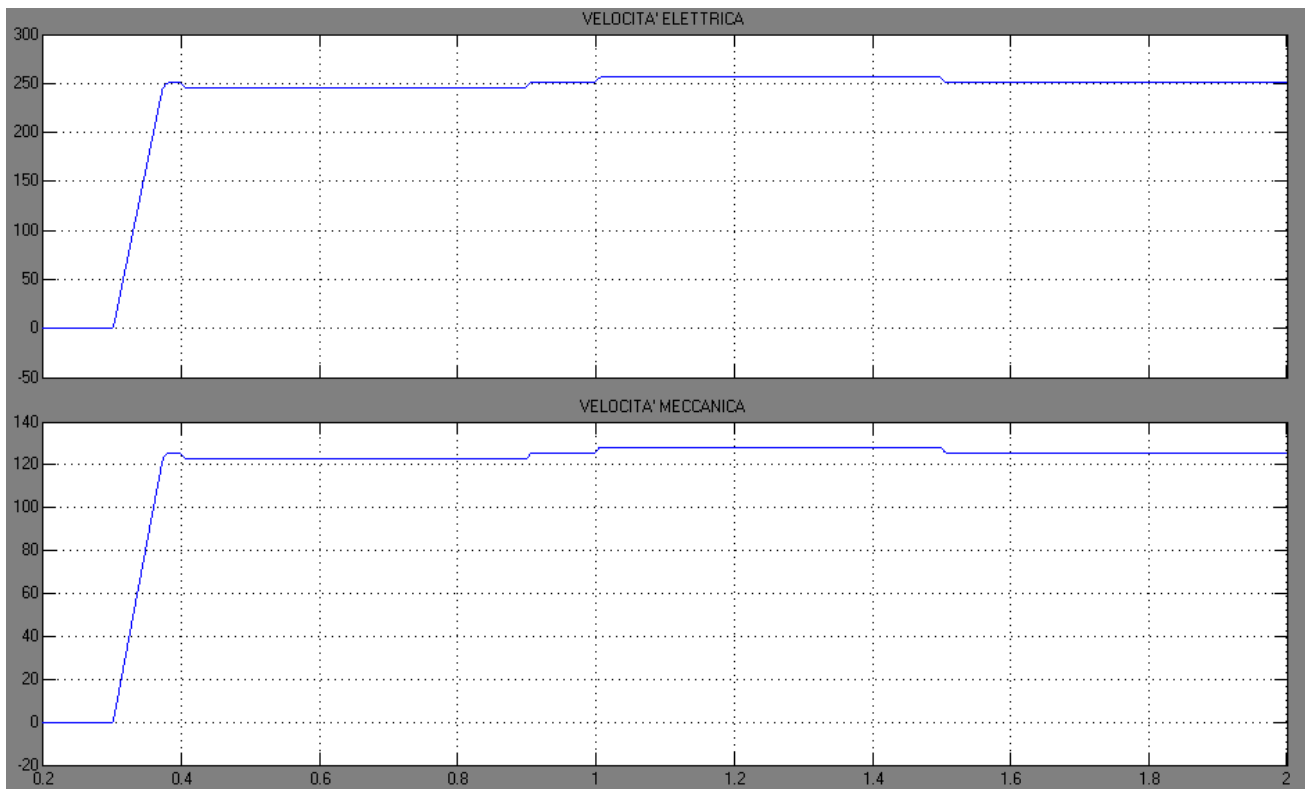


Grafico 5.7 Andamenti Velocità meccanica-elettrica controllo velocità con coppia di carico $mI=2500 Nm$

Abbiamo quindi implementato un controllo base che difficilmente potrebbe essere implementato nella pratica in quanto presenta frequenze di switching medie per componente troppo elevate ed inoltre è un tipo di controllo che interviene sempre, bisogna pensare ad una procedura dotata di un certo gap di tolleranza eventualmente da noi impostato in modo che l'algoritmo intervenga solo se si esce dalla banda di isteresi scelta dall'operatore.

Una costrizione all'errore nullo è infatti pura utopia a causa delle inerzie del sistema e sottoporrebbe i componenti a maggiori stress diminuendone la vita utile.

5.5-Valutazione errore

Procediamo ora modificando leggermente il blocco principale dello schema sulla base delle seguenti osservazioni: in primo luogo come già accennato vogliamo che il controllo intervenga

solamente se l'errore esce da una certa fascia di tolleranza, inoltre se valutiamo gli ordini di grandezza delle relative derivate di coppia e flusso notiamo che sono molto diversi tra loro dobbiamo quindi normalizzare tutti i valori per ottenere un controllo più preciso e corretto.

Quindi dividiamo dapprima tutti i valori per le loro grandezze nominali, in modo da avere valori in per unità, avendo l'accortezza di moltiplicare i termini di flusso per un opportuno coefficiente generalmente maggiore di uno necessario per uniformare la variazione del flusso rispetto a quella della coppia.

Inseriti i coefficienti opportuni modifichiamo l'algoritmo andando ad inserire la condizione che il controllo intervenga se supera un certo E_{lim} , altrimenti la tensione da applicare all'inverter resta quella attuata in precedenza.

Il tutto si traduce nel nostro schema simulink inserendo una semplice struttura if-else che serve per valutare la condizione di errore, e aggiungendo tre ingressi e tre uscite che collegati a tre particolari blocchi *memory* ci permettono di tenere memorizzati i valori di tensione e l'indice corrispondente all'istante precedente, i quali verranno applicati quando l'errore è contenuto nel range di tolleranza.

Quindi il blocco esternamente diventa:

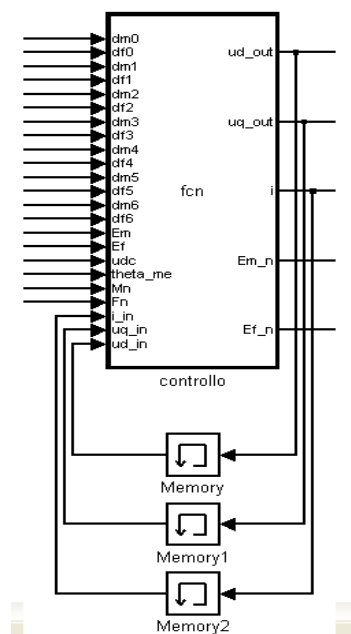


Fig. 5.13 Controllo errore

L'algoritmo pur mantenendo la sua struttura base viene lievemente modificato nella definizione dei dati e nell'aggiunta di una if-structure:

```
function [ud_out,uq_out,i,Em_n,Ef_n]=
fcn(dm0,df0,dm1,df1,dm2,df2,dm3,df3,dm4,df4,dm5,df5,dm6,df6,Em,Ef,udc,theta_me,M
n,Fn,i_in,uq_in,ud_in)

%assegnazione delle variabili, memorizzazione dei dati normalizzandoli e
%definendo errore limite ed il coefficiente a per equilibrare le df con le dm
a=1.8;
E_lim=5e-2;
der_m=[-dm0,-dm1,-dm2,-dm3,-dm4,-dm5,-dm6,0];
der_f=[-df0,-df1,-df2,-df3,-df4,-df5,-df6,0];

%normalizzazioni
der_m_n=(1/Mn)*der_m;
```

```

der_f_n=(a/Fn)*der_f;
Em_n=Em/Mn;
Ef_n=a*(Ef/Fn);

val_m=Em_n*der_m_n;
val_f=Ef_n*der_f_n;

c=zeros(1,7);

%definizione dell'errore
E=sqrt(Em_n^2+Ef_n^2);

ud_out=0;
uq_out=0;

% inizio valutazione errore
if E>E_lim;
for i= 1:8
c(i)=val_m(i)+val_f(i);
end

[z,i]=min(c);
k=i;
u_alfa=0;
u_beta=0;

switch k
case 1
u_alfa=0;
u_beta=0;
case 2
u_alfa=(2/3)*udc;
u_beta=0;
case 3
u_alfa=(2/3)*udc*cos(pi/3);
u_beta=(2/3)*udc*sin(pi/3);
case 4
u_alfa=(2/3)*udc*cos((2*pi)/3);
u_beta=(2/3)*udc*sin((2*pi)/3);
case 5
u_alfa=(2/3)*udc*cos((3*pi)/3);
u_beta=(2/3)*udc*sin((3*pi)/3);
case 6
u_alfa=(2/3)*udc*cos((4*pi)/3);
u_beta=(2/3)*udc*sin((4*pi)/3);
case 7
u_alfa=(2/3)*udc*cos((5*pi)/3);
u_beta=(2/3)*udc*sin((5*pi)/3);
case 8
u_alfa=0;
u_beta=0;
otherwise
-1;
end

ud_out=u_alfa*cos(theta_me)+u_beta*sin(theta_me);
uq_out=-u_alfa*sin(theta_me)+u_beta*cos(theta_me);

else

ud_out=ud_in;

```

```
uq_out=uq_in;  
i=i_in;  
  
end
```

Ottenendo i seguenti grafici di coppia e flusso (senza la coppia di carico):

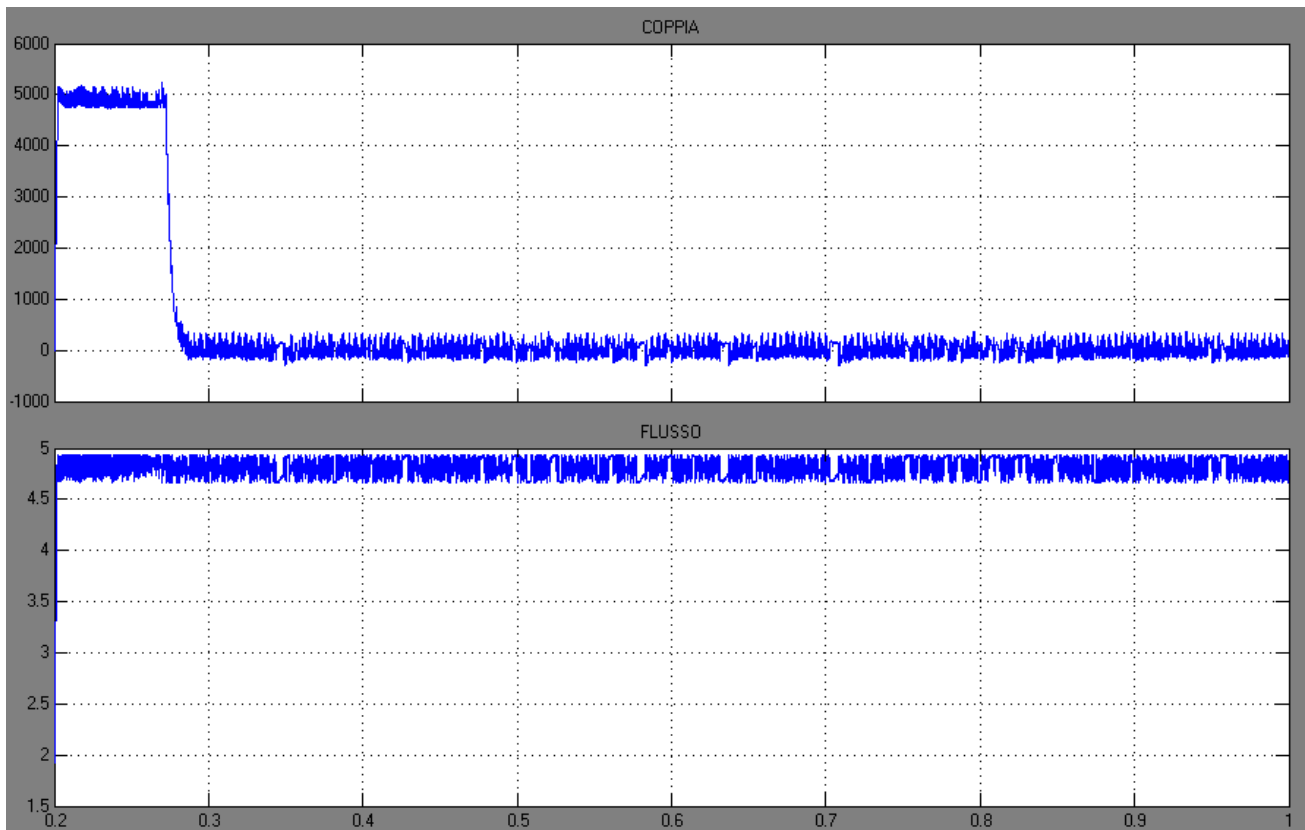


Grafico 5.8 Andamenti Coppia e Flusso controllo velocità con limitazione errore

Possiamo anche graficare E_{m_n} E_{f_n} , avremo in tal modo una rappresentazione intuitiva che esprime su un piano l'ellisse limite dell'errore:

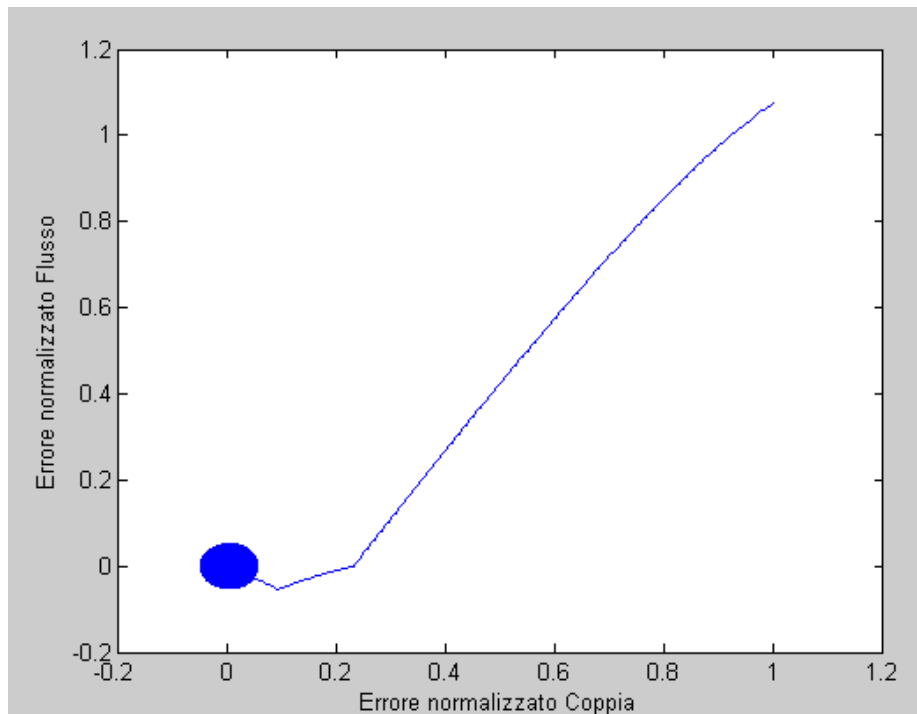


Grafico 5.8 Rappresentazione sul piano complesso degli errori

Notiamo che dopo i primi passi iniziali l'errore forma una figura tendenzialmente ellittica le cui dimensioni si mantengono entro i valori di E_{lim} .

5.6-Controllo commutazioni

Nella realtà gli inverter sono costituiti da componenti di potenza quali IGBT per esempio che come ben noto ad ogni commutazione presentano perdite relativamente importanti. Se noi riusciamo a pensare ad un modo che riduca al minimo le commutazioni nell'inverter necessarie al controllo abbassiamo la frequenza di switching media per componente e di conseguenza calano le perdite. Non solo, riusciamo anche a limitare i disturbi elettromagnetici emessi dal dispositivo di controllo del motore.

Per ridurre questi switch dobbiamo dapprima studiare un modo per analizzarli, pensiamo allora ad una struttura che ricevendo in ingresso l'indice i corrispondente alla tensione attuata, associa uno stato logico relativo alle posizioni degli interruttori nell'inverter. Successivamente se noi calcoliamo la differenza binaria tra uno stato logico ed il suo precedente e sommiamo i vari termini del risultato otteniamo il numero di commutazioni effettuate tra uno stato ed il suo successivo.

Avendo quindi il numero di commutazioni effettuate e l'intervallo di simulazione è facile ricavare la frequenza di switching media per componente dalla formula:

$$f_{sw_{comp}} = \frac{\sum \text{commutazioni}}{6T_{oss}}$$

Dove T_{oss} è il periodo in cui valutiamo la frequenza, $\sum \text{commutazioni}$ è la sommatoria di tutti i cambi di stato degli interruttori che si sono succeduti durante l'intervallo di osservazione e 6 perché abbiamo sei componenti che cambiano posizione.

Osserviamo allora la struttura che ci permette questo calcolo:

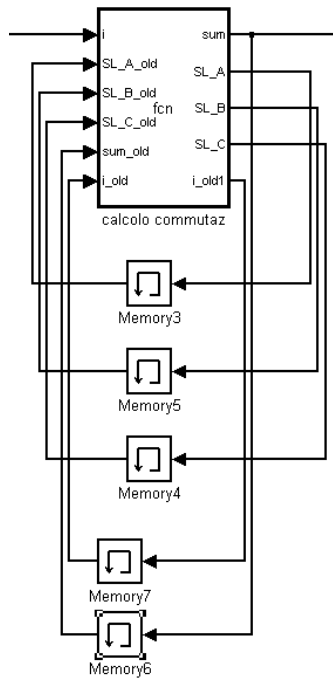


Fig. 5.14 Calcolo delle commutazioni

Riceve in ingresso i , con una struttura switch associamo ad ogni indice uno stato logico ben definito e poi sempre con l'utilizzo dei blocchi di memoria già visti, andiamo a salvare il valore degli stati logici in modo che ad ogni passo di calcolo possiamo eseguire la differenza tra quello attuale ed il precedente.

```
function [sum,SL_A,SL_B,SL_C,i_old1]=
fcn(i,SL_A_old,SL_B_old,SL_C_old,sum_old,i_old)
```

```
SL_A=SL_A_old;
SL_B=SL_B_old;
SL_C=SL_C_old;
i_old1=i_old;
```

```
%associo all'indice lo stato logico corrispondente
%se è diverso dall'indice dello stato precedente
```

```
if i ~= i_old1;
    switch i
        case 1
            SL_A=0;
            SL_B=0;
            SL_C=0;

            case 2
                SL_A=1;
                SL_B=0;
                SL_C=0;

            case 3
                SL_A=1;
```

```

SL_B=1;
SL_C=0;

    case 4
SL_A=0;
SL_B=1;
SL_C=0;

    case 5
SL_A=0;
SL_B=1;
SL_C=1;

    case 6
SL_A=0;
SL_B=0;
SL_C=1;

    case 7
SL_A=1;
SL_B=0;
SL_C=1;

    case 8
SL_A=1;
SL_B=1;
SL_C=1;

    otherwise
        -1;
end

%calcolo le commutazioni tra un indice e l'altro
sum=abs(SL_A-SL_A_old)+abs(SL_B-SL_B_old)+abs(SL_C-SL_C_old);
i_old1=i;

else

    %se invece sono uguali la somma sarà nulla
    i_old1=i;

    SL_A=SL_A_old;
    SL_B=SL_B_old;
    SL_C=SL_C_old;
    sum=abs(SL_A-SL_A_old)+abs(SL_B-SL_B_old)+abs(SL_C-SL_C_old);
end

```

Possiamo allora visualizzare l'evolversi delle commutazioni durante il tempo di simulazione relativo al raggiungimento della coppia nominale: 0.3 sec.

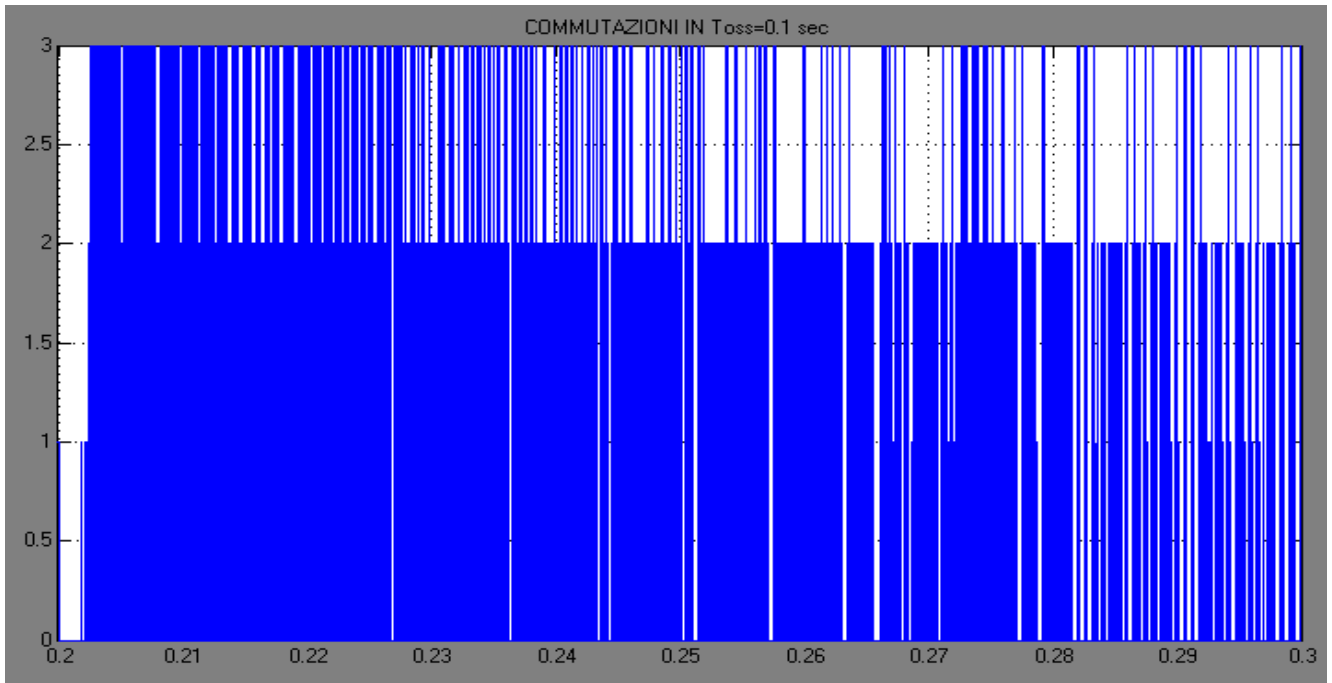


Grafico 5.9 Calcolo delle commutazione senza grafo-stati

In generale si avremo un massimo di tre che indica lo switch di tutti e tre gli interruttori ed un minimo di zero quando l'errore è ancora nella fascia di tolleranza e il vettore di tensione applicato resta costante da un istante di calcolo al successivo.

Calcoliamo la $f_{sw_{comp}}$ salvando nel workspace le commutazioni. Creiamo quindi un vettore di nome comm e sommando le componenti abbiamo il termine $\sum commutazioni$.

$$f_{sw_{comp}} = \frac{2296}{6 \cdot 0.1} = 3827 \text{ Hz}$$

Calcolata la frequenza di switching media per componente (usando la formula appena analizzata) notiamo che per controlli di questo tipo non è poi così elevata, possiamo comunque migliorare le cose.

Possiamo intervenire modificando l'algoritmo di controllo secondo la seguente filosofia: costringere la commutazione in modo che si abbia al massimo un solo cambio di stato nell'inverter, una sola inversione degli IGBT.

A tal fine osserviamo il seguente grafo:

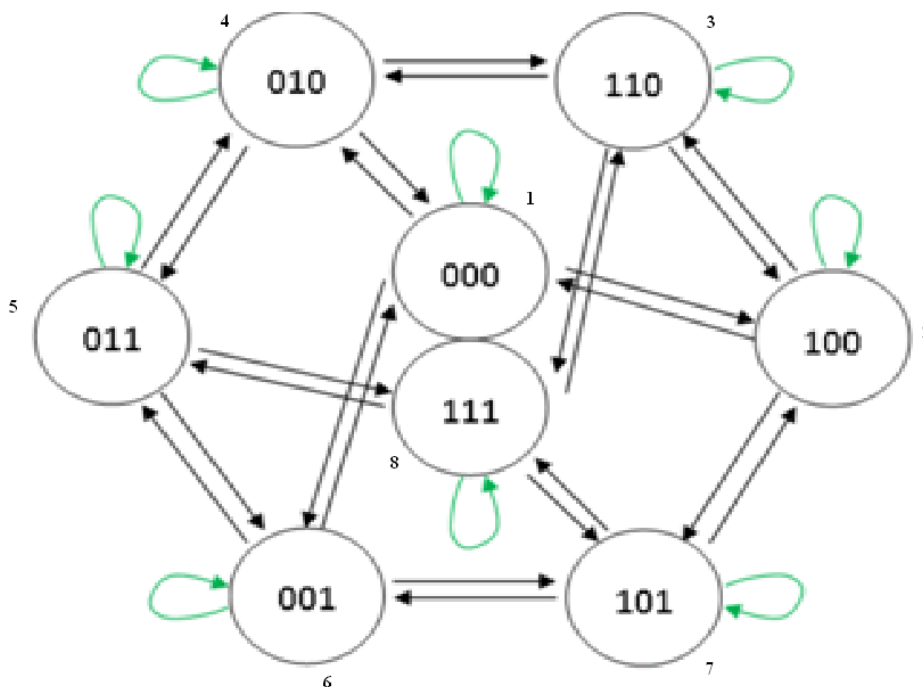


Fig. 5.15 Possibili stati attuabili (vedi [1])

Non è più possibile passare da uno stato logico ad un qualsiasi altro, ma solo ed esclusivamente a quelli che comportano un solo cambio.

Il grafo può essere tradotto nella seguente tabella:

Stati Logici ADESSO	Stati Logici NUOVI POSSIBILI			
1	2	4	6	Se stesso
2	1	3	7	Se stesso
3	2	4	8	Se stesso
4	1	3	5	Se stesso
5	4	6	8	Se stesso
6	1	5	7	Se stesso
7	2	6	8	Se stesso
8	3	5	7	Se stesso

Osserviamo quindi che obblighiamo le commutazioni a seguire un certo ordine, dettato dal fatto che tra un cambio e il successivo varia la sua posizione al massimo un solo interruttore.

Abbiamo inoltre introdotto un ulteriore vettore nullo l'ottavo, in modo da simulare con completezza tutte le posizioni possibili dei vari switch

Possiamo implementare tutto questo nell'algoritmo mediante una semplice struttura if, dopo aver aggiunto ai vari array una componente ulteriore di valore pari al vettore nullo.

L'algoritmo diventa:

```
function [ud_out,uq_out,i,Em_n,Ef_n]=
fcn(dm0,df0,dm1,df1,dm2,df2,dm3,df3,dm4,df4,dm5,df5,dm6,df6,Em,Ef,udc,theta_me,M
n,Fn,i_in,uq_in,ud_in)

%assegnazione delle variabili
a=1.8
E_lim=5e-2;

der_m=[-dm0,-dm1,-dm2,-dm3,-dm4,-dm5,-dm6,0];
der_f=[-df0,-df1,-df2,-df3,-df4,-df5,-df6,0];

%normalizzazioni,memorizzazione dei dati, con l'aggiunta di un valore di
derivata %pari a quello del vettore nullo

der_m_n=(1/Mn)*der_m;
der_f_n=(a/Fn)*der_f;

Em_n=Em/Mn;
Ef_n=a*(Ef/Fn);

val_m=Em_n*der_m_n;
val_f=Ef_n*der_f_n;

val_m(8)=val_m(1);
val_f(8)=val_f(1);

c=zeros(1,8);

E=sqrt(Em_n^2+Ef_n^2);

ud_out=0;
uq_out=0;

% inizio valutazione errore
if E>E_lim;
for i= 1:8
%scelgo vettori furbi (vedi grafo Matthias)
if (i_in==1 && (i==2 || i==4 || i==6 || i==1) ) || ...
(i_in==2 && (i==1 || i==3 || i==7 || i==2) ) || ...
(i_in==3 && (i==2 || i==4 || i==8 || i==3) ) || ...
(i_in==4 && (i==1 || i==3 || i==5 || i==4) ) || ...
(i_in==5 && (i==4 || i==6 || i==8 || i==5) ) || ...
(i_in==6 && (i==1 || i==5 || i==7 || i==6) ) || ...
(i_in==7 && (i==2 || i==6 || i==8 || i==7) ) || ...
(i_in==8 && (i==3 || i==5 || i==7 || i==8) )

c(i)=val_m(i)+val_f(i);
else
c(i)=1e9;
end
end
end
```

```

    end
end

[z, i]=min(c);
k=i;
u_alfa=0;
u_beta=0;

switch k
    case 1
        u_alfa=0;
        u_beta=0;
    case 2
        u_alfa=(2/3)*udc;
        u_beta=0;
    case 3
        u_alfa=(2/3)*udc*cos(pi/3);
        u_beta=(2/3)*udc*sin(pi/3);
    case 4
        u_alfa=(2/3)*udc*cos((2*pi)/3);
        u_beta=(2/3)*udc*sin((2*pi)/3);
    case 5
        u_alfa=(2/3)*udc*cos((3*pi)/3);
        u_beta=(2/3)*udc*sin((3*pi)/3);
    case 6
        u_alfa=(2/3)*udc*cos((4*pi)/3);
        u_beta=(2/3)*udc*sin((4*pi)/3);
    case 7
        u_alfa=(2/3)*udc*cos((5*pi)/3);
        u_beta=(2/3)*udc*sin((5*pi)/3);
    case 8
        u_alfa=0;
        u_beta=0;
    otherwise
        -1;
end

ud_out=u_alfa*cos(theta_me)+u_beta*sin(theta_me);
uq_out=-u_alfa*sin(theta_me)+u_beta*cos(theta_me);
else

    ud_out=ud_in;
    uq_out=uq_in;
    i=i_in;
end

```

In tal modo quando interviene il controllo perché l'errore è maggiore del voluto, gli indici applicabili seguono la logica del grafo e se osserviamo l'andamento delle commutazioni si nota che al massimo si ha una sola commutazione.

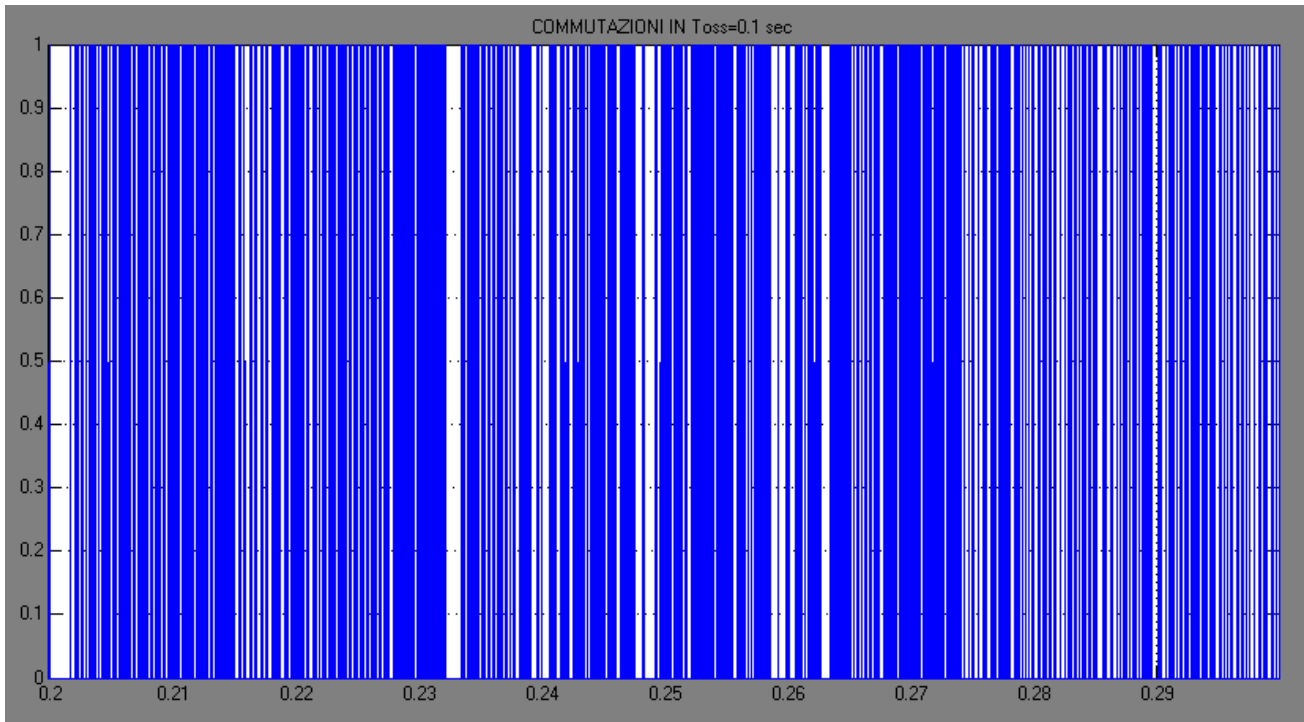


Grafico 5.9 Calcolo delle commutazione con grafo-stati

La frequenza cala fino a:

$$f_{sw_{comp}} = \frac{1360}{60.1} = 2267 \text{ Hz}$$

Otteniamo i seguenti andamenti di coppia e flusso per un tempo di simulazione pari a 1 sec.

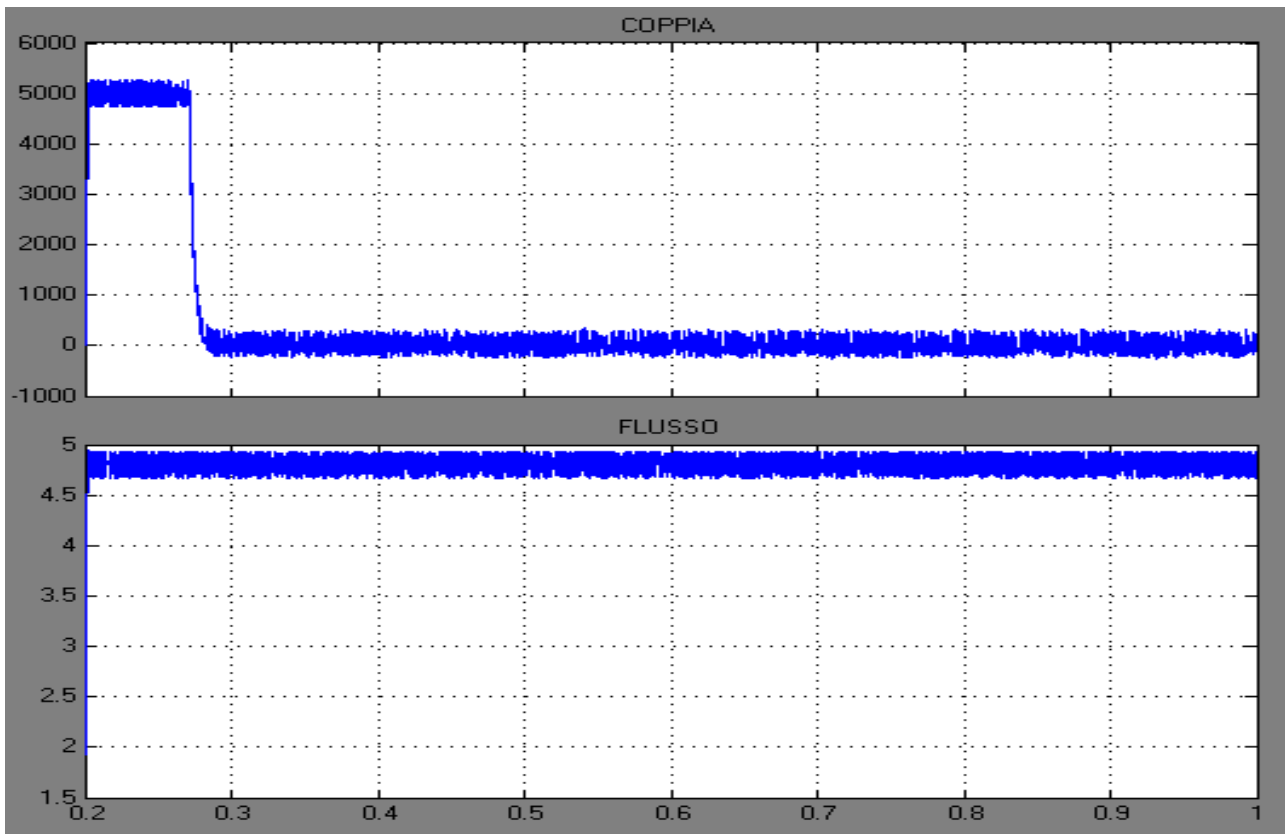


Grafico 5.10 Andamenti Coppia e Flusso controllo velocità con limitazione errore e grafo stati logici

Come ultima osservazione prendiamo il grafico dell'errore 5.8 relativo al caso senza controllo delle commutazioni e andiamo ad ingrandirlo:

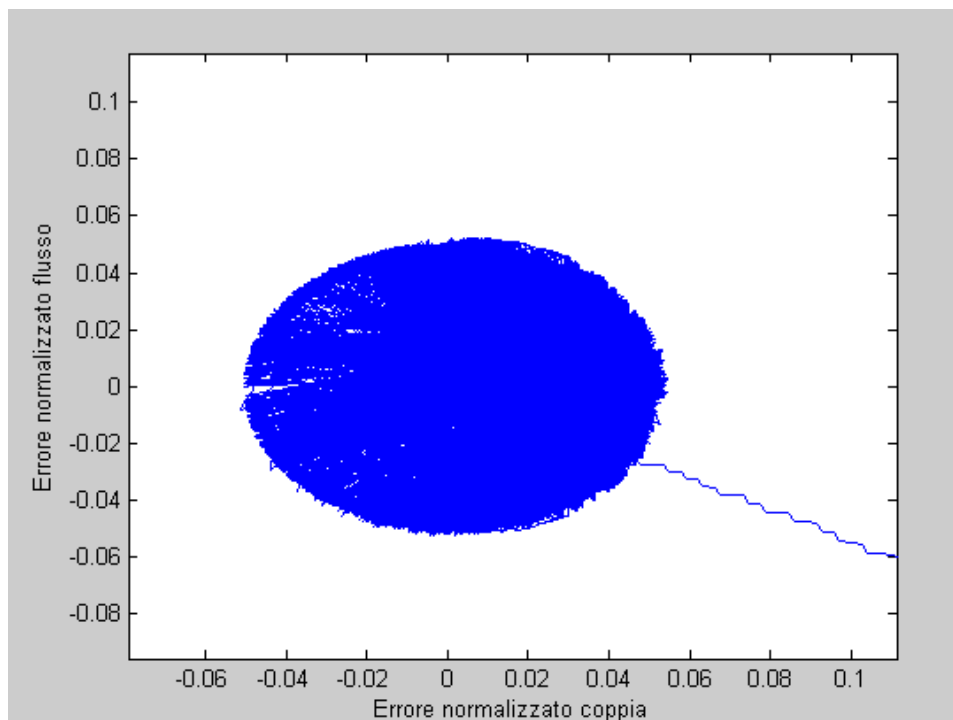


Grafico 5. 11Rappresentazione sul piano complesso degli errori ingrandito

Prendiamo anche l'ingrandimento dell'errore con controllo delle commutazioni:

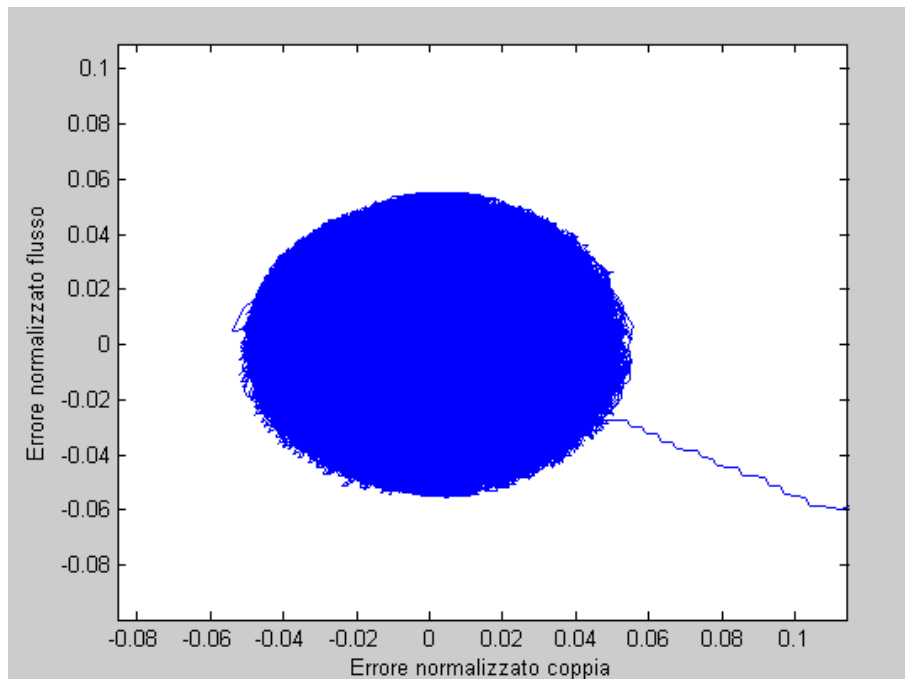


Grafico 5.12 Rappresentazione sul piano complesso degli errori con controllo delle commutazioni, grafico ingrandito

Ciò che si nota subito è la diversa “densità” dei due, infatti con il controllo delle commutazioni costringo la scelta imponendo vettori che magari non sono quelli ottimali. Di conseguenza l'errore esce dal limite più volte del caso precedente si cambia stato logico più volte, anche se lo switch che commuta è solo uno.

6-SPERIMENTAZIONE

6.1-Introduzione:

In questo capitolo si descriveranno i risultati sperimentali preliminarmente il controllo visto in ambito di simulazione su di un motore IPM, dovremo quindi fare alcuni ragionamenti preliminari per prepararci alla realtà pratica, arrivando ad un tipo di comando leggermente differente.

Dovremo allora provvedere alla scrittura dell'algoritmo nel linguaggio C per riuscire a programmare il microprocessore che comanda l'inverter ed il motore brushless Lafert IPM. Eseguendo la sperimentazione abbiamo adottato una macchina più piccola di quella simulata in matlab, ragion per cui adatteremo i calcoli ai differenti valori dei parametri.

6.2-Unicaembedded:

Come primo passo vogliamo innanzitutto compattare lo schema di controllo visto in simulink per prepararci al meglio all'implementazione.

Il linguaggio di programmazione C infatti, essendo di basso livello non prevede l'uso di figure e blocchetti preimpostati ma esclusivamente calcoli matematici usando valori presenti su definite celle di memoria, risulta quindi opportuno pensare ad un'unica funzione che incorpora tutte le componenti del controllo usando esclusivamente passaggi appunto matematici.

Altra riflessione è stata fatta riguardo le tensioni attuate: se in ambito di simulazione possiamo permetterci di applicare le grandezze vere e proprie, in ambito reale le tensioni sono date da delle specifiche posizioni degli interruttori all'interno dell'inverter, dovremo quindi rendere in qualche modo disponibili le indicazioni degli stati da dare al dispositivo per far funzionare correttamente il controllo.

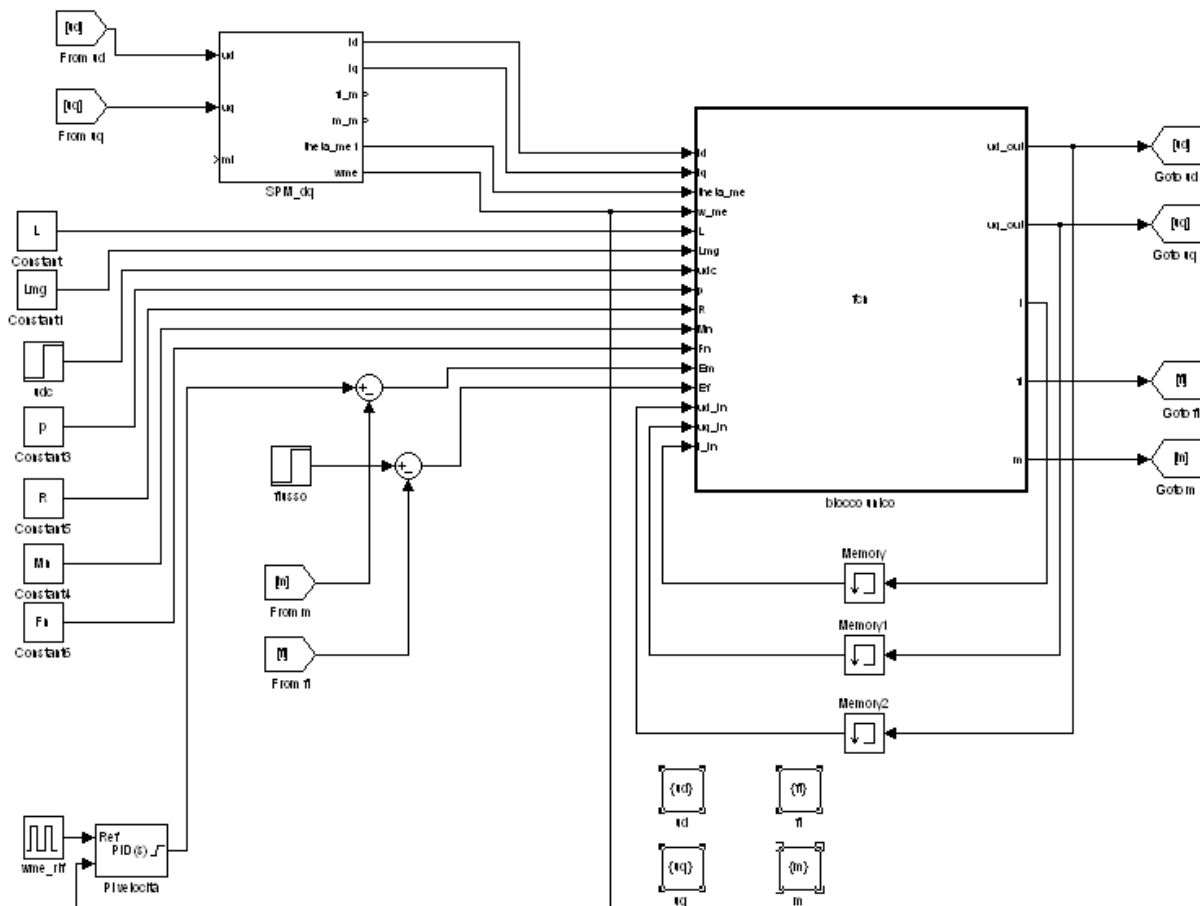


Fig. 6.1 schema di controllo con unica funzione

L'algorithmo globale di controllo si presenta in questo modo:

```
function [ud_out,uq_out,i,fl,m,Em_n,Ef_n,E]=
fcn(id,iq,theta_me,w_me,L,Lmg,udc,p,R,Mn,Fn,Em,Ef,ud_in,uq_in,i_in)

%dalle ia,ib,ic ho bisogno di id e iq

%dalle correnti calcolo i flussi e la coppia
fld=L*id+Lmg;
flq=L*iq;
fl=sqrt((fld*fld)+(flq*flq));
m=(3/2)*p*Lmg*iq;

%calcolo in d-q i sei vettori possibili in uscita dall'inverter
ud0=0;
uq0=0;
u_alfa=0;
u_beta=0;

u_alfa1=(2/3)*udc;
u_beta1=0;
ud1=u_alfa1*cos(theta_me)+u_beta1*sin(theta_me);
uq1=-u_alfa1*sin(theta_me)+u_beta1*cos(theta_me);

u_alfa2=(2/3)*udc*cos(pi/3);
u_beta2=(2/3)*udc*sin(pi/3);
```

```
ud2=u_alfa2*cos(theta_me)+u_beta2*sin(theta_me);
uq2=-u_alfa2*sin(theta_me)+u_beta2*cos(theta_me);
```

```
u_alfa3=(2/3)*udc*cos(2*(pi/3));
u_beta3=(2/3)*udc*sin(2*(pi/3));
ud3=u_alfa3*cos(theta_me)+u_beta3*sin(theta_me);
uq3=-u_alfa3*sin(theta_me)+u_beta3*cos(theta_me);
```

```
u_alfa4=(2/3)*udc*cos(3*(pi/3));
u_beta4=(2/3)*udc*sin(3*(pi/3));
ud4=u_alfa4*cos(theta_me)+u_beta4*sin(theta_me);
uq4=-u_alfa4*sin(theta_me)+u_beta4*cos(theta_me);
```

```
u_alfa5=(2/3)*udc*cos(4*(pi/3));
u_beta5=(2/3)*udc*sin(4*(pi/3));
ud5=u_alfa5*cos(theta_me)+u_beta5*sin(theta_me);
uq5=-u_alfa5*sin(theta_me)+u_beta5*cos(theta_me);
```

```
u_alfa6=(2/3)*udc*cos(5*(pi/3));
u_beta6=(2/3)*udc*sin(5*(pi/3));
ud6=u_alfa6*cos(theta_me)+u_beta6*sin(theta_me);
uq6=-u_alfa6*sin(theta_me)+u_beta6*cos(theta_me);
```

```
%per ciascuno dei sei vettori di uscita dell'inverter calcolo la rispettiva
%derivata di coppia e flusso, utilizzando le formule ricavate per il sistema
%di riferimento d-q
```

```
dm0=((3/2)*p)*((Lmg*iq*(-R/L))+(uq0*(Lmg/L))+(w_me*Lmg*(-id-(Lmg/L))));
df0=(1/fl)*(ud0*fld)+(uq0*flq)-R*(fld*id+flq*iq);
```

```
dm1=((3/2)*p)*((Lmg*iq*(-R/L))+(uq1*(Lmg/L))+(w_me*Lmg*(-id-(Lmg/L))));
df1=(1/fl)*(ud1*fld)+(uq1*flq)-R*(fld*id+flq*iq);
```

```
dm2=((3/2)*p)*((Lmg*iq*(-R/L))+(uq2*(Lmg/L))+(w_me*Lmg*(-id-(Lmg/L))));
df2=(1/fl)*(ud2*fld)+(uq2*flq)-R*(fld*id+flq*iq);
```

```
dm3=((3/2)*p)*((Lmg*iq*(-R/L))+(uq3*(Lmg/L))+(w_me*Lmg*(-id-(Lmg/L))));
df3=(1/fl)*(ud3*fld)+(uq3*flq)-R*(fld*id+flq*iq);
```

```
dm4=((3/2)*p)*((Lmg*iq*(-R/L))+(uq4*(Lmg/L))+(w_me*Lmg*(-id-(Lmg/L))));
df4=(1/fl)*(ud4*fld)+(uq4*flq)-R*(fld*id+flq*iq);
```

```
dm5=((3/2)*p)*((Lmg*iq*(-R/L))+(uq5*(Lmg/L))+(w_me*Lmg*(-id-(Lmg/L))));
df5=(1/fl)*(ud5*fld)+(uq5*flq)-R*(fld*id+flq*iq);
```

```
dm6=((3/2)*p)*((Lmg*iq*(-R/L))+(uq6*(Lmg/L))+(w_me*Lmg*(-id-(Lmg/L))));
df6=(1/fl)*(ud6*fld)+(uq6*flq)-R*(fld*id+flq*iq);
```

```
%algoritmo di scelta dei vettori da applicare, inserendo un errore limite,
%normalizzando tutti i valori, usando grafo Matthias
```

```
a=1.8;
E_lim=5e-2;
```

```
der_m=[-dm0,-dm1,-dm2,-dm3,-dm4,-dm5,-dm6,0];
```

```

der_f=[-df0,-df1,-df2,-df3,-df4,-df5,-df6,0];

%normalizzazioni
der_m_n=(1/Mn)*der_m;
der_f_n=(a/Fn)*der_f;

Em_n=Em/Mn;
Ef_n=a*(Ef/Fn);

val_m=Em_n*der_m_n;
val_f=Ef_n*der_f_n;

val_m(8)=val_m(1);
val_f(8)=val_f(1);

c=zeros(1,8);

E=sqrt(Em_n^2+Ef_n^2);

ud_out=0;
uq_out=0;

SL_A=0;
SL_B=0;
SL_C=0;
% inizio valutazione errore
if E>E_lim;
    for i= 1:8
        %scelgo vettori furbi (vedi grafo Matthias)
        if (i_in==1 && (i==2 || i==4 || i==6 || i==1) ) || ...
            (i_in==2 && (i==1 || i==3 || i==7 || i==2) ) || ...
            (i_in==3 && (i==2 || i==4 || i==8 || i==3) ) || ...
            (i_in==4 && (i==1 || i==3 || i==5 || i==4) ) || ...
            (i_in==5 && (i==4 || i==6 || i==8 || i==5) ) || ...
            (i_in==6 && (i==1 || i==5 || i==7 || i==6) ) || ...
            (i_in==7 && (i==2 || i==6 || i==8 || i==7) ) || ...
            (i_in==8 && (i==3 || i==5 || i==7 || i==8) )
            c(i)=val_m(i)+val_f(i);
        else
            c(i)=1e9;
        end
    end
    [z,i]=min(c);
    k=i;

switch k
    case 1
        SL_A=0;
        SL_B=0;
        SL_C=0;
    case 2
        SL_A=1;
        SL_B=0;
        SL_C=0;
    case 3
        SL_A=1;
        SL_B=1;
        SL_C=0;
    case 4
        SL_A=0;

```

```

        SL_B=1;
        SL_C=0;
    case 5
        SL_A=0;
        SL_B=1;
        SL_C=1;
    case 6
        SL_A=0;
        SL_B=0;
        SL_C=1;
    case 7
        SL_A=1;
        SL_B=0;
        SL_C=1;
    case 8
        SL_A=1;
        SL_B=1;
        SL_C=1;
    otherwise
        -1;
end

u_alfa_beta=(2/3)*udc.*(SL_A*exp(1j*0)+SL_B*exp(1j*((2*pi)/3))+SL_C*exp(1j*((4*pi)/3)));
u_alfa=real(u_alfa_beta);
u_beta=imag(u_alfa_beta);

ud_out=u_alfa*cos(theta_me)+u_beta*sin(theta_me);
uq_out=-u_alfa*sin(theta_me)+u_beta*cos(theta_me);

else
    ud_out=ud_in;
    uq_out=uq_in;
    i=i_in;
end

%dalle ud e uq ho bisogno degli stati da applicare all'inverter

```

I parametri, eventualmente modificabili e diversi a seconda della macchina, sono tenuti come variabili di ingresso. Oltre a questi riceviamo le correnti e la posizione-velocità angolare del motore, sono queste infatti le grandezze che andiamo a trasdurre mediante opportune sonde di tensione ed encoder.

Calcoliamo dapprima le grandezze di controllo: coppia e flusso mediante le note formule teoriche, successivamente calcoliamo l'errore e lo normalizziamo. Se esce dalla fascia di tolleranza parte il controllo: calcola i sei vettori ottenibili dall'inverter, le rispettive derivate di coppia e di flusso, e li valuta usando la solita funzione costo attuando gli stati che minimizzano e comportano le minime commutazioni.

I grafici sono del tutto equivalenti a quelli analizzati in precedenza nel capitolo cinque, è sempre previsto il controllo di velocità, una banda di isteresi e la riduzione della frequenza di switching.

6.3-Controllo predittivo

Il passo di preparazione successivo consiste nella seguente osservazione: noi ora stiamo simulando lo schema che rappresenta il sistema con un certo passo fisso impostato, equivale a dire che l’algoritmo calcola le grandezze di controllo mentre tutto il sistema è fermo. Nella realtà le cose sono molto diverse: il microprocessore esegue i calcoli con una certa cadenza temporale (ammesso che riesca ad eseguirli tutti) mentre il motore non si blocca, segue la sua evoluzione dettata dalle tensioni attuate in quell’istante.

Per simulare e risolvere questo problema possiamo discretizzare l’algoritmo di comando imponendo un ritardo di calcolo nell’attuazione delle tensioni, in tal modo si applica la grandezza di controllo appena calcolata nel passo successivo, così facendo il motore segue la sua evoluzione con le tensioni imposte in precedenza.

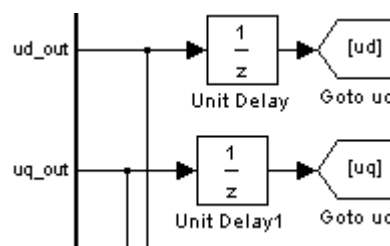


Fig. 6.2 blocchi ritardo

Per rispondere meglio alla problematica andiamo ad implementare un controllo leggermente differente detto controllo predittivo.

L’idea nasce dal fatto che nella realtà la macchina è interessata da certe grandezze (correnti e tensioni) mentre l’algoritmo di comando esegue i suoi calcoli. Con il predittivo possiamo, adottando certe approssimazioni, ricavare proprio quelle grandezze future che interesseranno il motore mentre il microprocessore esegue l’elaborazione numerica, “prevedendo” in tal modo le correnti e le tensioni che interesseranno il nostro sistema.

La filosofia portante dell’algoritmo non cambia: cerchiamo sempre una funzione di minimo costo, ora però andiamo a valutare le possibili correnti future ottenibili da relativi vettori spaziali dell’inverter. Di queste correnti future ricaviamo i corrispondenti valori di coppia e flusso futuri, scegliendo poi quelli che minimizzeranno l’errore futuro.

Andiamo allora a determinare questi valori futuri sulla base dei seguenti ragionamenti

Dalle equazioni portanti del motore BL SPM:

$$\begin{cases} u_d = Ri_d + L \frac{di_d}{dt} - \omega_{me} Li_q \\ u_q = Ri_q + L \frac{di_q}{dt} + \omega_{me} (Li_d + \Lambda_{mg}) \end{cases}$$

Cerchiamo l’espressione della derivata della corrente:

$$\begin{cases} \frac{di_d}{dt} = \frac{u_d}{L} - \frac{R}{L}i_d + \omega_{me}i_q \\ \frac{di_q}{dt} = \frac{u_q}{L} - \frac{R}{L}i_q - \frac{\omega_{me}}{L}(Li_d + \Lambda_{mg}) \end{cases}$$

Dalla teoria matematica sappiamo che la derivata di una funzione calcolata su un intervallo sufficientemente piccolo diventa:

$$\frac{df(x)}{dx} \cong \frac{f(x + \Delta) - f(x)}{\Delta}$$

Se consideriamo l'intervallo Δ come lo step temporale T_s tra due successivi istanti di calcolo t e $t + T_s$ possiamo scrivere:

$$\frac{df(t)}{dt} \cong \frac{f(t + T_s) - f(t)}{T_s}$$

Quindi per le nostre correnti, partendo per esempio da i_d :

$$\frac{di_d}{dt} = \frac{i_d(t + T_s) - i_d(t)}{T_s} \Rightarrow i_d(t + T_s) = T_s \frac{di_d}{dt} + i_d(t)$$

Pensiamo quindi che date le correnti misurate all'istante k-esimo, possiamo prevedere il valore che avranno le stesse dopo un passo di calcolo T_s , e quindi all'istante k+1.:

$$\begin{aligned} i_d(k + 1) &= T_s \frac{di_d}{dt} + i_d(k) = T_s \left(\frac{u_d}{L} - \frac{R}{L}i_d + \omega_{me}i_q \right) + i_d(k) \\ &= \frac{T_s u_d(k)}{L} - \frac{R T_s i_d(k)}{L} + i_d(k) + T_s \omega_{me} i_q \\ &= \left(1 - \frac{R T_s}{L} \right) i_d(k) + \frac{T_s u_d(k)}{L} + T_s \omega_{me} i_q \end{aligned}$$

Analogamente per la corrente i_q :

$$\begin{aligned} i_q(k + 1) &= T_s \frac{di_q}{dt} + i_q(k) = T_s \left(\frac{u_q}{L} - \frac{R}{L}i_q + \frac{\omega_{me}}{L}(Li_d + \Lambda_{mg}) \right) + i_q(k) \\ &= \frac{T_s u_q(k)}{L} - \frac{R T_s i_q(k)}{L} + i_q(k) + \omega_{me} \left(i_d(k) + \frac{\Lambda_{mg}}{L} \right) \\ &= \left(1 - \frac{R T_s}{L} \right) i_q(k) + \frac{T_s u_q(k)}{L} + \omega_{me} \left(i_d(k) + \frac{\Lambda_{mg}}{L} \right) \end{aligned}$$

Siamo quindi in grado di calcolare le correnti future sotto opportune ipotesi, a partire dalle grandezze che interessano la macchina nell'istante presente. Dalle correnti passiamo ai flussi e la coppia che sono le grandezze che ai noi interessa controllare.

Possiamo adesso applicare un criterio di controllo simile al precedente ovvero: implementiamo la formule trovate per ciascuno dei sei vettori spaziali di tensione in uscita dall'inverter valutando il vettore che presenta l'errore di coppia e flusso futuro minimo, utilizzando sempre i precedenti accorgimenti per affinare il controllo.

L'algoritmo diventa:

```
function [ud_out,uq_out,i,Em_n,Ef_n] =
fcn(id,iq,theta_me,m_rif,fl_rif,R,L,Lmg,p,w_me,udc,Mn,Fn,ud_in,uq_in,i_in)

Ts=1e-5;

%rozzi stimatori
fld=L*id+Lmg;
flq=L*iq;
fl=sqrt((fld*fld)+(flq*flq));
m=(3/2)*p*Lmg*iq;

%valuto errore
Em=m_rif-m;
Ef=fl_rif-fl;
a=1.8;

Em_n=Em/Mn;
Ef_n=a*(Ef/Fn);

E=sqrt(Em_n^2+Ef_n^2);

E_lim=5e-2;

ud_out=0;
uq_out=0;

%id_post=0;
%iq_post=0;

% inizio valutazione errore
if E>E_lim;
%calcolo in d-q i sei vettori possibili in uscita dall'inverter
ud0=0;
uq0=0;
u_alfa=0;
u_beta=0;

u_alfa1=(2/3)*udc;
u_beta1=0;
ud1=u_alfa1*cos(theta_me)+u_beta1*sin(theta_me);
uq1=-u_alfa1*sin(theta_me)+u_beta1*cos(theta_me);

u_alfa2=(2/3)*udc*cos(pi/3);
u_beta2=(2/3)*udc*sin(pi/3);
ud2=u_alfa2*cos(theta_me)+u_beta2*sin(theta_me);
```

```
uq2=-u_alfa2*sin(theta_me)+u_beta2*cos(theta_me);
```

```
u_alfa3=(2/3)*udc*cos(2*(pi/3));  
u_beta3=(2/3)*udc*sin(2*(pi/3));  
ud3=u_alfa3*cos(theta_me)+u_beta3*sin(theta_me);  
uq3=-u_alfa3*sin(theta_me)+u_beta3*cos(theta_me);
```

```
u_alfa4=(2/3)*udc*cos(3*(pi/3));  
u_beta4=(2/3)*udc*sin(3*(pi/3));  
ud4=u_alfa4*cos(theta_me)+u_beta4*sin(theta_me);  
uq4=-u_alfa4*sin(theta_me)+u_beta4*cos(theta_me);
```

```
u_alfa5=(2/3)*udc*cos(4*(pi/3));  
u_beta5=(2/3)*udc*sin(4*(pi/3));  
ud5=u_alfa5*cos(theta_me)+u_beta5*sin(theta_me);  
uq5=-u_alfa5*sin(theta_me)+u_beta5*cos(theta_me);
```

```
u_alfa6=(2/3)*udc*cos(5*(pi/3));  
u_beta6=(2/3)*udc*sin(5*(pi/3));  
ud6=u_alfa6*cos(theta_me)+u_beta6*sin(theta_me);  
uq6=-u_alfa6*sin(theta_me)+u_beta6*cos(theta_me);
```

```
%per ciascuno dei sei vettori di uscita dell'inverter, calcolo le correnti  
%future e di conseguenza i flussi e la coppia futura utilizzando le formule  
%ricavate per il sistema di riferimento d-q
```

```
id_post_0=(1-((R*Ts)/L))*id+(Ts/L)*ud0+w_me*L*Ts*iq;  
iq_post_0=(1-((R*Ts)/L))*iq+(Ts/L)*uq0-w_me*Ts*(id+(Lmg/L));  
fld_post_0=L*id_post_0+Lmg;  
flq_post_0=L*iq_post_0;  
fl_post_0=sqrt((fld_post_0*fld_post_0)+(flq_post_0*flq_post_0));  
m_post_0=(3/2)*p*Lmg*iq_post_0;
```

```
id_post_1=(1-((R*Ts)/L))*id+(Ts/L)*ud1+w_me*L*Ts*iq;  
iq_post_1=(1-((R*Ts)/L))*iq+(Ts/L)*uq1-w_me*Ts*(id+(Lmg/L));  
fld_post_1=L*id_post_1+Lmg;  
flq_post_1=L*iq_post_1;  
fl_post_1=sqrt((fld_post_1*fld_post_1)+(flq_post_1*flq_post_1));  
m_post_1=(3/2)*p*Lmg*iq_post_1;
```

```
id_post_2=(1-((R*Ts)/L))*id+(Ts/L)*ud2+w_me*L*Ts*iq;  
iq_post_2=(1-((R*Ts)/L))*iq+(Ts/L)*uq2-w_me*Ts*(id+(Lmg/L));  
fld_post_2=L*id_post_2+Lmg;  
flq_post_2=L*iq_post_2;  
fl_post_2=sqrt((fld_post_2*fld_post_2)+(flq_post_2*flq_post_2));  
m_post_2=(3/2)*p*Lmg*iq_post_2;
```

```
id_post_3=(1-((R*Ts)/L))*id+(Ts/L)*ud3+w_me*L*Ts*iq;  
iq_post_3=(1-((R*Ts)/L))*iq+(Ts/L)*uq3-w_me*Ts*(id+(Lmg/L));  
fld_post_3=L*id_post_3+Lmg;  
flq_post_3=L*iq_post_3;  
fl_post_3=sqrt((fld_post_3*fld_post_3)+(flq_post_3*flq_post_3));  
m_post_3=(3/2)*p*Lmg*iq_post_3;
```

```
id_post_4=(1-((R*Ts)/L))*id+(Ts/L)*ud4+w_me*L*Ts*iq;  
iq_post_4=(1-((R*Ts)/L))*iq+(Ts/L)*uq4-w_me*Ts*(id+(Lmg/L));  
fld_post_4=L*id_post_4+Lmg;  
flq_post_4=L*iq_post_4;
```

```

fl_post_4=sqrt((fld_post_4*fld_post_4)+(flq_post_4*flq_post_4));
m_post_4=(3/2)*p*Lmg*iq_post_4;

id_post_5=(1-((R*Ts)/L))*id+(Ts/L)*ud5+w_me*L*Ts*iq;
iq_post_5=(1-((R*Ts)/L))*iq+(Ts/L)*uq5-w_me*Ts*(id+(Lmg/L));
fld_post_5=L*id_post_5+Lmg;
flq_post_5=L*iq_post_5;
fl_post_5=sqrt((fld_post_5*fld_post_5)+(flq_post_5*flq_post_5));
m_post_5=(3/2)*p*Lmg*iq_post_5;

id_post_6=(1-((R*Ts)/L))*id+(Ts/L)*ud6+w_me*L*Ts*iq;
iq_post_6=(1-((R*Ts)/L))*iq+(Ts/L)*uq6-w_me*Ts*(id+(Lmg/L));
fld_post_6=L*id_post_6+Lmg;
flq_post_6=L*iq_post_6;
fl_post_6=sqrt((fld_post_6*fld_post_6)+(flq_post_6*flq_post_6));
m_post_6=(3/2)*p*Lmg*iq_post_6;

%algoritmo di scelta dei vettori da applicare, inserendo un errore limite,
%normalizzando tutti i valori, usando grafo Matthias

a=1.8;
E_lim=5e-2;

post_m=[-m_post_0,-m_post_1,-m_post_2,-m_post_3,-m_post_4,-m_post_5,-
m_post_6,0];
post_f=[-fl_post_0,-fl_post_1,-fl_post_2,-fl_post_3,-fl_post_4,-fl_post_5,-
fl_post_6,0];

%normalizzazioni
post_m_n=(1/Mn)*post_m;
post_f_n=(a/Fn)*post_f;

val_m=Em_n*post_m_n;
val_f=Ef_n*post_f_n;

val_m(8)=val_m(1);
val_f(8)=val_f(1);

c=zeros(1,8);

for i= 1:8
    %scelgo vettori furbi (vedi grafo Matthias)
    if (i_in==1 && (i==2 || i==4 || i==6 || i==1) ) || ...
        (i_in==2 && (i==1 || i==3 || i==7 || i==2) ) || ...
        (i_in==3 && (i==2 || i==4 || i==8 || i==3) ) || ...
        (i_in==4 && (i==1 || i==3 || i==5 || i==4) ) || ...
        (i_in==5 && (i==4 || i==6 || i==8 || i==5) ) || ...
        (i_in==6 && (i==1 || i==5 || i==7 || i==6) ) || ...
        (i_in==7 && (i==2 || i==6 || i==8 || i==7) ) || ...
        (i_in==8 && (i==3 || i==5 || i==7 || i==8) )
        c(i)=val_m(i)+val_f(i);
    else
        c(i)=1e9;
    end
    [z,i]=min(c);
    k=i;
    u_alfa=0;
    u_beta=0;

```

```

switch k
  case 1
    u_alfa=0;
    u_beta=0;
  case 2
    u_alfa=(2/3)*udc;
    u_beta=0;
  case 3
    u_alfa=(2/3)*udc*cos(pi/3);
    u_beta=(2/3)*udc*sin(pi/3);
  case 4
    u_alfa=(2/3)*udc*cos((2*pi)/3);
    u_beta=(2/3)*udc*sin((2*pi)/3);
  case 5
    u_alfa=(2/3)*udc*cos((3*pi)/3);
    u_beta=(2/3)*udc*sin((3*pi)/3);
  case 6
    u_alfa=(2/3)*udc*cos((4*pi)/3);
    u_beta=(2/3)*udc*sin((4*pi)/3);
  case 7
    u_alfa=(2/3)*udc*cos((5*pi)/3);
    u_beta=(2/3)*udc*sin((5*pi)/3);
  case 8
    u_alfa=0;
    u_beta=0;
  otherwise
    -1;
end
ud_out=u_alfa*cos(theta_me)+u_beta*sin(theta_me);
uq_out=-u_alfa*sin(theta_me)+u_beta*cos(theta_me);

end
else
  ud_out=ud_in;
  uq_out=uq_in;
  i=i_in;
end

```

Cambiano allora le formule che valutano le grandezze future per ciascun vettore di tensione applicabile dall' inverter e i nomi delle variabili da memorizzare negli array. Otteniamo in fase di simulazione i seguenti andamenti di coppia e flusso (discretizzazione $1e^{-4}$):

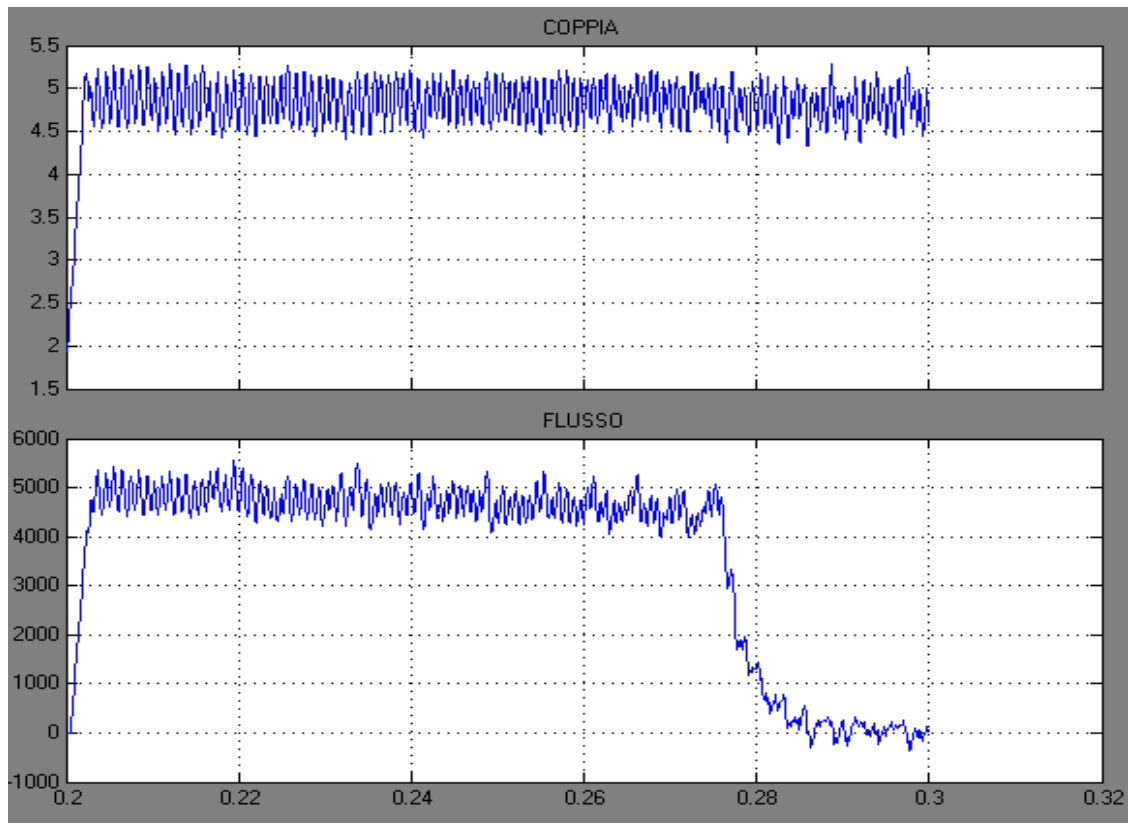


Grafico 6.1 Coppia-Flusso con controllo predittivo discreto

Il grafico dell'errore peggiora notevolmente:

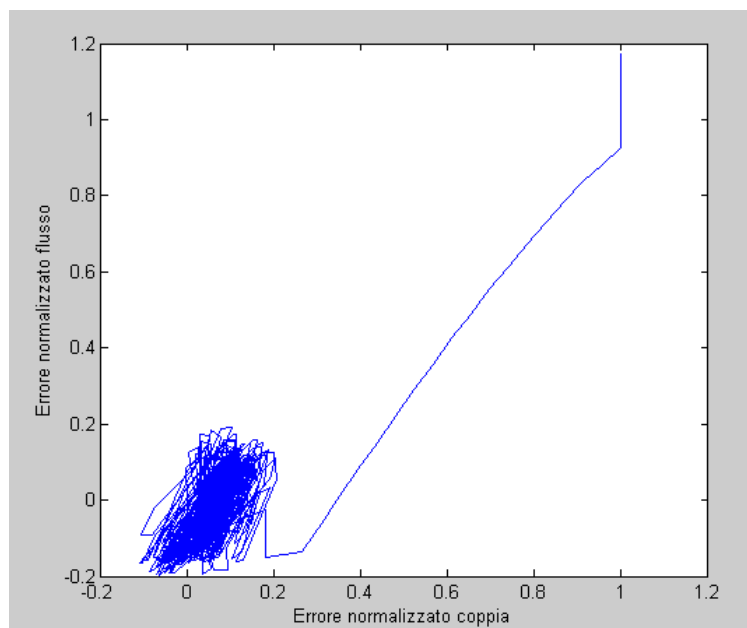


Grafico 6.2 rappresentazione sul piano complesso degli errori coppia flusso con controllo predittivo discretizzato

6.4-Passaggio da SPM a IPM

Dobbiamo adattare adesso il nostro controllo pensato per un motore SPM all' IPM perché la sperimentazione viene fatta appunto su una macchina anisotropa.

Possiamo allora seguire due strade: o riscrivere le equazioni di derivata di coppia e flusso oppure adattare il controllo predittivo semplicemente considerando che le induttanze sincrone sono ora due e diverse l'una dall'altra.

Le particolarizziamo per un motore IPM, che sarà quello che andremo ad analizzare :

$$\begin{cases} u_d = R i_d + L_d \frac{di_d}{dt} - \omega_{me} L_q i_q \\ u_q = R i_q + L_q \frac{di_q}{dt} + \omega_{me} (L_d i_d + \Lambda_{mg}) \end{cases}$$

Cerchiamo l'espressione della derivata della corrente:

$$\begin{cases} \frac{di_d}{dt} = \frac{u_d}{L_d} - \frac{R}{L_d} i_d + \omega_{me} \frac{L_q}{L_d} i_q \\ \frac{di_q}{dt} = \frac{u_q}{L_q} - \frac{R}{L_q} i_q - \frac{\omega_{me}}{L_q} (L_d i_d + \Lambda_{mg}) \end{cases}$$

Quindi per le nostre correnti, partendo per esempio da i_d :

$$\frac{di_d}{dt} = \frac{i_d(t + Ts) - i_d(t)}{Ts} \Rightarrow i_d(t + Ts) = Ts \frac{di_d}{dt} + i_d(t)$$

Pensiamo quindi che date le correnti misurate all'istante k-esimo, possiamo prevedere il valore che avranno le stesse dopo un passo di calcolo T_s , e quindi all'istante k+1.:

$$\begin{aligned} i_d(k+1) &= Ts \frac{di_d}{dt} + i_d(k) = Ts \left(\frac{u_d}{L_d} - \frac{R}{L_d} i_d + \omega_{me} \frac{L_q}{L_d} i_q \right) + i_d(k) \\ &= \frac{Ts u_d(k)}{L_d} - \frac{R Ts i_d(k)}{L_d} + i_d(k) + \frac{Ts \omega_{me} L_q i_q(k)}{L_d} \\ &= \left(1 - \frac{R Ts}{L_d} \right) i_d(k) + \frac{Ts u_d(k)}{L_d} + \frac{Ts \omega_{me} L_q i_q(k)}{L_d} \end{aligned}$$

Analogamente per la corrente i_q :

$$\begin{aligned}
i_q(k+1) &= Ts \frac{di_q}{dt} + i_q(k) = Ts \left(\frac{u_q}{L_q} - \frac{R}{L_q} i_q - \frac{\omega_{me}}{L_q} (L_d i_d + \Lambda_{mg}) \right) + i_q(k) \\
&= \frac{Ts u_q(k)}{L_q} - \frac{R Ts i_q(k)}{L_q} + i_q(k) - \frac{\omega_{me} L_d}{L_q} \left(i_d(k) + \frac{\Lambda_{mg}}{L_d} \right) \\
&= \left(1 - \frac{R Ts}{L_q} \right) i_q(k) + \frac{Ts u_q(k)}{L_q} - \frac{\omega_{me} L_d}{L_q} \left(i_d(k) + \frac{\Lambda_{mg}}{L_d} \right)
\end{aligned}$$

Basta quindi cambiare i pochi termini relativi alle induttanze nelle varie equazioni per passare da un modello all'altro e l'algoritmo cambia solo nella parte in cui si definiscono i valori futuri per ciascun vettore di tensione:

$$m = \frac{3}{2} p (\lambda_d i_q - \lambda_q i_d) = \frac{3}{2} p (\Lambda_{mg} + \Delta L i_{dn}) i_{qn}$$

```

function [ud_out,uq_out,i] =
fcn(id,iq,theta_me,m_rif,fl_rif,R,Ld,Lq,Lmg,p,w_me,udc,Mn,Fn,ud_in,uq_in,i_in)

Ts=1e-5;

ud_out=0;
uq_out=0;

%id_post=0;
%iq_post=0;

%calcolo in d-q i sei vettori possibili in uscita dall'inverter
ud0=0;
uq0=0;
u_alfa=0;
u_beta=0;

u_alfa1=(2/3)*udc;
u_beta1=0;
ud1=u_alfa1*cos(theta_me)+u_beta1*sin(theta_me);
uq1=-u_alfa1*sin(theta_me)+u_beta1*cos(theta_me);

u_alfa2=(2/3)*udc*cos(pi/3);
u_beta2=(2/3)*udc*sin(pi/3);
ud2=u_alfa2*cos(theta_me)+u_beta2*sin(theta_me);
uq2=-u_alfa2*sin(theta_me)+u_beta2*cos(theta_me);

u_alfa3=(2/3)*udc*cos(2*(pi/3));
u_beta3=(2/3)*udc*sin(2*(pi/3));
ud3=u_alfa3*cos(theta_me)+u_beta3*sin(theta_me);
uq3=-u_alfa3*sin(theta_me)+u_beta3*cos(theta_me);

u_alfa4=(2/3)*udc*cos(3*(pi/3));
u_beta4=(2/3)*udc*sin(3*(pi/3));
ud4=u_alfa4*cos(theta_me)+u_beta4*sin(theta_me);
uq4=-u_alfa4*sin(theta_me)+u_beta4*cos(theta_me);

```



```

u_alfa5=(2/3)*udc*cos(4*(pi/3));
u_beta5=(2/3)*udc*sin(4*(pi/3));
ud5=u_alfa5*cos(theta_me)+u_beta5*sin(theta_me);
uq5=-u_alfa5*sin(theta_me)+u_beta5*cos(theta_me);

```

```

u_alfa6=(2/3)*udc*cos(5*(pi/3));
u_beta6=(2/3)*udc*sin(5*(pi/3));
ud6=u_alfa6*cos(theta_me)+u_beta6*sin(theta_me);
uq6=-u_alfa6*sin(theta_me)+u_beta6*cos(theta_me);

```

```

%per ciascuno dei sei vettori di uscita dell'inverter, calcolo le correnti
%future e di conseguenza i flussi e la coppia futura utilizzando le formule
%ricavate per il sistema di riferimento d-q

```

```

id_post_0=(1-((R*T_s)/L_d))*id+(T_s/L_d)*ud0+w_me*(L_q/L_d)*T_s*iq;
iq_post_0=(1-((R*T_s)/L_q))*iq+(T_s/L_q)*uq0-w_me*(L_d/L_q)*T_s*(id+(L_mg/L_d));
fld_post_0=L_d*id_post_0+L_mg;
flq_post_0=L_q*iq_post_0;
fl_post_0=sqrt((fld_post_0*fld_post_0)+(flq_post_0*flq_post_0));
m_post_0=(3/2)*p*(L_mg+(L_d-L_q)*id_post_0)*iq_post_0;

```

```

id_post_1=(1-((R*T_s)/L_d))*id+(T_s/L_d)*ud1+w_me*(L_q/L_d)*T_s*iq;
iq_post_1=(1-((R*T_s)/L_q))*iq+(T_s/L_q)*uq1-w_me*(L_d/L_q)*T_s*(id+(L_mg/L_d));
fld_post_1=L_d*id_post_1+L_mg;
flq_post_1=L_q*iq_post_1;
fl_post_1=sqrt((fld_post_1*fld_post_1)+(flq_post_1*flq_post_1));
m_post_1=(3/2)*p*(L_mg+(L_d-L_q)*id_post_1)*iq_post_1;

```

```

id_post_2=(1-((R*T_s)/L_d))*id+(T_s/L_d)*ud2+w_me*(L_q/L_d)*T_s*iq;
iq_post_2=(1-((R*T_s)/L_q))*iq+(T_s/L_q)*uq2-w_me*(L_d/L_q)*T_s*(id+(L_mg/L_d));
fld_post_2=L_d*id_post_2+L_mg;
flq_post_2=L_q*iq_post_2;
fl_post_2=sqrt((fld_post_2*fld_post_2)+(flq_post_2*flq_post_2));
m_post_2=(3/2)*p*(L_mg+(L_d-L_q)*id_post_2)*iq_post_2;

```

```

id_post_3=(1-((R*T_s)/L_d))*id+(T_s/L_d)*ud3+w_me*(L_q/L_d)*T_s*iq;
iq_post_3=(1-((R*T_s)/L_q))*iq+(T_s/L_q)*uq3-w_me*(L_d/L_q)*T_s*(id+(L_mg/L_d));
fld_post_3=L_d*id_post_3+L_mg;
flq_post_3=L_q*iq_post_3;
fl_post_3=sqrt((fld_post_3*fld_post_3)+(flq_post_3*flq_post_3));
m_post_3=(3/2)*p*(L_mg+(L_d-L_q)*id_post_3)*iq_post_3;

```

```

id_post_4=(1-((R*T_s)/L_d))*id+(T_s/L_d)*ud4+w_me*(L_q/L_d)*T_s*iq;
iq_post_4=(1-((R*T_s)/L_q))*iq+(T_s/L_q)*uq4-w_me*(L_d/L_q)*T_s*(id+(L_mg/L_d));
fld_post_4=L_d*id_post_4+L_mg;
flq_post_4=L_q*iq_post_4;
fl_post_4=sqrt((fld_post_4*fld_post_4)+(flq_post_4*flq_post_4));
m_post_4=(3/2)*p*(L_mg+(L_d-L_q)*id_post_4)*iq_post_4;

```

```

id_post_5=(1-((R*T_s)/L_d))*id+(T_s/L_d)*ud5+w_me*(L_q/L_d)*T_s*iq;
iq_post_5=(1-((R*T_s)/L_q))*iq+(T_s/L_q)*uq5-w_me*(L_d/L_q)*T_s*(id+(L_mg/L_d));
fld_post_5=L_d*id_post_5+L_mg;
flq_post_5=L_q*iq_post_5;
fl_post_5=sqrt((fld_post_5*fld_post_5)+(flq_post_5*flq_post_5));
m_post_5=(3/2)*p*(L_mg+(L_d-L_q)*id_post_5)*iq_post_5;

```

```

id_post_6=(1-((R*T_s)/L_d))*id+(T_s/L_d)*ud6+w_me*(L_q/L_d)*T_s*iq;
iq_post_6=(1-((R*T_s)/L_q))*iq+(T_s/L_q)*uq6-w_me*(L_d/L_q)*T_s*(id+(L_mg/L_d));
fld_post_6=L_d*id_post_6+L_mg;

```

```

flq_post_6=Lq*iq_post_6;
fl_post_6=sqrt((fld_post_6*fld_post_6)+(flq_post_6*flq_post_6));
m_post_6=(3/2)*p*(Lmg+(Ld-Lq)*id_post_6)*iq_post_6;

%algoritmo di scelta dei vettori da applicare, inserendo un errore limite,
%normalizzando tutti i valori, usando grafo Matthias

post_m=[-m_post_0,-m_post_1,-m_post_2,-m_post_3,-m_post_4,-m_post_5,-
m_post_6,0];
post_f=[-fl_post_0,-fl_post_1,-fl_post_2,-fl_post_3,-fl_post_4,-fl_post_5,-
fl_post_6,0];

Em_post=m_rif+post_m;
Ef_post=fl_rif+post_f;

val_m=Em_post;
val_f=1e2*Ef_post;

val_m(8)=val_m(1);
val_f(8)=val_f(1);

c=zeros(1,8);

for i= 1:8
    %scelgo vettori furbi (vedi grafo Matthias)
    if (i_in==1 && (i==2 || i==4 || i==6 || i==1) ) || ...
        (i_in==2 && (i==1 || i==3 || i==7 || i==2) ) || ...
        (i_in==3 && (i==2 || i==4 || i==8 || i==3) ) || ...
        (i_in==4 && (i==1 || i==3 || i==5 || i==4) ) || ...
        (i_in==5 && (i==4 || i==6 || i==8 || i==5) ) || ...
        (i_in==6 && (i==1 || i==5 || i==7 || i==6) ) || ...
        (i_in==7 && (i==2 || i==6 || i==8 || i==7) ) || ...
        (i_in==8 && (i==3 || i==5 || i==7 || i==8) )
        c(i)=(val_m(i)^2)+(val_f(i)^2);
    else
        c(i)=1e9;
    end
end
[z,i]=min(c);
k=i;
u_alfa=0;
u_beta=0;
switch k
    case 1
        u_alfa=0;
        u_beta=0;
    case 2
        u_alfa=(2/3)*udc;
        u_beta=0;
    case 3
        u_alfa=(2/3)*udc*cos(pi/3);
        u_beta=(2/3)*udc*sin(pi/3);
    case 4
        u_alfa=(2/3)*udc*cos((2*pi)/3);
        u_beta=(2/3)*udc*sin((2*pi)/3);
    case 5

```

```

        u_alfa=(2/3)*udc*cos((3*pi)/3);
        u_beta=(2/3)*udc*sin((3*pi)/3);
    case 6
        u_alfa=(2/3)*udc*cos((4*pi)/3);
        u_beta=(2/3)*udc*sin((4*pi)/3);
    case 7
        u_alfa=(2/3)*udc*cos((5*pi)/3);
        u_beta=(2/3)*udc*sin((5*pi)/3);
    case 8
        u_alfa=0;
        u_beta=0;
    otherwise
        -1;
end
ud_out=u_alfa*cos(theta_me)+u_beta*sin(theta_me);
uq_out=-u_alfa*sin(theta_me)+u_beta*cos(theta_me);

end

```

6.5-Implementazione in C

Siamo ora pronti per scrivere l'algoritmo di controllo nel linguaggio C, precisiamo fin d'ora che il listato complessivo è composto da più parti, a cui hanno lavorato altri operatori per implementare l'acquisizione delle correnti, l'interfaccia di comando, le sicurezze per non danneggiare i componenti e d altre modalità di controllo.

Nel nostro caso andiamo ad operare nella parte mpc.c che implementa il controllo predittivo delle correnti e scriviamo la nostra formulazione ai flussi e coppia minimizzandone l'errore futuro.

Sostanzialmente la sintassi del linguaggio non presenta particolari differenze, dobbiamo però considerare che C è di livello più basso quindi alcune funzioni comode come per esempio il calcolo del valore minimo di un array non è previsto, bisognerà quindi pensare ad opportuni cicli che esaminano n valori e ne valutano il più piccolo. Altra differenza fondamentale poi sta nella definizione delle variabili e la loro memorizzazione per indice o per valore.

Passando alla programmazione vera e propria possiamo quindi passare al listato mpc.c comprendente il controllo predittivo dopo aver definito con degli struct due variabili relative alla macchina e al controllo andiamop a valutare la frequenza di switching e i valori futuri dei parametri. Implementiamo poi la funzione che calcola il costo minimo futuro che poi verrà richiamata dalla funzione principale predittiva.

Per prendere confidenza con l'ambiente di simulazione, verificare che l'algoritmo di controllo venga compilato e funzioni correttamente, facciamo una prima sperimentazione con motore a vuoto trattato in qualche modo come un SPM in tenendo a zero la corrente di asse diretto.

Così facendo, se incontriamo eventuali problematiche possiamo ragionare in maniera più semplice anche analiticamente

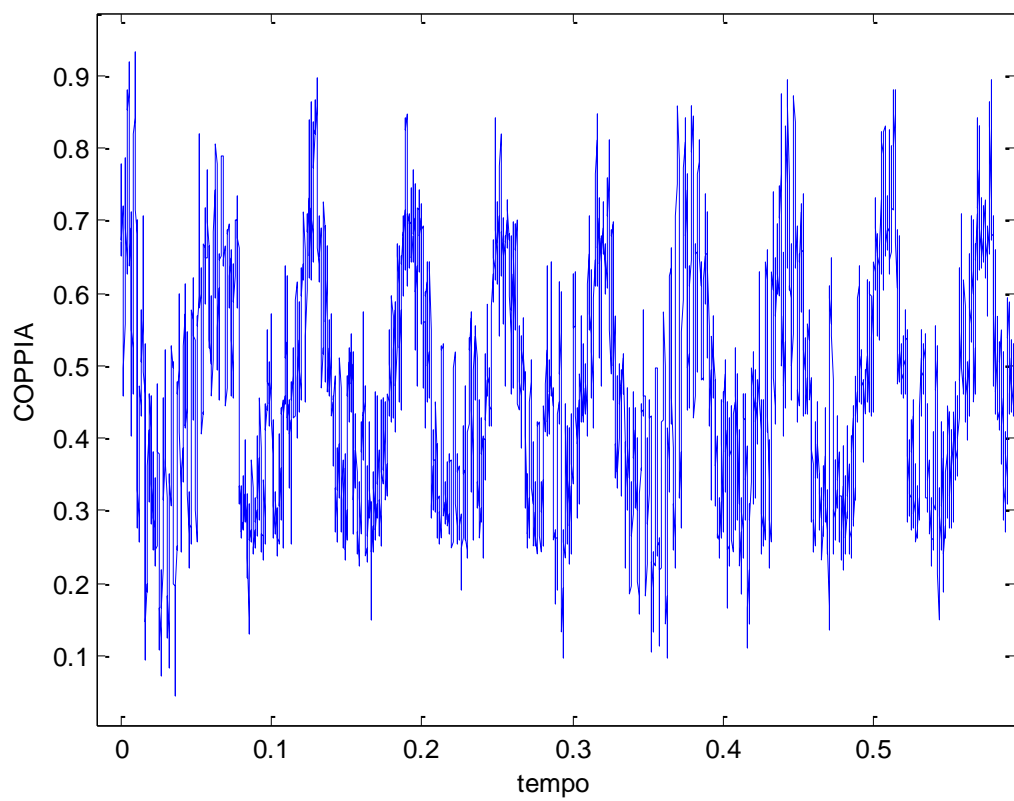


Grafico 6.3 Coppia reale con motore a vuoto e $i_d=0$ e 150 rpm

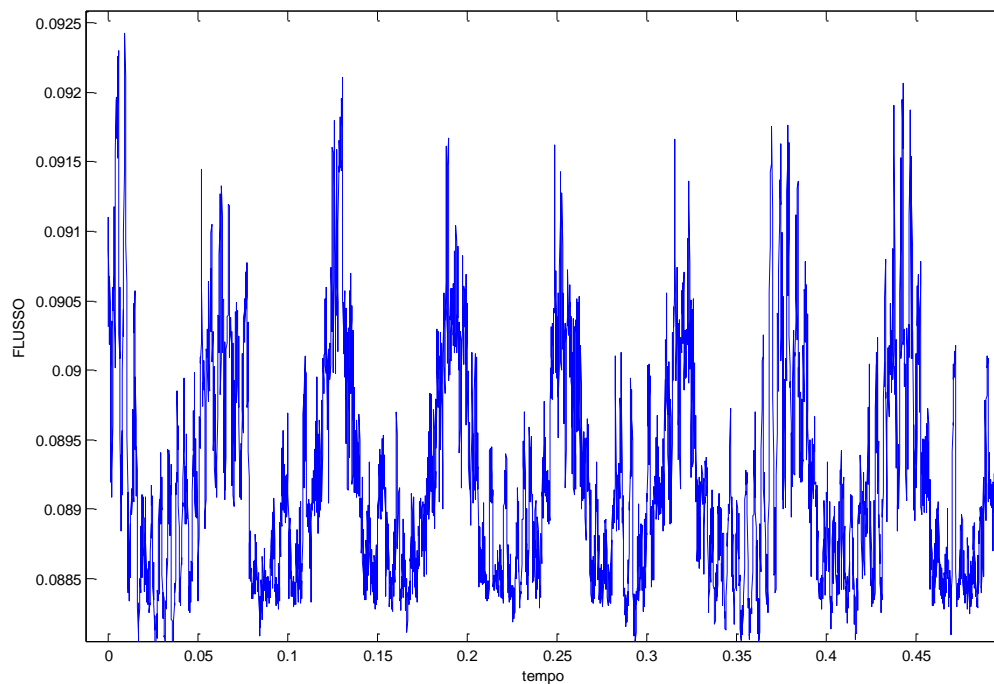


Grafico 6.4 Flusso reale con motore a vuoto e $i_d=0$ e 150 rpm

Analizziamo ora le problematiche relative alla sperimentazione.

In primo luogo a basse velocità il movimento del rotore non è fluido a causa della precisione finita dell'encoder che non è in grado di valutare con continuità la posizione angolare.

Altra difficoltà su cui però possiamo intervenire è la stima del flusso di riferimento. Se per un primo approccio si è preso un valore ottenuto imponendo a 0 la corrente i_d . Possiamo adesso affinare il controllo individuando un criterio che ci permette di imporre alla macchina un flusso di riferimento (costante) più consono.

Il filo conduttore del nostro ragionamento è stato il seguente: noi vogliamo lavorare nel punto di lavoro nominale della macchina ovvero se imponiamo la corrente nominale (nota) dobbiamo riuscire ad individuare un criterio che stando sulla curva MTPA individua le componenti d-q che danno la massima coppia ottenibile, quella nominale appunto.

Quindi dalla i_n vado a calcolare i_d i_q oppure con una formulazione più comoda l'angolo δ_{opt} ottimale che mi da la coppia nominale M_n .

$$i_n \rightarrow \delta_{opt} = f(i_n) \rightarrow T_n = f(\delta_{opt}, i_n)$$

Partiamo allora dalla corrente vista in forma polare:

$$i_n = |I| \langle \delta_{opt}$$

Le cui componenti sono:

$$i_d = |I| \sin \delta_{opt}$$

$$i_q = |I| \cos \delta_{opt}$$

Dobbiamo rispettare la condizione di Max Torque Per Ampere, vincoliamo le componenti a stare sulla curva descritta dalla seguente equazione (ponendo $\Delta L = (L_d - L_q)$):

$$-\Lambda_{mg}|I| \sin \delta_{opt} + \Delta L(|I|^2 \sin^2 \delta_{opt} - |I|^2 \cos^2 \delta_{opt}) = 0$$

Dalla trigonometria sappiamo che:

$$\cos^2 \delta_{opt} = 1 - \sin^2 \delta_{opt}$$

Quindi:

$$\Lambda_{mg}|I| \sin \delta_{opt} + \Delta L |I|^2 (2 \sin^2 \delta_{opt} - 1) = \Lambda_{mg} \sin \delta_{opt} + \Delta L |I| (2 \sin^2 \delta_{opt} - 1) = 0$$

Ne risulta una equazione trigonometrica di secondo grado del tipo:

$$2 \Delta L |I| \sin^2 \delta_{opt} + \Lambda_{mg} \sin \delta_{opt} - \Delta L |I| = 0$$

Le cui soluzioni sono:

$$\sin \delta_{opt} = \frac{-\Lambda_{mg} \pm \sqrt{\Lambda_{mg}^2 + 8 \Delta L |I|^2}}{4 \Delta L |I|}$$

Da cui ricaviamo, se serve (perché solitamente bastano seno e coseno), δ_{opt} ovvero l'angolo ottimo che deve formare la corrente per stare su MTPA.

Possiamo ora calcolare la coppia nominale:

$$M_n = \frac{3}{2} p |I| \cos \delta_{opt} (\Lambda_{mg} + \Delta L |I| \sin \delta_{opt})$$

Applicando allora questi ragionamenti al nostro motore

```
%Parametri motore Lafert

%parametri elettrici
Ld=12e-3;
Lq=20e-3;
Lmg=0.088;
p=5;

%parametri meccanici
J=2.9*1.7e-3;
B=1e-3;
n=200;

%inverter
```

udc=50;

Per una corrente nominale $|I| = 10A$

Ricavo:

$$\delta_{opt} = 29^\circ$$

$$M_n = 8.37 Nm$$

Di conseguenza

$$i_{dn} = -4.84 A$$

$$i_{qn} = 8.75 A$$

calcoliamo allora i flussi:

$$\begin{cases} fl_{dn} = L_d i_{dn} + \Lambda_{mg} = 0.03 Vs \\ fl_{qn} = L_d i_{qn} = 0.175 Vs \end{cases}$$

$$fl_n = \sqrt{fl_{dn}^2 + fl_{qn}^2} = 0.177 Vs$$

Questi sono i valori di riferimento relativi al punto di lavoro nominale che possiamo inserire come valori di riferimento nell'algorithm.

Osserviamo allora le stampe per un tempo di simulazione pari a 0.5 sec, velocità meccanica 200 rpm, senza coppia di carico

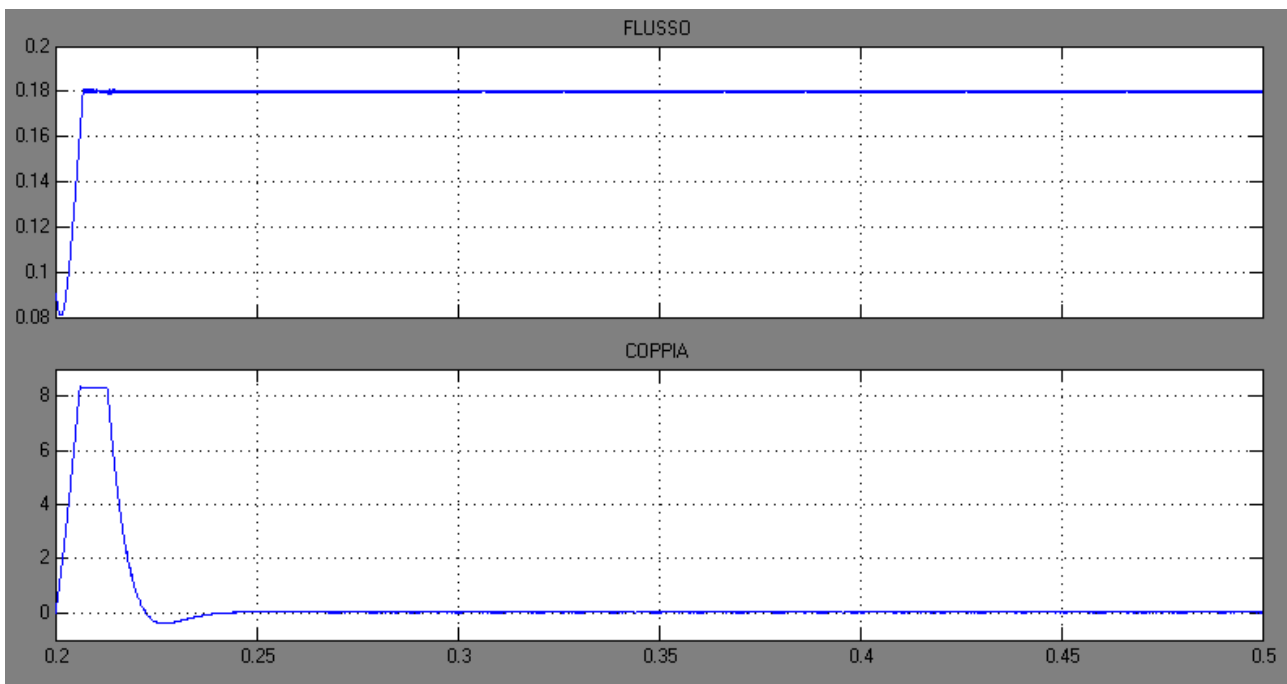


Grafico 6.5 Flusso e Coppia simulati

Le correnti nel relativo piano:

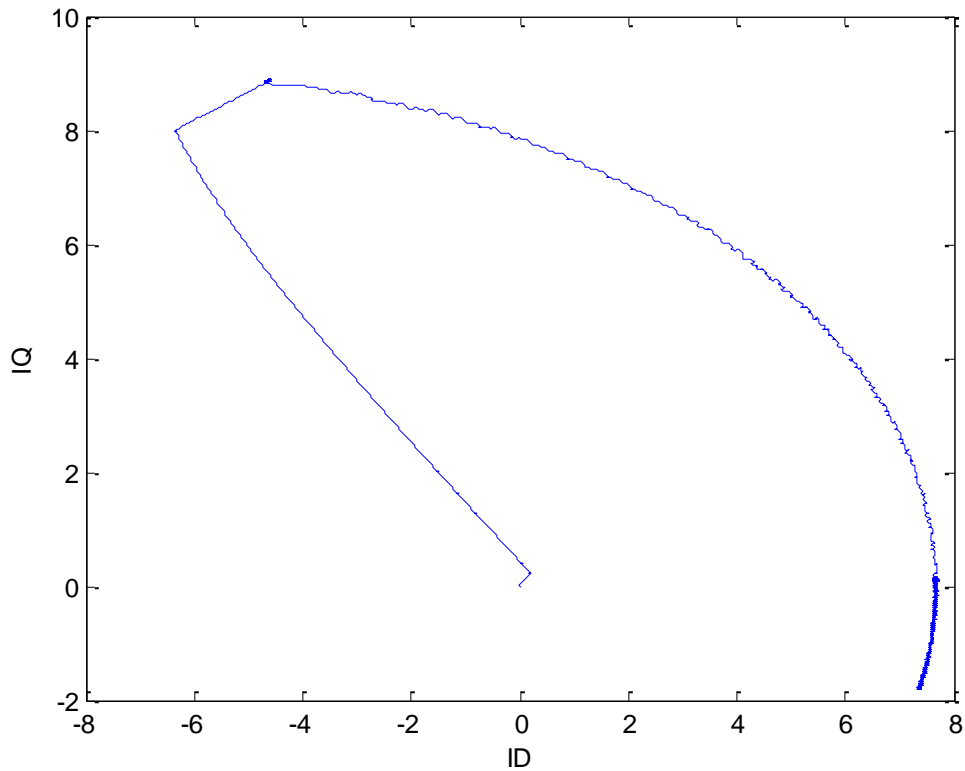


Grafico 6.6 piano delle correnti simulate

Andiamo ad analizzare i risultati sperimentali ottenuti senza imporre coppia di carico e per una velocità di 200 rpm.

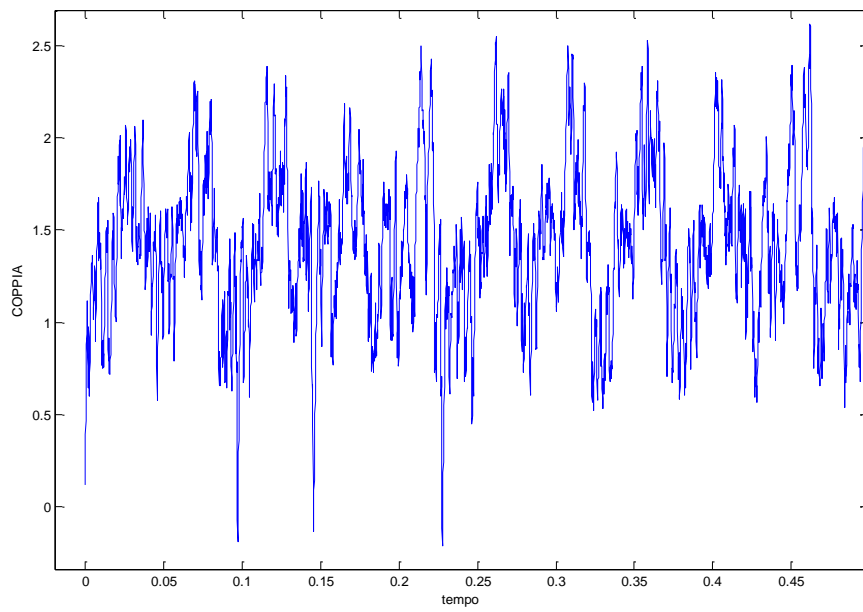


Grafico 6.6 Coppia reale

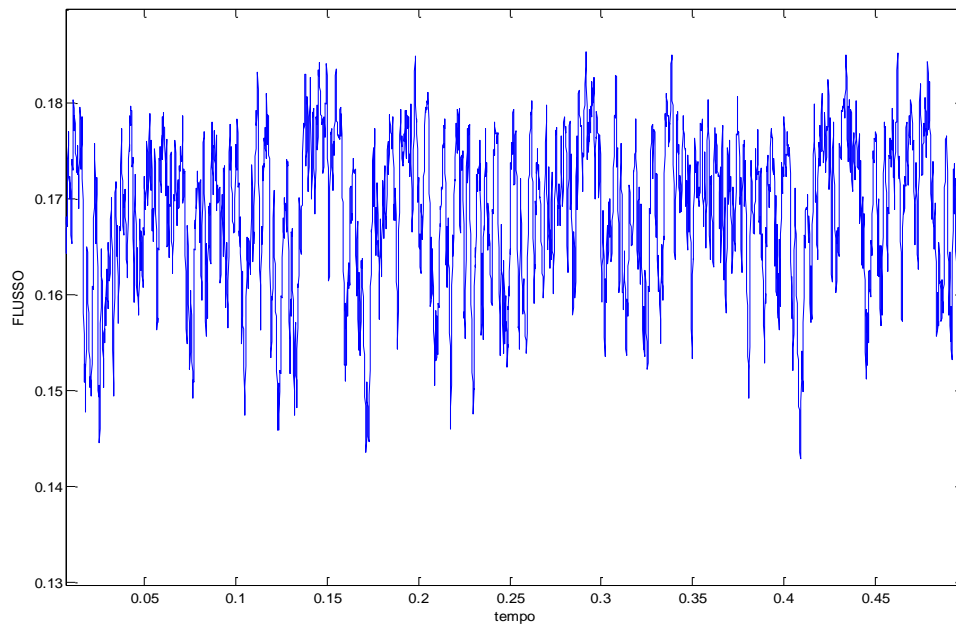


Grafico 6.7 Flusso reale

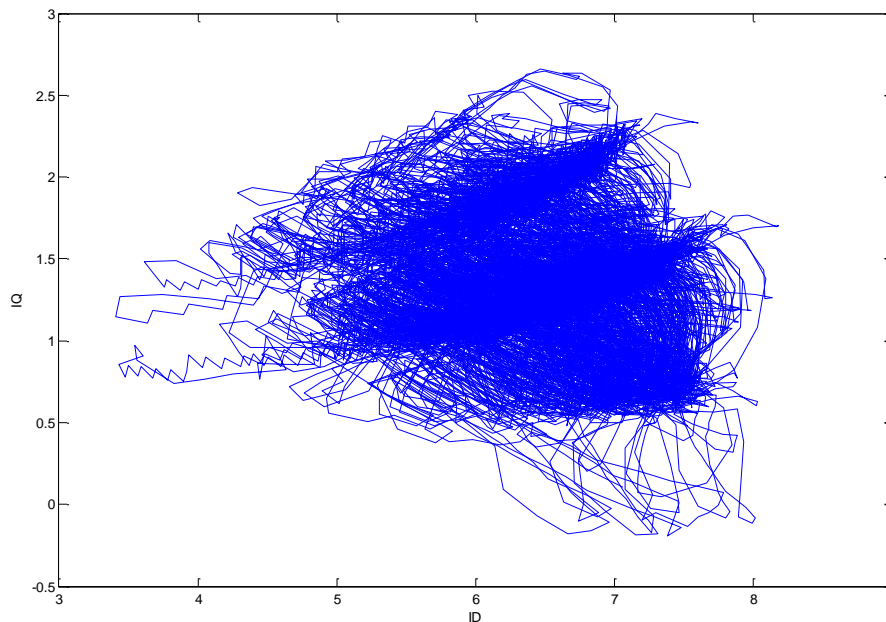


Grafico 6.8 piano delle correnti reale per un punto di lavoro a coppia nulla

6.6-Commenti simulato-reale

Osservando le precedenti figure notiamo subito che il comportamento simulato è in genere molto più regolare di quello reale, in particolare:

flusso simulato-flusso reale: i valori sono circa i medesimi, ovvero pari a 0.177 Vs calcolati analiticamente anche se nel reale si nota meglio il ripple del controllo all'incirca di 0.3Vs

piano correnti simulato-piano correnti reale: siamo nel punto di lavoro a coppia nulla, possiamo quindi pensare che il grafico 6.8 è un ingrandimento del 6.6 nella zona in cui la componente in quadratura è nulla o quasi.

Coppia simulata-coppia reale: mentre la prima si annulla raggiunti 200 rpm, la seconda non esattamente e ciò è dovuto al fatto dei coefficienti con cui peso coppia e flusso nell'algoritmo di comando. Ovvero per essere più stringenti sull'uno si perde controllo sull'altro e viceversa. Una possibile soluzione consiste nel normalizzare entrambi i fattori in modo che le dimensioni dei due siano le stesse.

CONCLUSIONI:

In questa trattazione abbiamo dimostrato che è possibile attuare il controllo diretto di coppia e di flusso per entrambi i tipi di motore brushless analizzati, in particolare se adottiamo il DTC predittivo passare dall'una all'altra tipologia è estremamente semplice.

Possiamo affermare inoltre che è un tipo di azionamento robusto e veloce che però presenta oneri computazionali non trascurabili vista la quantità di calcoli che il microprocessore deve eseguire.

In particolare durante la fase di sperimentazione si sono riscontrate alcune problematiche, forse utili spunti per studi futuri.

In primo luogo sarebbe utile implementare una trasformazione matematica contraria a quella vista nelle pagine del paragrafo 6.5 ossia ricavare dalla coppia le correnti in forma cartesiana (sistema d-q) o polare (modulo e fase):

$$i_n = f(\delta_{opt}, i_n) \leftarrow \delta_{opt} = f(T_n) \leftarrow T_n$$

Percorro in senso inverso la funzione vista.

Andiamo ora a precisare quello che è il problema principale del controllo: teniamo ben presente che in questa trattazione l'obiettivo principale è stato l'implementazione del DTC lavorando a flusso (e quindi tensione) costante.

La macchina IPM analizzata presenta, date le sue caratteristiche costruttive, un termine (corrente i_d di cto cto):

$$i_{ccd} = -\frac{\Lambda_{mg}}{L_d} = 7.33$$

che sta dentro al limite di corrente (una crf di raggio 10 A).

I nostri punti di lavoro a tensione costante stanno quindi su una curva ellittica con centro i_{ccd} e vertice caratterizzato da i_q nulla e una componente i_d positiva (Grafico 6.6).

La nostra idea è quella di muoverci su una curva che assomigli il più possibile a quella MTPA che presenta vertice dell'ellisse nell'origine degli assi ovvero componenti di corrente nulle per coppia nulla.

Si consiglia quindi un ulteriore controllo, esterno al DTC che provvederà a regolare il flusso della macchina in base alla coppia applicata in modo che la macchina lavori effettivamente su una curva MTPA

BIBLIOGRAFIA:

[1]-M. Preindl S.Bolognani -Model Predictive Direct Speed Control with Finite Control Set of PMSM-VSI Drive Systems-IEEE paper

[2] -P. Todesco- Controllo diretto di flusso e di coppia-Tesi

[3]-S. Bolognani- Dispense del corso di Azionamenti Elettrici

[4]- V. Veronese- Appunti delle lezioni del corso di Azionamenti Elettrici